Leopold-Franzens-Universität Innsbruck

Department of Computer Science

Security & Privacy Lab

Dissertation submitted in partial fulfillment of
the requirements for the degree of Doctor of Philosophy
in the discipline of Computer Science

The Role of Lossy Compression in Digital Image Forensics

Nora Hofer, M. A.

nora.hofer@uibk.ac.at

advised by

Univ.-Prof. Dr.-Ing. Rainer Böhme

February 2026

Für Tamara

Summary

Lossy image compression is a fundamental source of traces for passive image forensics to infer how an image was acquired and processed. As compression algorithms evolve, however, the statistical and semantic properties of compressed images change. This development can both enable new forensic cues and challenge assumptions underlying existing methods. In this dissertation, we investigate the role of modern lossy image compression in digital image forensics and aim to improve the reliability of forensic analysis in practice. In the first part, we focus on the emerging neural image compression. These learning-based codecs are trained to achieve high perceptual quality at unprecedented bitrates. However, they may introduce miscompressions: reconstruction errors that change the semantics of an image or image detail. We identify and formalize this novel class of compression artifacts, propose a taxonomy, and quantify the frequency of their occurrence across different codecs. To verify our subjective understanding of image semantics, we conduct a controlled user study to evaluate human perception of miscompressions. In addition, we curate and release a large dataset of miscompressions annotated in reconstructed images by three trained human labelers. The dataset is intended to pave the way for further research on mitigation strategies. In the second part, we investigate conventional JPEG compression as it is deployed in practice. Many forensic techniques assume a canonical encoder model, yet practical implementations might incorporate optimizations. Focusing on the *mozjpeg* codec, we analyze encoder optimizations, most notably the trellis optimization, and describe how it modifies transform coefficients in forensically relevant ways. We demonstrate that these benign trellis artifacts can confound existing forensic methods and propose analytic and learning-based detectors that identify such traces and can serve as preprocessing steps for more robust analyzes. Overall, this dissertation seeks to strengthen the understanding of current and future compression codec implementations and contribute to increasing the reliability of forensic analysis.

Acknowledgements

I would not have been able to complete this dissertation without the encouragement and support of the wonderful people around me. I am grateful to the former and current members of the *Security and Privacy Lab*, as well as the SCLIC team, who made my time at the office both inspiring and genuinely joyful. In particular, I would like to thank Tobias, without whom I would never have joined the lab, and Alex, who welcomed me so warmly and always made me feel comfortable asking questions. Special thanks go to my dear friend Martin for his honest feedback, constant encouragement, and the many joyful moments we shared. I am also very thankful to Benedikt, from whom I learned a great deal and who never tired of answering my many questions, and Kristina for valuable feedback on the draft of this manuscript. My biggest thank you is dedicated to Rainer for unwavering faith in me and support through every challenge along the way. Thank you for your trust and all the opportunities you provided me with. Finally, I would like to thank all the wonderful people I am fortunate to call my friends and family, whose love, patience, and support carried me along the way.

Contents

List of Acronyms

Note: This list includes only acronyms that are used in abbreviated form without repeated definition. Acronyms occurring exclusively in the appended research papers of Part II are omitted.

AC	Alternating current
BPP	Bits per pixel
CDC	Conditional diffusion compression
DC	Direct current
DCT	Discrete cosine transform
GAN	Generative adversarial network
H	Hypothesis
HiFiC	High fidelity compression
HVS	Human visual system
JBIG2	Joint Bi-level Image Experts Group 2
JPEG	Joint Photographic Experts Group
JPEG AI	JPEG artificial intelligence
JRM	JPEG rich model
LPIPS	Learned perceptual image patch similarity
MSE	Mean squared error
MS–SSIM	Multi–scale SSIM
PIM	Perceptual information metric
PSNR	Peak signal-to-noise ratio
RGB	Red, green, and blue
RLE	Run length encoding
SCLIC	Semantic changes of learning-based image compression
SSIM	Structural similarity index
STF	Symmetrical transformer
VAE	Variational autoencoder
YCbCr	Luminance-chrominance color space

Part I

Research Summary

1. Introduction

Digital image forensics is a field of research that explores methods for verifying the authenticity of digital images. The literature distinguishes forensic approaches based on the traces which are exploited by forensic methods. Active approaches verify images using information that was specifically embedded into the image for this purpose, such as cryptographic signatures [59, 146] or digital watermarks [17]. In this thesis, we focus on passive approaches. Methods in passive approaches exploit distinctive traces, which are introduced into the image during the acquisition and subsequent processing operations. By modeling such traces, passive methods try to reconstruct the image processing history and identify inconsistencies that may indicate image manipulation or unexpected processing operations. A broad range of passive forensic methods exist. Scene-based methods, for example, use visible traces in the image content to reveal manipulations, *e.g.*, by assessing the physical plausibility of depicted real-world phenomena. Inconsistencies in shadows, reflections, or lightning that violate physical constraints may indicate manipulations. Signal-based methods rely on assumptions about the statistical distribution of the image signal, which results from the acquisition and processing operations. When an image is manipulated, for instance, by pasting a part from a different image, the inserted region often needs to be rotated or scaled. Such adjustments involve interpolation, which introduces artifacts in the signal that deviate from the expected statistical distribution and can therefore indicate manipulation [136]. Another source of signal-based forensic traces in the image is lossy image compression.

The goal of lossy image compression is to reduce file size while preserving visual image quality. To achieve this, compression algorithms typically transform an image into a domain that concentrates signal energy. This transform allows the algorithm to employ quantization and remove information that is assumed to contribute least to the human visual perception. Entropy coding then reduces the remaining redundancies and produces a compressed bitstream. Lossy compression algorithms balance two conflicting objectives. They try to minimize the bitrate, commonly measured in bits per pixel (BPP), while maintaining an acceptable level of visual quality. Visual quality is commonly expressed in terms of distortion relative to the original image. Distortion appears as visible artifacts, for instance, as blurring or blockiness. This compromise is known as the rate–distortion tradeoff and is balanced in practice by the codec that implements the compression algorithm.

Image compression standards, such as the widely deployed Joint Photographic Experts Group (JPEG) standard [161] or the recent JPEG artificial intelligence (JPEG AI) standard [80], define the syntax of the compressed bitstream to ensure that encoded images can be decoded correctly by any compliant decoder. However, these standards intentionally leave design choices to the encoder, notably those affecting the rate–distortion tradeoff. As a result, compression standards guarantee interoperability at the decoding stage, but do not prescribe which information from the original image will be preserved in the compressed image. In practice, encoder implementations are continuously refined in response to technological advances or to meet application-specific requirements. As a result, different encoders may produce different outputs from the same image while remaining

fully compliant with the standard.
This dissertation comprises five research papers, in which we investigate developments in lossy image compression and their role in digital image forensics. Our overarching research goal is to contribute to improving the reliability of forensic methods when they are applied to images produced by current compression implementations. We address two branches of lossy image compression, namely the emerging neural image compression, and the conventional JPEG compression. The maturity level of these two branches differs, and with it, the extent of existing forensic literature. Our approaches to address their role on digital image forensics reflects this.

Neural Image Compression Neural image compression constitutes a recent development in lossy image compression. Existing codecs employ neural networks with learned analysis and synthesis functions and achieve compression rates that, in terms of perceptual quality at a given bitrate, surpass conventional image compression codecs. Neural codecs are commonly based on nonlinear transform coding, which allows them to adapt to varying data distributions [15]. In contrast to conventional codecs, they can be trained to optimize for different distortion measures which are designed to better align with human perception of visual image quality. Advances in the perceptual optimization have further been achieved by the integration of generative models in the reconstruction of the compressed image. Current codecs employ architectures based on variational autoencoders (VAEs) [13, 14], generative adversarial networks (GANs) [119], diffusion models [170], or transformer networks [7, 176]. These codecs achieve reconstructions of high perceptual quality at low bitrates by synthesizing "expensive" parts of the image.

This dissertation studies a resulting pitfall of neural compression, which we call "miscompression", in the following three research papers.

A) N. Hofer and R. Böhme. A taxonomy of miscompressions: Preparing image forensics for neural compression. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2024 *(Workshop, no CORE ranking available)*

B) N. Hofer and R. Böhme. When the codec hallucinates: User perceptions of miscompressed images. In *CHI Conference on Human Factors in Computing Systems*. ACM, 2026. to appear *(Conference, CORE 2023 A* ranking)*

C) N. Hofer and R. Böhme. Challenging cases of neural image compression: A dataset of visually compelling yet semantically incorrect reconstructions. In *International Conference on Multimedia*, pages 13318–13324. ACM, 2025 *(Conference, CORE 2023 A* ranking)*

We show that current codecs introduce a novel type of compression artifacts that, unlike conventional compression artifacts, alters the semantics of an image or image detail. Examples of miscompressions include flipped digits, altered colors of depicted objects, or the disappearance of small but semantically meaningful objects, such as tattoos or persons in the background of an image. Miscompressions are small, in the order of 0.1% of all pixels, and can easily go unnoticed. Importantly, there is often no reliable way to infer a semantic change by inspecting the reconstructed image alone, as the affected regions are often visually plausible and locally consistent. Beyond the resulting potential risks for end users, such as misunderstandings or the inadvertent spread of misinformation, miscompressions might also pose a risk in future image forensics scenarios. In particular, this could affect scene-based forensic methods that often rely on the semantic integrity of small image details. Until recently, neural image compression has been limited to academic research. The approval of JPEG AI [80] as an official neural compression standard indicates a growing interest from industry,

and may mark a potential transition into the real world. The possibility of semantic changes in reconstructions is occasionally acknowledged in the literature on neural image compression.[1] However, these changes are typically described only briefly in limitation sections and the frequency with which they occur is not quantified.

In this dissertation, we describe and categorize miscompressions as a novel, distinct class of compression artifacts and measure how frequently they occur. We do so across multiple current neural compression codecs, including the JPEG AI reference implementation. Our research shows that miscompressions are not rare outliers. In fact, they occur in up to half of the neurally reconstructed images, and we observe them in images produced by all analyzed codecs. Studying miscompressions is challenging from both technical and semantic perspectives. For one, no currently known method can automatically and deterministically generate miscompressions for given images or image details. Furthermore, the semantic meaning of an image or image detail is inherently subjective, and changes to this meaning cannot easily be identified automatically, at this point. In our initial paper, we define that a miscompression is present when there is a discrepancy in the verbal description for a specific image detail before and after neural compression [71, p. 3]. In that work, the semantic interpretation underlying this verbal description was based on our own subjective perception. However, semantics are strongly influenced by human experience, cultural background, and the context in which an image is viewed. Consequently, both the perceived severity of miscompressions and the assessment of whether they may lead to problems, such as misunderstandings, are subjective. Even when effects can be measured objectively, determining the appropriate level of mitigation effort is often difficult. The subjectivity of miscompressions makes such decisions even more challenging. Therefore, we conduct a user study to establish intersubjective agreement and evaluate how consistently semantic changes and their severity are perceived across a broader group of people. We design the study as a controlled environment user study and place it in the hypothetical context of social network image sharing. This context was chosen because we could not expect participants to be familiar with neural compression. Moreover, we aimed to collect their unbiased perception without explicitly introducing the concept. Participants compared two versions of the same image and assessed the risk of misunderstandings between a viewer of the original image and a viewer who has access only to the second version, which was distributed online. The results validate our initial subjective perception and show that participants perceived an increased risk when the second image contained a miscompression, compared to when it was processed otherwise. Another difficulty arising from the semantic nature of miscompressions is the challenge of reliably preventing or detecting miscompressions automatically. To address this, we curate a large dataset that could facilitate future research on the implications for forensics, as well as the development of improved neural compression codecs. The dataset is based on never-compressed images from existing benchmark datasets, which we compress using different neural codecs, quality settings, and optimization metrics. We then train expert human labelers to compare the original images to the neural reconstructions and annotate semantic changes within the images. The dataset is publicly available on Zenodo (DOI:10.5281/zenodo.16780952) [73].

JPEG Compression Given the popularity of JPEG images on the web [44], JPEG forensics is an extensively researched area in digital image forensics. When researchers develop forensic methods that exploit traces from compression, they make assumptions about the images and the processes that generated them. Commonly, researchers assume canonical encoder behavior, modeled according to the baseline described in the standard or a reference implementation. Deviations of this baseline

[1] For example, Relic et al. note that "in specific cases" diffusion-based decoding "might result in inaccurate reconstruction, such as bending straight lines or warping the boundary of small objects." [142, p. 316], and Mentzer et al. remark that "in theory [a generator G] can produce images that are very different from the input." [119, p. 10].

model can constitute forensic traces. Such traces can provide cues about the processing history, for example, to fingerprint known encoder implementations, or indicate manipulations. However, such traces might also pose a risk to the reliability of forensic analyses. If methods are sensitive to subtle changes in the image signal, they might misinterpret benign artifacts of unknown codecs as traces of manipulation.

In this dissertation, we analyze JPEG codecs in practice, specifically the *mozjpeg* [121] codec, and investigate the implications of its encoding optimizations for digital image forensics in the following two research papers.

D) N. Hofer and R. Böhme. Progressive JPEGs in the wild: Implications for information hiding and forensics. In *Workshop on Information Hiding and Multimedia Security*, pages 47–58. ACM, 2023 *(Workshop, CORE 2023 C ranking, Best Student Paper Award)*

E) N. Hofer. Increasing trust in image analysis by detecting trellis quantization in JPEG images. In *International Conference on Image Processing*, pages 3834–3840. IEEE, 2024 *(Conference, CORE 2023 B ranking)*

Designed for web publishers, Mozilla's *mozjpeg* codec improves user experience by employing the progressive mode and reducing web page loading times as it yields smaller file sizes at comparable image quality. We find that the improved rate–distortion tradeoff is accomplished primarily through the use of the so-called trellis optimization. Similar as known from video encoding, trellis optimization relaxes the classical separation between quantization and entropy coding by incorporating entropy coding costs into the quantization process. This results in shorter bitstreams.
While the research community has largely neglected *mozjpeg* so far, we demonstrate that its optimization algorithms introduce forensically relevant changes in the image. In our papers, we show how the trellis optimization algorithm modifies transform coefficient values to reduce file size and provide an example in which this change of the image signal impacts the reliability of a forensic method. To address this and contribute to improving forensic reliability, we propose pre-processors that detect artifacts of trellis optimization in the signal of an image under investigation. The proposed analytic and learning-based detectors are based on statistical models of traces of trellis optimization. We also show how the optimization of the progressive mode can leave traces in the JPEG file which can be exploited to identify the source social network from which the image originated.

Other Papers Not Included During my time at the Security and Privacy Lab, I have co-authored other papers, not included in this dissertation:

F) M. Beneš, N. Hofer, and R. Böhme. Know your library: How the libjpeg version influences compression and decompression results. In *Workshop on Information Hiding and Multimedia Security*, pages 19–25. ACM, 2022 *(Workshop short paper, CORE 2021 C ranking)*

G) M. Beneš, N. Hofer, and R. Böhme. The effect of the JPEG implementation on the cover-source mismatch error in image steganalysis. In *European Signal Processing Conference*, pages 1057–1061. IEEE, 2022 *(Conference, CORE 2018 B ranking)*

H) A. Schlögl, N. Hofer, and R. Böhme. Causes and effects of unanticipated numerical deviations in neural network inference frameworks. *Advances in Neural Information Processing Systems*, 36, 2024 *(Conference, CORE 2023 A* ranking)*

Papers *F)* and *G)* are collaborations with Martin Beneš and Rainer Böhme. They are not included in this dissertation because I am not the main author, and both are short papers. Paper *H)* is a

collaboration with Alexander Schlögl and Rainer Böhme. It is not included as I am not the main author, and the paper does not fall within the thematic scope of this dissertation.

Outline This cumulative dissertation is structured into two parts. In Part I, we present a research summary of the five research papers that constitute this thesis. Part II contains the full versions of the papers, reformatted to comply with the formatting of this document. The original publications are available in the respective conference proceedings. The remainder of Part I is organized as follows. In Chapter 2, we provide the necessary theoretical background on digital images (Section 2.1), digital image forensics (Section 2.2), and lossy image compression (Section 2.3). In Section 2.4, we review existing research on the role of lossy compression in digital image forensics. The Chapters 3 and 4 present our core research contributions to the forensics literature, with focus on neural image compression (Chapter 3), and JPEG compression (Chapter 4). Both chapters follow a common structure. First, we introduce and motivate the research problems in Sections 3.1 and 4.1. Then, we summarize the respecive research papers in Sections 3.2 and 4.2, and discuss key results in Sections 3.3 and 4.3. Lastly, in Sections 3.4 and 4.4, we identify open problems and suggest directions for future research. Chapter 5 concludes the dissertation.

2. Preliminaries

To understand the role of image compression for digital image forensics, we need to understand the foundational principles. This section provides the theoretical background relevant for the remainder of this dissertation. We introduce a working definition for the digital image and introduce the imaging pipeline in Section 2.1. In Section 2.2, we provide an overview of digital image forensics and describe selected methods from three different analysis directions. In Section 2.3, we explain the core principles of lossy image compression and differentiate between conventional and neural image compression algorithms. Finally, we connect the two previous sections and review related work on differences in compression implementations that may have implications for forensics in Section 2.4.

2.1 Digital Images as Forensic Evidence

A digital image is a projection of a continuous scene to a discrete representation. A scene is part of the real world and can refer to any natural phenomena or describe arbitrary imaginary phenomena that result from human creativity [30, p. 81]. Thus, an image conveys information about a depicted scene that, when interpreted by a human, translates to a particular semantic meaning [94, p. 7]. A digital image is typically modeled as a two-dimensional array of pixels where each pixel represents the intensity or color of the image at a specific spatial location. For grayscale images, a single numerical value encodes brightness. Color images commonly use three channels, red, green, and blue, which together form a pixel's color through additive mixing [89]. Most widely used image formats represent pixel values with 8-bit precision, allowing values from 0 to 255.

2.1.1 The Imaging Pipeline

Pixel values are the result of a multistep imaging pipeline that consists of image acquisition and optional post-processing operations. Each component of this pipeline affects the final appearance and characteristic properties of the digital image. Figure 2.1 depicts a typical variant of the pipeline. We will now provide working definitions for the acquisition, processing operations, and manipulation.

Acquisition The image acquisition is the interface between the real and the digital world and projects a 3D scene into a 2D image. Optical light consists of waves of various wavelengths, which humans perceive as different colors. During acquisition, the light that is reflected by the captured scene enters the camera through an optical lens, which projects this incoming light field onto the camera sensor (typically a complementary metal-oxide-semiconductor (CMOS) sensor in modern digital cameras). The sensor acts as the "film" of the digital camera. It is a two-dimensional array of photoelectric elements that get electrically charged when exposed to light [48]. This means, the sensor is sensitive to light intensity but not colors. It is therefore overlaid by a color filter array (CFA), a grid of small, colored filters that allow only light of a specific range of wavelengths to pass

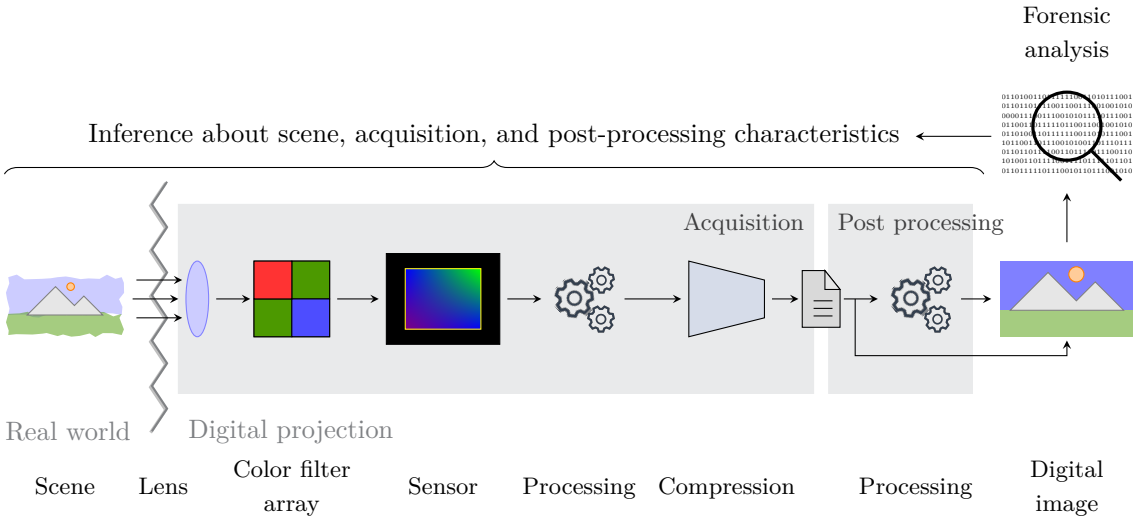


Figure 2.1: The imaging pipeline of a digital image. Acquisition refers to all in-camera operations before the image is stored. Post-processing steps are commonly applied after the image leaves the acquisition device, *e.g.*, re-compressed during transmission. In practice, the pipeline may incorporate additional or differing steps. Parts of the figure are adopted from lecture slides of the Security and Privacy Lab and the dissertation by Matthias Kirchner [94].

through [147]. As a result, each sensor cell records either red, green, or blue color information. The sensor then converts the intensity information into a digital signal. The missing color measurements are reconstructed by an interpolation algorithm, called demosaicing [104]. The resulting pixel values are commonly further processed to increase the perceived image quality and reduce the memory required to store the image.

Processing We denote processing as any change of the initial image as it was recorded by the sensor, both in-camera, and external, *i.e.*, post-processing operations. Common in-camera processing operations include white balancing [16], gamma correction [137], noise reduction [33], sharpening, and commonly the final operation of the acquisition pipeline, image compression. We provide background on image compression in Section 2.3 and refer the reader to one of many excellent books on digital image processing for details on other in-camera operations [153]. After the image leaves the acquisition device, it might undergo further post-processing operations. While in-camera processing operations depend on the acquisition device, post-processing operations often depend on the use case of the image. For example, images that are shared via messaging apps or uploaded on social network platforms might undergo color corrections, retouching, beauty filters, and double compression. Images that are used in newspapers or documentation might be cropped or sharpened.

Manipulation While processing refers to any change of the initial image, we denote manipulations as specific processing operations, that, whether intentional or unintentional, cause a change of the semantic meaning of the image or image details. Harmless cropping can be a manipulation, if it removes objects, people, or background details that contribute to the semantic meaning of a scene. For example, a police officer can appear aggressive, if cropped details cover that they were responding to danger. Changing the overall color of an image can change the perceived lighting conditions, creating the impression that the portrayed scene took place at a different time, or season.

Retouching of *e.g.*, tattoos or birthmarks might destroy information relevant to identify a person. Examples of manipulations that might change the semantic meaning intentionally include splicing and copy-move forgeries, face swapping and reenactment, inpainting, and morphing. Manipulations have in common that they modify the truthful representation of a captured scene and therefore impair image authenticity.

2.2 Principles of Digital Image Forensics

A task of digital image forensics is to verify the authenticity of images. In this thesis, we focus on passive methods that aim to reconstruct the image processing history and detect signs of manipulations. To do so, these methods use distinctive image characteristics that are shaped during acquisition and processing operations.

Traces We describe the term trace as a detectable *visual*, *structural*, or *statistical* pattern that can be attributed to a specific step in the imaging pipeline. It is important to mention that not only the presence or characteristic of traces provide insights in forensic analyses, but also the non-existence of expected traces, or deviations thereof. We refer to such deviations as inconsistencies, *i.e.*, traces that are compatible with reference traces of a different, possibly contradictory, imaging pipeline. An example would be a mismatch between an image's dimensions and the sensor size of the camera model as specified in the accompanying metadata.

Analysis Directions In the remainder of this section, we introduce three analysis directions that differ by the traces the forensic investigator uses for their analysis. For each of these directions, we present prominent established methods and refer the reader to the literature of a more extensive overview [53, 90, 133]. We will omit the forensic branches that address source attribution [39, 113] and information hiding [42, 55].

2.2.1 Scene-Based Analysis

Scene-based analyses investigate the content of an image and typically have one of two goals: (i) revealing image manipulations or (ii) extracting contextual information about the depicted scene. Methods aimed at revealing manipulations often rely on visual traces that originate from the imaging pipeline, such as color fringes from chromatic aberration. In an ideal camera, every ray of light passing through the lens reaches the sensor with pinpoint accuracy. In practice, however, this does not work equally well for all colors [32, p. 685]. Depending on its wavelength and position relative to the optical axis, the light reaches the sensor with slight spatial deviations. This results in color fringes that can be used to identify inconsistencies introduced by manipulation [63, 85]. Another class of scene-based analyses uses physical traces to reveal image manipulations. The premise of these analyses is that real phenomena follow the laws of nature, whereas manipulations might introduce inconsistencies, or violate physical constraints. Physical traces include inconsistencies in lighting [84, 91, 145], shadow geometry [175], reflections [86], perspective projection [41], or other aspects of physical scene modeling. Existing methods estimate the physical properties of the scene and assess whether these estimates are mutually consistent across the image [144, pp. 219]. Scene-based analyses might also use depicted objects as semantic traces to identify manipulations by leveraging contextual world knowledge. Examples of such semantic traces include clocks or watches that display inconsistent times, as well as implausible or non-existent street names and landmarks. Scene-based analyses that focus on the second goal, extracting contextual information, require semantic scene

understanding. Semantic scene understanding is the identification and interpretation of the depicted scene, the objects, actions, and events [87, 173]. Examples for methods include photographic comparisons [165] and content analysis [120]. Approaches for scene-based analysis may be fully based on human visual inspection or tool-based [116].

2.2.2 Metadata-based Analysis

Structural traces are present in the file syntax or metadata, *i.e.*, information embedded in the image file header. Examples of metadata include the acquisition date and location, the camera model, and acquisition device settings, comment-tags, and the thumbnail. Inconsistencies in the metadata or file syntax can indicate manipulations [61, 62, 134]. Additionally, metadata can provide contextual information that might be important during forensic analysis [132] or improve the confidence when interpreting the results of other forensic methods [134]. It can provide information about the image acquisition and processing history. For example, the camera model and acquisition settings can allow for a comparison of noise patterns or lens distortions [50, 92] and serve as traces for source identification [92]. Included comment-tags from processing or editing software like "Adobe Photoshop" or "GIMP" indicate post-processing [114]. The image thumbnail is a miniaturized, compressed version of a main image that is used for preview without the need to decompress the main image. Depending on the editing operation, the original thumbnail might persist in the metadata of the processed image, thus "leaking" insightful information about image processing operations [20, 123]. The metadata of images that underwent compression include information used during the decompression. Examples are quantization and Huffman tables in JPEG images (*cf.* Section 2.3). This information might differ across codecs, cameras, and photo-editing software and therefore serve as a forensic trace [51, 130].

2.2.3 Signal-based Analysis

Signal-based forensics analyze image data using the low-level signal values. Typically, methods make assumptions about the *statistical* distribution of the signal that is to be expected for a given imaging pipeline. During manipulation, parts of an image often undergo transformations which introduce detectable traces. One example are interpolation operations, which change dependencies between neighboring pixels [93, 136]. Many signal-based forensic methods use traces from compression, specifically JPEG, for the detection of manipulations. A common premise is that a manipulated JPEG image is saved again in JPEG format and thus recompressed. If the quantization tables of the two compression operations do not match, the resulting image may contain detectable double-compression artifacts [52, 135, 172]. If altered regions, *e.g.*, in copy-move forgeries, come from images of different compression histories, the resulting image contains locally inconsistent artifacts [58]. Compression artifacts can also be used for the reconstruction of an image's processing history [154], for example revealing information about the source compression codec [34, 110].

2.3 Principles of Image Compression

Image compression applies concepts from information and coding theory to encode image data efficiently by identifying redundancies. Image compression can be categorized into lossless or lossy algorithms. Lossless compression requires perfect restoration of the original image after decompression, while lossy compression reduces the file size by permanently discarding some image data based on characteristics of the human visual system. In this thesis, we focus on lossy image compression

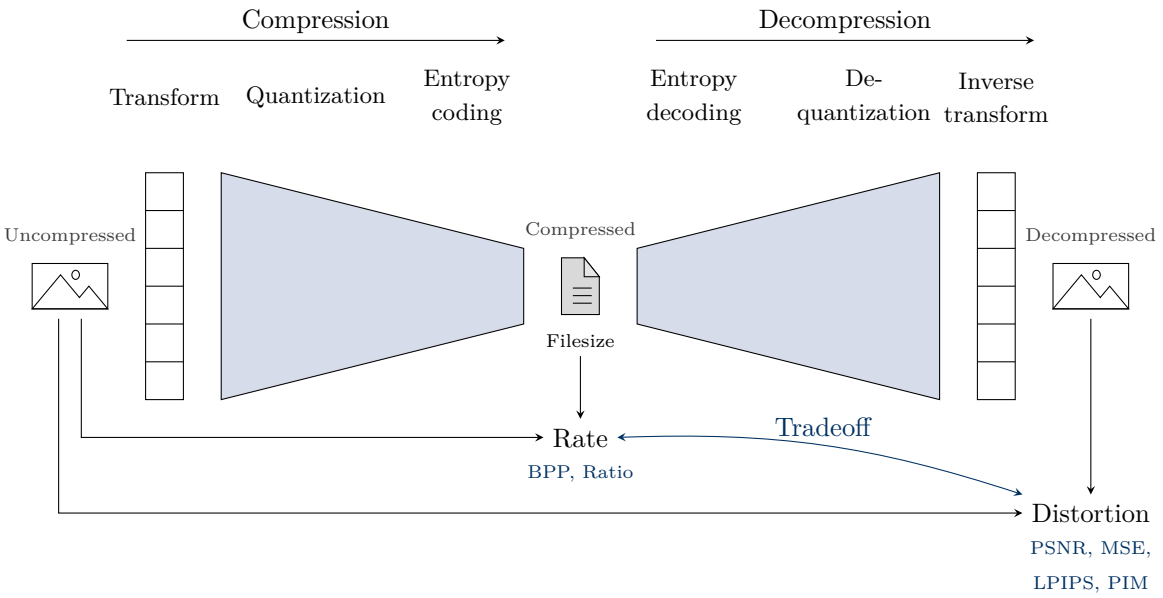


Figure 2.2: An input image is encoded into a representation of fewer bits during compression. The resulting bitstream is saved in the image file. During decompression, the steps are reversed, to the extent possible. The compression pipeline is designed to optimize a rate–distortion tradeoff. The rate describes the compression efficiency and is measured by comparing the size of the input image to the bits needed to store the compressed image. The distortion describes the quality degradation introduced during compression and is measured by comparing the input image to the output image.

and the characteristics it leaves behind. Existing algorithms can be categorized into conventional and neural compression methods. Conventional compression relies on hand-designed transforms and probabilistic models derived from principles of signal processing and coding theory. They are often tuned based on psychovisual experiments involving human subjects [108, Ch. 5]. Neural compression replaces some of these operators with neural networks. In particular, it uses trained models for the transform coding and for the probability models used in entropy coding. In the following section, we recall the core concepts of lossy image compression and describe both the established conventional, as well as the emerging neural image compression.

2.3.1 Core Concepts of Lossy Image Compression

The goal of image compression is to reduce the number of bits required to store or transmit an image while preserving an acceptable level of visual quality.

Compression Pipeline Although existing codecs differ in their implementation, they commonly rely on a pipeline incorporating the same three key components: transform coding, quantization, and entropy coding. Figure 2.2 visualizes the pipeline. The first component, transform, maps the spatial pixel values into a domain where the most meaningful visual information is concentrated in fewer coefficients. A typical representation is the discrete cosine transform (DCT) [5] domain. The next component, quantization, is a deliberately lossy process. It maps a range of values into single discrete values, reducing the precision of coefficients based on their frequency, *i.e.*, discarding more information from high-frequency components to which the human visual system is less sensitive [35]. Finally, the lossless entropy coding approximates Shannon's theorem for a noiseless channel [152,

Part I, Section 9] by encoding a sequence of quantized values into a sequence of bits, thereby assigning shorter code words to frequently occurring values and longer ones to rare values. The better the input distribution is known, the tighter the approximation and the shorter the resulting bitstream.

The decompression pipeline reverses these steps to the extent possible. Due to the permanent discarding of information during quantization, and rounding and truncation during transformation, the decompressed image differs from the input image.

Rate–Distortion Tradeoff Image compression can be described as an optimization problem that tries to find an optimal balance between the compression rate and the distortion introduced in the decompressed image. A common way to measure compression efficiency is the compression ratio. It is defined as the ratio of the original uncompressed image size to the compressed image size, *i.e.*, the uncompressed size divided by compressed size. The uncompressed file size is given by the number of pixels in the image calculated as height $\times$ width $\times$ bit depth $\times$ number of color channels. Compression efficiency is also expressed in BPP, which normalizes storage cost by image resolution. The introduced distortion is defined as the difference between the uncompressed input image and the decompressed image. Quantifying this difference is non-trivial, and there is an ongoing discussion in the field regarding which metric best represents human perception [28, 43]. Existing metrics can generally be separated into two categories. Mathematical approaches measure distortion based on pixel-wise differences. Metrics such as the mean squared error (MSE) and the peak signal-to-noise ratio (PSNR) are analytically convenient but often align poorly with human visual perception [35]. To address this, different perceptual metrics have been proposed. The PSNR-human visual system (HVS) metric [66] uses a perceptual model to reflect the varying sensitivity of the human eye to different types of errors. Similarly, the structural similarity index (SSIM) [168] and the multi–scale SSIM (MS–SSIM) [163] metrics assess distortion based on differences in luminance, contrast, and structural content. More recent approaches employ learning-based metrics that compare images using learned feature representations to better approximate human perceptual similarity. They are commonly employed as a term in the rate–distortion objective of neural compression codecs and will be described in Section 2.3.3.

2.3.2 Conventional Image Compression

In 1992, the International Organization for Standardization (ISO) and the International Telecommunication Union (ITU) approved the JPEG standard [161] as an international standard for image processing [79, 81]. Despite being around for more than 30 years, JPEG remains the most prominent conventional compression format on the web [44, 150]. For this reason, it is also most relevant to forensic analysis and thus the focus of this thesis. Figure 2.3 depicts the components of its pipeline.

Pre-processing The first step in the JPEG pipeline involves pre-processing operations. A property of the human visual system is that our eyes are more sensitive to changes in brightness (luminance, denoted Y) than to changes in color (chrominance, denoted Cb and Cr) [140, p. 221]. To exploit this, codecs apply color space conversion and map the input image from the red, green, and blue (RGB) into the luminance-chrominance color space ($YCbCr$). In this color space, the luminance information can be separated from the chrominance information, which allows for color subsampling. Typically, color subsampling reduces the resolution of the chrominance information while keeping the full resolution of the luminance information. The resulting image data is then divided into non-overlapping 8×8 pixel blocks.

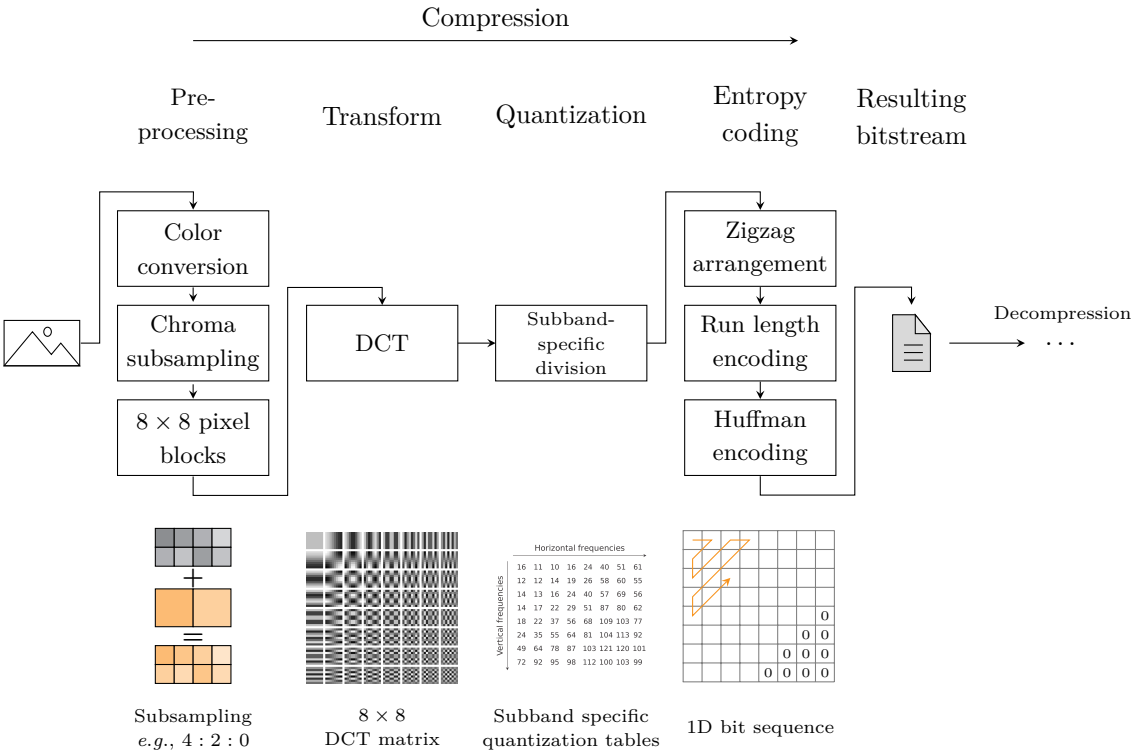


Figure 2.3: The JPEG compression pipeline. The lower row visualizes a typical partitioning pattern of chroma channels, the 2D base cosine functions used during DCT, a base quantization table that is scaled according to the given quality factor, and how quantized DCT blocks are entropy coded.

Transform The second component is transform coding. Each pixel block undergoes a 2D DCT [5], which maps the spatial information of the block into the frequency domain using 64 base cosine functions, as depicted in Figure 2.3. The resulting DCT coefficients represent the image block's frequency content. The coefficient value in the top-left corner of the 8×8 block contains the direct current (DC) component and represents the block average. All remaining values are alternating current (AC) components, with each coefficient corresponding to a specific frequency. This transformation is suitable for the subsequent quantization step, which exploits another characteristic of the human visual system [35, p. 551]. Lower-frequency components, which contribute more to human's perception of image quality, are preserved, and higher-frequency components, which are less perceptible to the human eye, are discarded or quantized more aggressively.

Quantization The DCT coefficients are quantized using predefined 8×8 quantization tables, which reduce the precision of each frequency coefficient based on its visual importance. Quantization divides each DCT coefficient by the corresponding value in the quantization table, and rounds the result to the nearest integer. The quantization table is constructed by scaling a base table according to the user-defined quality factor. An example of a base table is shown in Figure 2.3.

Entropy Coding The quantized coefficients are then arranged into a zigzag order that converts the 8×8 block into a 1D coefficient sequence, as depicted in Figure 2.3. The algorithm starts from the top-left corner and progresses toward the bottom-right of the block, which contains the higher

frequency AC coefficients. During quantization, high-frequency coefficients are often reduced to zero. The zigzag arrangement therefore yields long sequences of consecutive zeros, called zero runs. In JPEG, entropy coding combines run length encoding (RLE) of zeros and Huffman encoding. For each non-zero coefficient, the number of preceding zeros and the bit-length of the coefficient value are first determined. These two quantities are then packed into a so-called control byte, with the upper four bits containing the coefficient bit-length and the lower four bits containing the zero-run length. The control byte itself is then Huffman encoded, and the coefficient value is appended using variable-length encoding as specified in the standard. The resulting coefficient sequence consists of alternating Huffman-encoded control bytes and variable-length coefficient values. The codes assigned Huffman codes are defined in Huffman tables. They are included in the metadata of the JPEG files. Entropy coding is done separately for DC and AC, as well as Y, and Cb and Cr coefficients. Therefore, the JPEG file contains four Huffman tables. The entropy coding description above refers to AC coefficients. We refer the reader to the JPEG standard [161, Sect. 4.3] for details on the process of DC encoding, which differs slightly.

Finally, the resulting bitstream is stored together with the metadata in the file, according to the file syntax defined in the JPEG file interchange format (JFIF) [78]. The metadata includes, among other information, the quantization, and Huffman tables.

2.3.3 Neural Image Compression

Figure 2.4 depicts the neural compression pipeline. It uses the same basic compression steps, transform, quantization, and entropy coding, but replaces some operators with trained, non-linear neural networks. With new operators comes new terminology: In the neural compression literature, "encoding" or "analysis" refer to compression. "Decoding", "synthesis", or "reconstruction" are used for decompression.

Transform The analysis transform is implemented by an encoding network that maps the input pixels into a latent representation. Learning the transform has shown to isolate irrelevance in the input signal in the latent space better than with conventional linear transformations, such as DCT [12]. While the networks have shown to derive basis functions similar to those in linear transforms [45], nonlinear transforms offer better adaptation to varying data distributions [15] and can be optimized for specific distortion metrics.

Quantization The quantization in neural compression is controlled by scaling the last layer of the transform network and thus learned [12, 155]. Unlike JPEG, neural compression does not use and transmit quantization tables. Changing the scale factors post hoc is non-trivial, as the specific contribution of features for the reconstruction in the latent space is unknown. Therefore, neural compression codecs commonly require a separately trained model for each target quality.

Entropy Coding A drawback of this learned transform is that the statistical properties of the latent space might differ between trained networks, and making assumptions about these properties can lead to inefficient encoding or incorrectly reconstructed images. Neural compression commonly employs arithmetic encoding, which requires the probability distribution of symbols in the quantized latent representation. This is implemented with a parametric model. The distribution parameters for this model are predicted by a learned construction called scale hyperprior. The latent variables of this prediction model must themselves be quantized and transmitted to the decoder to enable

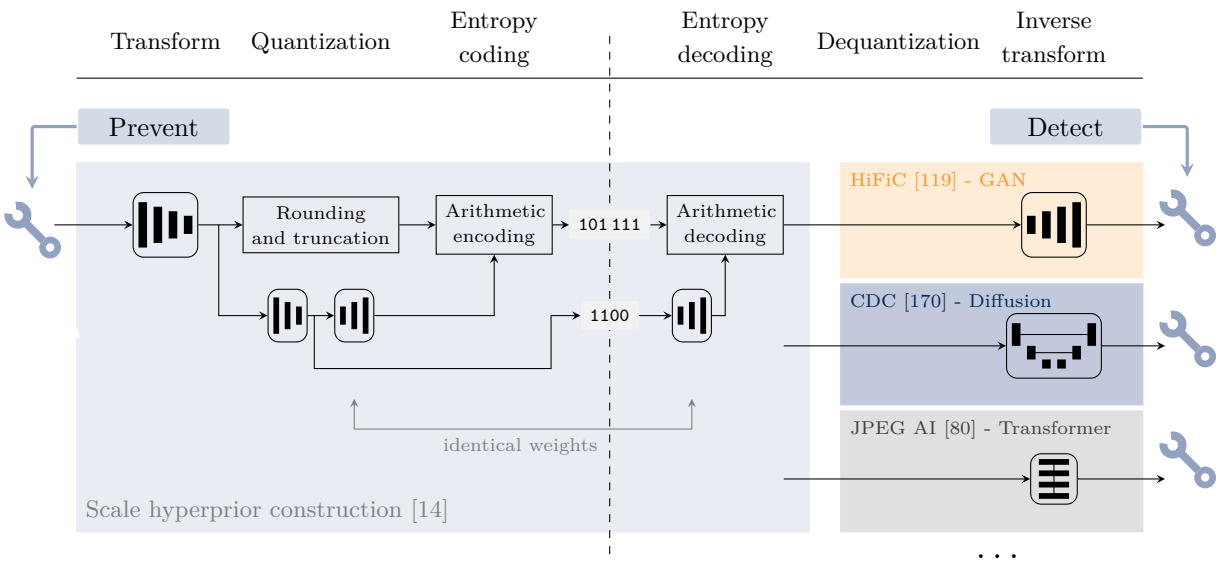


Figure 2.4: The neural compression pipeline. Most current codecs are based on this encoding construction, which uses a scale hyperprior for entropy coding. They commonly differ by the implementation of the inverse transform networks. We depict three examples from the literature. Blue wrenches refer to locations where miscompression mitigation approaches could be applied, see the outlook in Section 3.4.

reconstruction, as depicted in Figure 2.4. This entropy coding construction using the scale hyperprior was introduced by Ballé and colleagues [14] and constitutes the basis of most current neural compression codecs in the literature.

Once trained, the weights are stored in the compression and reconstruction networks. All steps are trained end-to-end and used in inference mode in the final codec. Note that this structure assumes deterministic behavior when compressed data is transmitted between a compression and a reconstruction network that run on different machines.[1]

Rate–distortion Tradeoff The networks of the whole pipeline are trained together by minimizing a rate–distortion objective. By weighting these two terms, different tradeoffs between bitrate and reconstruction quality can be achieved. Commonly used distortion metrics in neural compression include the SSIM and MS–SSIM, as well as learning-based perceptual distances such as the learned perceptual image patch similarity (LPIPS) [174] and the perceptual information metric (PIM) [28]. These metrics compute distances based on outputs of intermediate layers of pretrained networks to capture learned features such as edges, textures, and higher-level structures [175]. Although computationally more expensive, they have been shown to learn representations that correlate better with human's perception of image quality. However, increasing the weight of the perceptual metric gives the network flexibility to deviate from the input signal. This may result in deviations between the input image and its generated reconstruction.

[1] Differences in the encoder and decoder probability distributions may result in incorrectly reconstructed images, as observed in other domains [1]. However, achieving strict determinism in practice can be difficult with modern machine-learning architectures [149].

Codec Implementations in the Literature The first implementation of learned image compression in the literature used recurrent neural networks (RNNs) [156]. Since then, different architectures have been proposed. We refer the reader to the extensive review by Yang et al. [171] and continue with a brief overview of codecs used in the research papers of this dissertation. Commonly, codecs differ from Ballé et al.'s scale hyperprior construction in the implementation of the reconstruction. A prominent implementation is the GAN-based high fidelity compression (HiFiC) codec [119]. During training, the codec learns a probability model for the quantized latents for entropy coding. A generator network, which acts as the reconstruction model, synthesizes images from the decoded latents. The network is trained jointly with a competing discriminator network using an adversarial loss. The encoded side information from the hyperprior is used to improve the conditional probability estimates of the generator network. This adversarial training incorporates a perceptual realism term in its rate–distortion tradeoff objective. The conditional diffusion compression (CDC) codec [171] takes a similar approach and replaces the GAN-based generator with a diffusion model. This codec learns a conditional generative sampler that models the reconstruction distribution and iteratively samples the image conditioned on the given decoded latents. Other codecs use the attention mechanism with transformer networks [159]. This allows them to better model long-range dependencies in both the image representation and the entropy model. Examples of transformer-based codecs include the symmetrical transformer (STF) codec [176] and the JPEG AI reference implementation [80]. STF uses windowed self-attention in all three, the encoder, hyperprior, and reconstruction model. The JPEG AI codec uses the attention mechanism only for the reconstruction.

2.4 Forensicability of Compression Implementations

Forensic methods applied in practice might deal with images whose acquisition and post-processing histories are unknown. When developing such methods, researchers must inevitably make assumptions about these operations and typically base them on existing standards. This section connects forensics literature and image compression in practice. While differences between compression implementations can present opportunities, if they are well understood, they can also incur risks if practitioners are unaware. In this section, we review related work on both scenarios. As neural compression is still an emerging field, only few publications on its forensicability exist. The primary focus of this section is therefore JPEG forensics.

2.4.1 Traces of JPEG Encoder Implementation Differences

Since its standardization in 1992, JPEG is widely supported by imaging applications [76], and has been implemented in many different codecs. The most prominent codec might be the open-source C library *libjpeg* [77], which was released as the reference implementation of the JPEG standard. Given the flexibility that the standard offers for encoding, multiple different codecs, adopted for specific application requirements, have been launched over time. Many of then build on *libjpeg*. For example, *libjpeg–turbo* [105] improves compression performance through single instruction multiple data (SIMD) acceleration. *Mozjpeg* [121], a fork of *libjpeg–turbo*, is designed for web publishers and uses the progressive mode, as well as optimizations that reduce file size for the same image quality. As a consequence of the JPEG standard's flexibility, not every instance of a JPEG codec produces the same output for a given input, and even different versions of the same codec can produce differences. This has been shown by Beneš et al. [19] in a systematic comparison of compression and decompression outputs of different JPEG codec versions (all *libjpeg* versions between 6b and 9e, and *libjpeg–turbo* version 2.1.0) for a range of parameter settings. While such subtle differences might

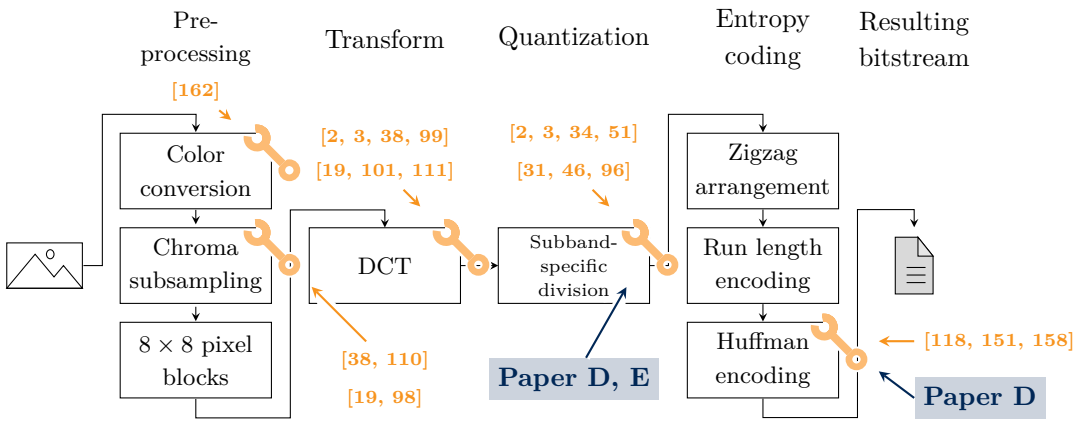


Figure 2.5: Locations in the JPEG compression pipeline where implementation differences can lead to output differences that affect signal-based forensic methods.

not be perceptible to the human eye, and thus not relevant in the majority of applications, they may affect methods in image forensics that rely on subtleties of acquisition and processing operations [18]. We now progress along the JPEG pipeline depicted in Figure 2.5 and review locations where related work has documented such forensically relevant differences. We focus on related work in signal-based forensic methods, and illustrate the research gaps that our Papers *D)* and *E)* address.

Color Conversion To convert an image from the *RGB* into the *YCbCr* color space, JPEG uses linear matrix operations with constants that approximate human perception. Specific constant values are suggested in the JPEG standard [161]. In practice, however, codecs might use different constants, integer arithmetic, and rounding, which introduces small errors. While we are not aware of any work identifying this as a risk for forensics, these artifacts have been used as traces for the detection of double JPEG compression of color images with consistent quantization tables [162].

Chroma Subsampling Block convergence is a method that estimates the number of prior JPEG compressions by repeatedly recompressing an image and measuring the behavior of pixel blocks [99]. Carnein et al., who extend the method from grayscale to color images, document differences for different *libjpeg* versions [38]. They pinpoint different chroma sub- and upsampling methods as the source of the distinct block convergence behaviors and demonstrate that these differences can be exploited to identify which *libjpeg* version was used.
Lorch and Riess present a novel artifact for signal-based forensic methods called chroma wrinkles [110]. Chroma wrinkles are subtle, periodic chroma variations caused by integer rounding and bit-shifting operations during the averaging of pixel values in chroma subsampling. They cause every second column of the compressed chroma channel to appear slightly brighter than its neighboring column. Chroma wrinkles exist in images from *libjpeg* version 6b, which has been the de-facto default JPEG codec for image processing applications for about one decade from 1998 to 2009 [19, 150]. From version 7 onwards, *libjpeg* resolved chroma wrinkles by changing the subsampling implementation to DCT-based scaling. However, these artifacts were migrated to *libjpeg–turbo*, which forked out of *libjpeg*, version 6b. *Mozjpeg* also introduces chroma wrinkles in image compressed with quality factor below 75 [110]. Lorch et al. [110] exploit these patterns in the DCT domain as a trace to distinguish between *libjpeg* versions and even to detect local tampering.

Beneš et al. [19] confirm these previous findings about differences in chroma sub- and upsampling and attribute them to the specific sampling algorithm. *Libjpeg* version 6b and *libjpeg–turbo* use the older, so-called *simple* and *fancy* spatial domain sub- and upsampling method, while the versions 7 and above use *DCT-based* scaling, which behaves differently whenever the subsampling factors are powers of two. Furthermore, they find that version 9e diverges further under certain subsampling conditions. For decompression, they find similar clustering with an additional difference between version 6b and *libjpeg–turbo* caused by the turbo-specific optimization in the *fancy* upsampling method for one specific sampling factor. These differences are forensically relevant and can, under specific circumstances, contribute to the cover-source mismatch problem in steganalysis [18].
Kumawat and colleagues [98] investigate forensic methods that use the cyclostationarity in decompressed color images, which is introduced during chroma upsampling. The authors evaluate different software and in-camera JPEG implementations and note that differences in decoder-specific chroma upsampling algorithms might affect the reliability of forensic methods.

DCT Implementation  Many improvements were developed since DCT was first implemented for digital image processing in 1974 [5]. Optimizations of *libjpeg*'s default implementation *DCT-slow* [107], also called *DCT-integer*, focus on computation complexity (*e.g.*, *DCT-fast* is faster but less accurate [131, p. 52]) and the type of arithmetic precision (*e.g.*, *DCT-float*). These different DCT implementations can introduce traces in the output signal, which can be forensically exploited. Agarwal et al. call these traces "JPEG dimples" and use them for encoder identification [3] and even to detect local manipulations [2].
However, artifacts from different DCT implementations may also affect the results of existing forensic methods, if they are sensitive to subtle signal differences and do not expect such artifacts. Lai et al. and Carnein et al. show that different DCT implementations in the *libjpeg* codec affect the convergence times of JPEG blocks under repeated recompression in grayscale [99] and color images [38]. Lorch et al. [111] investigates learning-based double-JPEG compression detectors that rely on traces such as blocking artifacts and periodicity of DCT coefficients. They demonstrate that differences in the image signal, which result from a mismatch in DCT implementation of training and test images (DCT-slow vs. DCT-fast), represent out-of-distribution samples and can cause these detectors to fail. Levecque et al. [101] propose a compression codec detector based on the compatibility of observed traces in the image with known traces of specific codecs. If the detector made wrong assumptions about the applied DCT implementation, it can no longer attribute the image to the correct codec. Beneš et al. [19] even find (small) differences in the compression outputs for the same DCT implementation in different codec versions.

Quantization  Annex K of the JPEG standard [161] provides two example base quantization tables, one for luminance, and one for chrominance, which were the result of psychovisual experiments conducted in the 1980s [108]. During JPEG compression, the base tables are scaled for the selected quality factor and stored in the metadata of the JPEG file. While the base tables from the standard are a popular choice, codec developers may customize them. When a forensic investigator extracts such a custom table, they can be used to reconstruct the processing history of the image under investigation [51, 96]. Another source of variation in quantization is the rounding operation. The JPEG standard's default recommended method is rounding to the nearest integer, however, variations thereof have been implemented among encoders including flooring, ceiling, and truncation. These variations produce systematically different quantized DCT values, even when the same quantization matrix is used [31, 34]. Albeit small, the traces persist in the encoded DCT

coefficients as well as the decoded pixel values. By measuring how an image differs from versions recompressed with controlled quantization, these artifacts can be exploited as a codec fingerprint [31], as well as forensic method to detect manipulations [46].
However, quantization implementation differences may also affect forensic methods. *Mozjpeg* implements an encoding optimization, called trellis optimization, which introduces subtle differences in the quantized DCT coefficients. Trellis optimization is explored in depth in Paper *D)* and detected in Paper *E)* of this thesis. The methods and findings of the papers will be described in detail in Chapter 4.

Entropy Coding Huffman tables are included in the JPEG header and can be chosen freely by the encoder. One option is to construct an optimal table, based on the respective image content. While this results in shorter bitstreams, the table construction might be time-consuming and, in practice, many codecs use the standard Huffman tables, which are predefined in Annex K of the JPEG standard [161]. Entropy coding is a lossless compression operation, therefore different implementations do not change the decoded pixels, but may produce different entropy coded bitstreams and header structures. Forensics uses Huffman tables, *e.g.*, for file recovery [151, 157], or to fingerprint for the encoder implementation if it uses consistent tables that differ from those described in the JPEG standard [122]. However, the use of custom tables can affect forensic methods for file recovery if they assume standard tables [151, 158]. McKeown et al. [118] exploit the use of optimized Huffman tables and describe a method to group JPEG images from with similar content.
The JPEG standard defines four compression modes. The two commonly used modes are the baseline sequential, and progressive mode. They differ by the order in which the quantized DCT coefficients are stored in an image. In the sequential mode, the image is encoded and decoded sequentially along the scan line, according to the rotation flag set in the image metadata [109]. In contrast, the progressive mode partitions the quantized DCT coefficients into multiple scans, as specified in the scan script, which is stored in the file header. The partitioning uses two mechanisms: spectral selection, which groups coefficients by frequency, and successive approximation, which groups them by precision (bit planes). Typically, low-frequency components and more significant bits are assigned to earlier scans and high-frequency components and less significant bits are assigned to later scans. This allows a decoder to display a coarse version of an image first and refine it progressively. In Chapter 4, we describe how custom scan scripts can be exploited for the attribution of an image to a social network platform (Paper *D)*).

2.4.2 Forensicability of Neural Compression Implementations

In 2019, the JPEG standardization committee launched the JPEG AI Project and developed a standard for learning-based image compression which was accepted and published in 2025 [80, 82]. We refer the reader to the paper by Esenlik et al. [49] for a systematic overview. As far as we know, JPEG AI has not yet been implemented in any user devices. However, members of the standardization committee have given keynotes at different conferences that indicate their intention to do so. In their talks, they presented proof of concepts of next generation mobile phones that support the JPEG AI standard [6, 10].
Nevertheless, neural compression is an emerging field and the digital image forensics community has just begun to address its implications. Berthet et al. [26, 27] were the first to show that signal-based forensic methods developed for conventional JPEG images do not transfer to neurally compressed images and cause misclassification. They analyze methods for copy-move forgery detection [26] and source social network identification [27] on neurally compressed images. Others have shown that

the performance of downstream computer vision tasks might degrade when operating on neurally compressed images [23, 83, 115], and that learning-based forensic methods for deep fake detection are similarly affected [36, 139]. By contrast, Cardenuto et al. [37] report that the evidential value of medical images in science does not deteriorate at an equal bitrate compared to conventional compression. The literature on forensic methods tailored to neurally compressed images is still sparse. Bergmann et al. develope approaches for identifying the use of HiFiC and JPEG AI by analyzing characteristic traces left in the image [24], including frequency [21] spatial information [22]. Beyond passive traces, neural compression has also been shown to affect the robustness of active forensic signals such as digital watermarks. Yakubenko and Gashnikov [169] report that spatial-domain embedding schemes, *e.g.*, LSB, are largely destroyed by neural codecs, whereas transform-domain approaches exhibit higher resilience.

3. Contributions Towards Neural Compression Forensics

In 2013, David Kriesel exposed a bug in the firmware of professional multifunction printers that caused the integrated scanners to unpredictably modify digits in scanned documents. The issue originated in the scanners' firmware, which used Joint Bi-level Image Experts Group 2 (JBIG2) compression [75] to reduce the file size of PDF documents. JBIG2 achieves compression by first segmenting a document into patches of pixels. Visually similar patches, such as character glyphs, are then matched to a shared representative symbol that is reused throughout the document. Due to a bug in the scanners' implementation of this pattern-matching algorithm, the scanners occasionally matched visually similar, but semantically different patches, substituting them at the pixel level. The results were visually coherent, but incorrectly substituted details, primarily digits, in scanned documents as shown in the example in Figure 3.1.

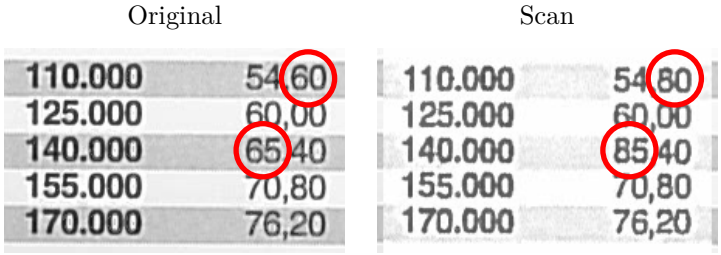


Figure 3.1: An example of a JBIG2 scanner bug, where the number 6 in the original document was incorrectly substituted with the number 8 in the scan. Images are reproduced from David Kriesel's blog article,[1] with kind permission from the author.

In this chapter, we summarize the research results of Papers *A)*, *B)*, and *C)*, in which we introduce and analyze what we term "miscompressions". Miscompressions are artifacts in neurally reconstructed images in which the reconstructed content remains visually plausible but is semantically different from the input image. Unlike the JBIG2 scanner bug, these artifacts do not stem from an isolated implementation error that can be corrected once identified. Instead, they originate from learning-based operators in neural codecs, which reconstruct images based on data-driven priors. Consequently, miscompressions constitute a systematic limitation of learning-based image compression, and there is currently no reliable way to detect or prevent them. Moreover, miscompressions extend beyond repeated digits in scanned documents and can occur in any images of arbitrary content. They may alter the attributes of depicted objects, *e.g.*, the color or shape, or even remove or introduce new details. As the reconstructed images commonly remain visually plausible, mis-

compressions can easily go unnoticed and may affect human interpretation, as well as automated downstream image processing.

In this chapter, we outline our contributions toward reliable neural compression forensics. Section 3.1 describes the research problem, and Section 3.2 summarizes the content of the three research papers. In Section 3.3, we discuss important results, before we highlight open problems and suggest directions for future research in Section 3.4.

3.1 Problem Description

By leveraging generative networks, neural compression codecs reconstruct images with high perceptual quality at low bitrates. They achieve this by substituting image regions that are expensive to encode with visually similar, "cheaper" representations. If this affects pixels which are critical to the semantic meaning of the image or image detail, such substitutions may introduce unintended and unnoticed image manipulations. Figure 3.2 shows an example of a miscompression in which, similar to the JBIG2 scanner bug in Figure 3.1, the digit 6 is incorrectly reconstructed as an 8.

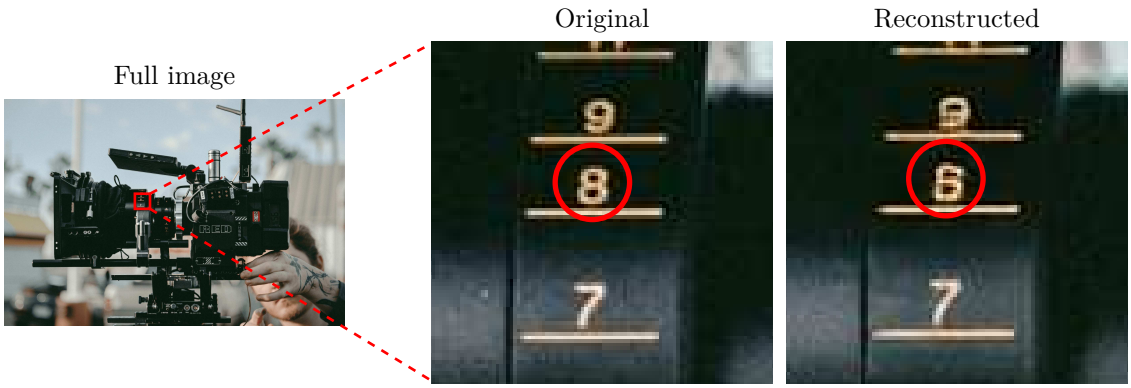


Figure 3.2: Neural compression artifacts may change the semantics of an image or image detail ("miscompression"). Here, the shape of the number 8 in the original image has changed in the reconstruction such that it could be mistaken for the number 6. The 64×64 pixel crops constitute 0.15% of the full image (2040×1365). The image is from the DIV2K dataset [4] and was compressed using HiFiC Lo [119] at 0.09 BPP.

Given the fundamentally different operators in the neural compression pipeline, it is unsurprising that signal-based forensic methods developed for conventional JPEG do not transfer to neural compression [22, 26, 27]. Less obvious, however, is the possibility that neural compression may undermine scene-based forensic methods that rely on semantic as well as physical traces. At present, these concerns remain speculative, as neural image compression is still in an early stage and has not yet been deployed in practice. However, properties of emerging compression algorithms warrant study and miscompressions, if left unaddressed, may constitute a future pitfall, even beyond forensics. In our papers, we define, describe, and demonstrate miscompressions, discuss their potential implications, and lay the groundwork for research on mitigation strategies.

3.2 Summary of Research Papers

This section summarizes the three research papers *A)*, *B)*, and *C)*. We place the papers in the context of this dissertation, describe the applied methods, and summarize the main results. Finally, I provide personal contribution statements. The percentages given in parentheses refer to subjective quantitative estimates of my contribution along three dimensions, listed in the following order: (i) idea and conception, (ii) operational execution and analysis, and (iii) writing. The framing of the individual papers was tailored to their publishing venues. Potential implications of our findings for image forensics will be discussed here in Section 3.3, while implications for society at large as well as future research to handle the apparent risks are addressed in the main papers in Part II.

3.2.1 Paper A*)*: Taxonomy of Miscompressions

N. Hofer and R. Böhme. A taxonomy of miscompressions: Preparing image forensics for neural compression. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2024 *(Workshop, no CORE ranking available)*

Context Neural networks employed in current codecs are optimized to selectively discard information while targeting what humans perceive as visually compelling quality. While authors of current codecs commonly acknowledge that their generators may not only reconstruct images of high perceptual quality but can also *"produce images that are very different from the input"* [119, p. 10], the characteristics, prevalence, and implications of these differences remain unknown. In this paper, we define and describe this new type of compression artifact that is specific to neural compression and may alter the semantics of an image or image detail. We coin a new term, introduce a formal definition, and propose a taxonomy that is intended to facilitate discussion among the image forensics and compression research communities, forensics practitioners, and the general public.

Method Our approach is exploratory. We focus on five state-of-the-art neural compression codecs covering relevant compression architectures, namely variational auto encoders [14], transformer-based models [80, 176], a GAN [119], and a diffusion model [170]. Using reference implementations provided by the respective authors, we deploy the codecs on a GPU cluster and compress and reconstruct around 500 test images drawn from three widely used benchmark datasets with different quality settings. We then use the VPV image viewer software[2] and manually inspect the image pairs to detect semantic differences between the originals and reconstructions. From a collection of observed cases, we derive a definition of the phenomenon and develop a taxonomy based on the apparent transformation of the signal.

Results We coin the term *miscompressions* and define them as "reconstruction errors that occur when there is a discrepancy between the semantic meaning of an original image (detail) and its reconstructed version after neural compression." [71, p. 3]. Our proposed taxonomy distinguishes three categories describing "what happens" (changes in *amplitude*, *geometry*, and *shape*), and includes an additional binary *high impact* flag to indicate miscompressions that alter symbols. Figure 3.2 in the previous Section 3.1 illustrates an example of a *high impact* miscompression of the category *shape*. Other examples of miscompression include color changes, such as a purple bag turning blue, darkened skin tones and changed eye colors, and objects that appear or disappear, such as additional

[2] `https://github.com/kidanger/vpv`

antennas on buildings, people in the background vanishing or birthmarks disappearing. We observe miscompressions in images compressed and reconstructed by all five codecs.

Contribution (80% / 90% / 60%) I was the leading contributor of this paper and responsible for the idea, and the setup of the infrastructure to execute the neural compression codecs. I also did the manual inspection of reconstructed images and the identification of miscompressions. Rainer Böhme contributed to this paper by developing the taxonomy and supported the write-up, visualizations, and the framing of the work within the broader context of image forensics.

3.2.2 Paper B*)*: User Perceptions of Miscompressions

N. Hofer and R. Böhme. When the codec hallucinates: User perceptions of miscompressed images. In *CHI Conference on Human Factors in Computing Systems*. ACM, 2026. to appear *(Conference, CORE 2023 A* ranking)*

Context The semantic meaning of an image or image detail is inherently subjective and influenced by the context, as well as the observer's experience and cultural background. So far, claims that neural compression artifacts may change image semantics are based on our subjective perception. We conduct a user study and collect empirical evidence of how miscompressions are perceived by a broader field of people with three hypothesis (H). Specifically, we want to understand whether humans perceive miscompressions as increasing risk of misunderstandings (H_1), misinterpret miscompressions as intentional editing (H_2), and correctly identify them as unintentional distortion caused *e.g.*, by image compression (H_3).

Method We conduct a controlled lab user study with 115 participants. Because miscompressions are defined as differences between images, we employ a full-reference design in which participants compare an original image with a test image displayed side-by-side. We frame the study as a hypothetical social network scenario and ask participants to compare an image they had taken and uploaded themselves (original) with an image depicting the same scene as shown to a friend on a different social network platform (test image). Each participant evaluates twelve image pairs, comprising six neurally compressed images containing miscompressions, two neurally compressed images *without* miscompressions, three JPEG compressed images, and one uncompressed attention-check image. To produce the stimuli we use five current neural codecs and compress and reconstruct the images. From this set we collect 19 scenes through manual inspection to ensure diversity in content and miscompression types (changes in *amplitude*, *geometry*, and *shape*). The selection allows us to generalize from the semantic influence of the individual context and characteristics of a scene. Participants are divided into four groups. The study follows a mixed within- and between-subject design, with fixed scene assignment per group and randomized presentation order within groups, resulting in a total of 1 380 image views. For each image pair, participants first indicate whether they see any difference between the two images. If so, they rate how certain they are that these differences could lead to misunderstandings between them, only seeing the original image, and a second person only seeing the test image. Next, they rate how likely they would attribute the image difference to intentional editing (*e.g.*, retouching, filters, or manipulation), and uncontrollable distortion (*e.g.*, transmission errors or image compression). The ratings are done on 6-point item scales ranging from "certainly" to "certainly not". We evaluate our hypotheses using ordinary least squares linear regression on participants' ratings of image pairs. The models compare miscompressed and JPEG

compressed images against neurally compressed images without miscompressions, while controlling for subject and scene-specific effects using fixed effects. We estimate different model specifications for three outcome variables: H_1, the perceived risk of misunderstanding; H_2, attribution to intentional editing; and H_3, attribution to uncontrollable distortion.

Results We find empirical support for all three hypotheses, showing that our perception of miscompressions transfers to a larger group of people. Specifically, on the 6-point scale, participants report an increased risk of misunderstandings of almost one scale point closer to "certainly" for miscompressed images, after controlling for all scene and subject-specific variation. Similarly, participants attribute the differences to intentional editing one point closer to "certainly", and to uncontrollable distortion approximately 0.7 points closer to "certainly not". To better interpret the difference of steps on our 6-point scale means for the noise level in our data, we calculate Cohen's d as a measure of effect size [40]. For misunderstandings (H_1) we observe a "large" effect with $d = 0.86$. For the other dependent variables we obtain "moderate" effects of $d = 0.78$ (H_2) and $d = 0.64$ (H_3), respectively.

Contribution (60% / 70% / 60%) I was the leading contributor to this paper and responsible for the study concept and setup including the selection and preparation of stimuli. I implemented the instrument, was responsible for the application for the IRB approval, and the execution of the study. Moreover, I did a careful review of related work, spanning human-computer-interaction and user studies in image compression literature. Rainer Böhme supported me with the instrument design, statistical analysis of the results, and the write-up and visualization. Judith Senn, Thomas Filips, Verena Lachner and Dennis Sommer supported the data collection by conducting the study with students in their seminar groups. Max Ninow supported us with the setup of the infrastructure.

3.2.3 Paper C*)*: A Research Dataset of Miscompressions

N. Hofer and R. Böhme. Challenging cases of neural image compression: A dataset of visually compelling yet semantically incorrect reconstructions. In *International Conference on Multimedia*, pages 13318–13324. ACM, 2025 *(Conference, CORE 2023 A* ranking)*

Context Miscompressions may also have consequences that go beyond high-stakes contexts, such as forensic investigations. For this reason, research should strive to identify implications and find ways to mitigate miscompressions. In this paper, we pave the way and introduce a large, publicly available dataset of human-annotated miscompressions spanning multiple neural codec architectures and quality metrics. The dataset is available, documented and ready to use for future research.

Method The dataset design follows four key objectives: sufficient scale to train large networks, coverage across multiple codec and compression rates, diversity in scenes and imaging conditions, and reproducibility for future research. We select 1 563 images from three public benchmark datasets, pre-screen, and preprocess them to ensure content diversity and compatibility with all selected neural compression codecs. We employ six neural codecs, covering a range of architectures and optimization metrics. Each codec compresses and reconstructs images at two target quality levels: a low-quality setting, targeting 0.25 BPP, and high quality setting, targeting 0.75 BPP. We then assign each image to one of three trained human lablers and task them to annotate miscompressions. The labelers receive an introduction to neural image compression and the concept of miscompressions. The training includes qualitative in-depth discussions of example images. To support consistent annotation

decisions, we design a decision tree guiding labelers through four steps: identifying visible changes in objects or image areas, assessing the semantic relevance of the change, determining whether the change would be noticeable without access to the original image, and evaluating whether the change corresponds to expected compression artifacts. Labelers compare original and reconstructed images and mark image areas with altered semantics by drawing bounding boxes using the VPV Software [8], which we extended with functionality to record annotation coordinates. Annotation progress and image assignments are coordinated in a central progress document. Finally, we measure inter-labeler agreement using Krippendorff's α [97, p. 211–256] and agreement scores at different units of granularity per image.

Results The resulting semantic changes of learning-based image compression (SCLIC) dataset [73] contains 18 019 human-annotated miscompressions identified across reconstructions generated by six neural compression codecs at two quality levels. The dataset is publicly available via Zenodo (record no. 16780952 [3]) and includes 1 563 original, uncompressed images and their respective compressed and reconstructed versions. Annotated miscompressions are provided in a CSV file as the coordinates of bounding boxes around miscompressed objects. The dataset further includes scripts that facilitate the reuse and further analysis of the images and miscompressions. On average, and across codecs, approximately one in two images contain miscompressions at the low quality setting, with a mean of 4.37 annotations per affected image. At the high-quality setting, these values decrease to approximately one in five miscompressed images and an average of 2.24 annotations. Prevalence varies between codecs: HiFiC exhibits the highest susceptibility (69.2% of images are affected at low quality), followed by JPEG-AI (64.5%). Of the two variants of the hyperprior codec, one optimized for the mathematical metric MSE and one optimized for the perceptual MSE-SSIM metric. At high quality, perpetual optimization leads to a threefold increase in miscompressions compared to MSE optimization (15.2% vs. 3.2% of images). A notable special case of miscompressions is multi-miscompressions, where multiple instance of similar objects are miscompressed in the same way, such as consistent color changes across several window shades of a building. Multi-miscompressions occur in approximately one in three miscompressed images for low quality and one in five at high quality. Inter-labeler agreement ranges from 58% to 97%, depending on the unit granularity. Krippendorff's alpha is between 0.29 and 0.48, indicating agreement above chance level.

Contribution (80% / 80% / 60%) I was the leading contributor of this paper and responsible for the concept, the execution, and the write-up. I have also applied for and received the funding to employ the three labelers from the Tiroler Nachwuchsforscher*innenförderung. Rainer Böhme supported me with the extension of the VPV software [8]. He also contributed to the write-up and statistical analysis. Leny Barry, Valerie Huter, and Max Ninow helped us with many hours of inspecting images an annotating miscompressions following the provided guidelines.

3.3 Summary and Discussion of Results

Isolated exception aside, so far, image compression has been regarded as a processing operation that, while providing a rich source of forensic traces, does not impair image authenticity. This assumption no longer holds for neural image compression. Current codecs can generate miscompressions, *i.e.*, compression artifacts that alter the semantic meaning of images or image details. Miscompressions appear to be a general problem of neural image compression, as they occur in all compression codecs we tested, including the reference implementation of the recently standardized JPEG AI algorithm.

[3] `https://zenodo.org/records/16780952`

Our findings suggest that miscompressions may have implications for digital image forensics and beyond. At present, people are not familiar with neural compression and do not expect compression artifacts that induce semantic changes. Moreover, miscompressions are perceived to increase the risk of misunderstandings between a person viewing the original version of an image, and a second person viewing a miscompressed version of the same image. People also tend to misattribute miscompressions to intentional editing and do not recognize them as uncontrollable distortions from image compression. The prevalence of miscompressions emphasizes the need for research on mitigation approaches: On average, one in three images contains at least one image detail that had a different semantic meaning before image compression. If neural compression becomes widespread in everyday applications without the users' knowledge, there is a risk that individuals may place unwarranted trust in unauthentic images, inadvertently spread misinformation, or misinterpret compression artifacts as intentional editing, potentially leading to false accusations of manipulation.

Loss functions that are optimized for learned perceptual metrics can produce visually convincing reconstructions and observers cannot recognize whether an image contains compression artifacts. Conventional artifacts, such as blurring or pixelation, therefore cease to serve as intuitive indicators of whether an image was modified by compression and should be trusted.

Special precautions will be required in the future, if images that have undergone neural compression are under investigation and analyzed with existing forensic methods in practice. Our findings suggest specific risks for scene-based forensic methods, which may account for known compression distortions, but might not assume semantic changes. Miscompressions may therefore lead to incorrect conclusions, for instance if forensic methods rely on physical traces mentioned in Section 2.2.1 or the geometry, texture, size, color, or shape of objects. Forensic methods that rely on semantic cues from the image content might be likewise affected. For example, a miscompression that changes the color of a bag may have serious consequences in investigative contexts such as person identification. Tool-based approaches, including face recognition or license plate recognition systems, may also be affected. In our research, we observe that faces are frequently distorted or altered, for instance appearing as a different age or gender. Qiu et al. [138] even report that the semantic distortions are subject to racial bias. African–American faces are often reconstructed to appear more Caucasian, while Caucasian faces largely retain their original features.

3.4 Open Problems and Future Work

Like any research, our studies have limitations. Detailed limitations are discussed in the respective papers in Part II and will not be repeated here. Instead, this section briefly reflects on decisions and design choices in a broader context and discusses open research problems.

Reflection on Codec Selection Neural image compression is still an emerging technology, and investigating its impact for forensics may therefore appear premature. However, the standardization of JPEG AI indicates growing interest from the industry and suggests that the deployment in consumer devices may be approaching. It is thus important to monitor these developments and to anticipate potential risks for both society and forensics. The selection of codecs that we analyzed in our studies was influenced by (i) the assumption that the JPEG AI reference implementation currently represents the most realistic candidate for future deployment in consumer devices, (ii) the objective of covering a broad range of neural compression architectures from the literature, and (iii) the availability of implementations and pretrained model weights, which enables us to use the codecs as intended by their respective authors.

Future Work to Investigate Forensic Implications of Miscompressions Norman et al. [124] investigate the impact of super-resolution, learning-based motion, and optical deblurring techniques on forensic facial recognition. The authors find that "under certain conditions, and with the appropriate choice of enhancement model," [124, p. 4313] such techniques may assist forensic analyses that rely on semantic traces. Their failure cases, however, are concerning: "enhancement has introduced or distorted facial features relative to the original leading to an apparent different identity" [124, Fig. 4]. Similar issues may arise from neural image compression, specifically in cases of miscompressions that alter facial or other identifiable features of individuals. In recently published work, Bergmann and colleagues [25] investigate the implication of miscompressions for tools used in scene-based forensic analyses. They report failure cases including modified tattoos and clothing items with removed or hallucinated details as well as changes in color. The authors identify both the training loss function and the network architecture employed in the codec as influencing factors. As described in Paper *B)*, people are currently unfamiliar with neural compression and therefore do not expect compression artifacts that alter image semantics. If neural compression is introduced into everyday applications, forensics investigators and judicial personnel who assess images as forensic evidence must be aware of this technology and take the possibility of miscompressions into account. A comprehensive study should therefore examine the extent to which our findings affect the conclusions drawn by forensic practitioners when analysing images. The design of the user study in Paper *B)* could serve as a starting point for such a study. The results could help to establish guidelines for the safe handling of neurally compressed images in forensic investigations.
Furthermore, two recent studies show that signal-based deepfake detectors tend to misclassify genuine neurally compressed images as deepfakes [36, 139]. This is because the generative models that are used in the compression codecs introduce traces which are similar to the traces such detectors exploit to identify deepfakes. This suggests that compression artifacts which are not covered by our definition of "semantic changes" may nonetheless affect forensic analyses. For example, scene-based methods may rely on visible traces with no apparent semantics, such as color fringes. As an extension of the experiments conducted in [25], future work should evaluate applied forensic methods and reassess their reliability when confronted with neurally compressed images. Our SCLIC dataset can be used for such an analysis, as it provides, next to annotated miscompressions, the original images from three benchmark datasets, along with their compressed and reconstructed versions.

Future Work Toward the Mitigation of Miscompressions Technical research studies that develop mitigation strategies for miscompressions are essential if we, as a society, expect future image compression codecs to produce outputs that align with current notions of image authenticity. We use the blue wrenches in Figure 2.4, Section 2.3.3, to position potential mitigation approaches at two locations in the compression and reconstruction pipeline.
Detect: In image compression we have access to the uncompressed reference image. The intuitive step forward is therefore to build a binary miscompression detection classifier which compares the input to the output image, and differentiates between acceptable compression distortion, and unacceptable semantic changes. A potential avenue could be to use foundational models that were trained to have semantic image understanding and exploit their embedding representation to "quantify" the semantics of a given scene. A detection model could then be trained to learn a benchmark of allowed semantic distance within the embedding space between an uncompressed reference and a neurally compressed image. The annotated miscompressions in the SCLIC dataset makes this possible. As the dataset contains images compressed and reconstructed by multiple different compression codecs, it provides positive *and* negative samples of the same scene. Positive samples refer to samples that contain miscompressions, *i.e.*, unacceptable semantic changes. Negative samples

refer to samples that contain acceptable neural compression distortion. Potential examples for suitable models include CLIP-based pretrained vision transformers, as well as more recent models like Meta's DINO [106, 127] or BLIP [103].

Prevent: Detecting miscompressions in reconstructed images can inform the users and mitigate their negative impact. However, it does not address the underlying issue of altered image semantics in reconstructed images. Therefore, miscompressions should be prevented before they occur, for example by adapting compression parameters to the image content. Possible approaches include selecting encoding models optimized for different perceptual metrics or adjusting the rate-distortion tradeoff to allocate more bits to perceptually or semantically important regions. The SCLIC dataset contains all images that our labelers viewed, regardless of whether miscompressions were observed. The exploration of the dataset might reveal image characteristics or scene content that are prone to be miscompressed. The images were taken from three source datasets and cover diverse scenes and different acquisition devices. The dataset could further enable neural networks that predict miscompressions directly from uncompressed images. Such predictions could guide neural codecs that support region-of-interest coding and enable content-adaptive bit allocation to critical image areas. Preventing miscompressions during encoding is particularly important when neural compression is used in downstream machine learning tasks, where the compressed bitstream is processed directly without reconstruction. The JPEG AI standard explicitly targets this use case and defines it as a decoder requirement [11, 80].

4. Contributions to JPEG Forensics

The JPEG standard specifies minimum requirements for the key compression components to guarantee that any compliant decoder can correctly decompress an encoded image. At the same time, it offers a high degree of freedom, allowing developers of compression codecs to optimize their implementation for specific applications and use case. As a result, many different JPEG codecs exist in practice and operate within the underlying infrastructure of capturing devices, image processing software, social network platforms, messaging services, and other platforms that edit, store, and transmit images. Our literature review of the forensicability of differences in codec implementations in Section 2.4 revealed that studies are largely limited to *libjpeg* and *libjpeg–turbo*.[1] Ignoring codec-specific optimizations from industry may have been justifiable in the early 2000s, when the image forensics community emerged and the reference implementation *libjpeg* was the dominant codec in practice. However, the web has since evolved, and multiple codecs with different optimizations exist. In this chapter, we summarize the contributions of Papers *D)* and *E)*, which investigate how artifacts introduced by *mozjpeg*'s optimizations affect signal-based image forensics and demonstrate how such artifacts can be detected to mitigate the risk they might pose to forensic methods.
The structure of this chapter follows the one of Chapter 3. In Section 4.1, we introduce the problem that motivated the research and in Section 4.2, we outline the content of the research papers. Important results are summarized and discussed in Section 4.3. Finally, in Section 4.4, we highlight open problems and suggests directions for future research.

4.1 Problem Description

A possible reason for the limited research concerning *mozjpeg* may relate to its default use of the progressive mode. Although it is part of the JPEG standard since the beginning, the progressive mode has received limited attention in the literature on signal-based forensics. This may stem from its perceived lack of real-world relevance, as early browser display bugs led to recommendations against its use [167]. Notably, these recommendations were issued in 2013 [166], one year before the release of the first version of *mozjpeg*. Another factor may be that, according to the standard, the progressive mode only changes the ordering of information in the bitstream and does not alter the image signal. For instance, Butora and Bas [34], who model decompression errors to identify JPEG compression implementations for high-quality images, omit *mozjpeg* from their analysis, stating that it uses the progressive mode and "produces the same DCT coefficients as *libjpeg*" [34, p. 4]. While no study we are aware of has measured the prevalence of *mozjpeg*, its popularity on GitHub[2] indicates

[1] Exceptions exist, *e.g.*, [110].
[2] 5.7k stars and 430 forks (accessed: Jan 06, 2026)

real-world relevance. Two papers from 2020 and 2021 propose steganographic embedding methods that are robust to JPEG compression on social networks [60, 112]. They model compression changes of *mozjpeg* and claim that it has been "widely used [...] in the real world" [60, p. 4] and "in recent years" [112, p. 2917]. However, these claims are not supported by external sources and the authors did not respond to our inquiries. In this dissertation, we investigate whether the limited research on the progressive mode and the *mozjpeg* codec are still justified.

4.2 Summary of Research Papers

In this section, we summarize the papers *D)* and *E)*. We put them into the context of this dissertation, and describe the applied methods and important results. As before, I provide personal contribution statements, with percentages in parentheses indicating my subjective contributions to (i) idea and conception, (ii) operational execution and analysis, and (iii) writing, in that order.

4.2.1 Paper D*)* - Understanding *Mozjpeg*

N. Hofer and R. Böhme. Progressive JPEGs in the wild: Implications for information hiding and forensics. In *Workshop on Information Hiding and Multimedia Security*, pages 47–58. ACM, 2023 *(Workshop, CORE 2023 C ranking, Best Student Paper Award)*

Context In this paper, we collect and analyze a sample of historic image data from the web spanning the past 20 years, to study the prevalence of progressively compressed JPEG images over time. Moreover, we analyze the *mozjpeg* library and document the inner workings underlying its optimizations, thereby making them more accessible to the research community. In particular, we describe *mozjpeg*'s trellis optimization algorithm, which modifies DCT coefficients to improve a rate–distortion tradeoff. We further identify and characterize the distinctive artifacts introduced by this algorithm and quantify the magnitude of the resulting coefficient changes. Finally, we examine *mozjpeg*'s scan script optimization algorithm.

Method We employ multiple complementary empirical analyzes. To determine the prevalence of progressive JPEG images on the web, we crawl a total of about 200 000 images from two sources: the image-sharing platform Flickr, and the Internet Archive's Wayback Machine [125]. Using the latter, we crawl archived images from the 2022 Tranco list of top-5000 websites [100]. We quantify the proportion of progressive images by detecting the `SOF2` (Start Of Frame, progressive DCT) marker in the JPEG headers. To investigate the internals of *mozjpeg*'s optimizations, we analyze its source code on GitHub,[3] specifically the trellis optimization and scan script optimization algorithm. Using an instrumented version of *mozjpeg*, we log block-level rate–distortion movements and illustrate how the trellis algorithm modifies DCT coefficients during quantization to minimize coding cost. To quantify the impact of the trellis algorithm, we compress images while isolating the optimization and measure the share of modified AC coefficient values in the luminance channel across different quality factors. We also measure the compression efficiency of *mozjpeg*'s optimization algorithms. Finally, we extract and analyze scan scripts of progressive images compressed with *mozjpeg* as well as image contained in the Forchheim dataset [67]. This dataset contains images that were distributed via different social networks.

[3] `https://github.com/mozilla/mozjpeg`

Results The prevalence of progressive images on the web has increased over time, with a stronger rise after 2014, coinciding with the release of *mozjpeg*. While fewer than 15% of images on the top-5000 websites were progressive in the 2000s, the share has since doubled. Today, approximately one in three JPEG images on the web is progressive. A similar trend is observed on Flickr. Whereas less than 2% of images were progressive in the 2000s, the share has tripled by 2020.
Trellis-based optimizations, originally proposed for video codecs [164], address the rate–distortion problem by identifying a path through a trellis structure that minimizes a cost function. *Mozjpeg* adopts a simpler approach that tries to replace quantized DCT coefficients with candidate values of shorter bit lengths, while constraining distortion using a perceptual model based on a variant of PSNR-HVS [47]. This approach exploits the additivity of the distortion model in the DCT domain as well as the fact that JPEG's variable-length encoding of coefficient values is fixed in the standard, independent of neighboring coefficients, and monotonic with respect to the absolute coefficient value. This optimization leaves characteristic traces in the histogram of quantized AC DCT coefficients, which for natural images typically follows a Laplacian distribution [141]. Finally, we show that scan scripts may serve as a forensics trace to attribute an image to its source social network. Images from the Forchheim dataset that were shared via Facebook, Telegram, and Twitter use the default scan script defined in the JPEG standard. Images from WhatsApp and Instagram, however, exhibit distinct, platform-specific scan scripts.

Contribution (60% / 80% / 60%) The idea for this paper came up during a discussion in a consortium meeting of an EU project. I was the leading contributor and responsible for the conception, implementation and execution of the experiments and the analysis. Rainer Böhme helped me with the method development, the write-up and visualization, as well as the interpretation of the results. The main challenge was the lack of documentation of the inner workings of the *mozjpeg* library. We had to gather information from forum posts and analyze the source code available on GitHub. Rainer Böhme supported me during the code analysis. Maximilian Hils helped us with the web crawling.

4.2.2 Paper E*)* - Detecting Trellis Artifacts

N. Hofer. Increasing trust in image analysis by detecting trellis quantization in JPEG images. In *International Conference on Image Processing*, pages 3834–3840. IEEE, 2024 *(Conference, CORE 2023 B ranking)*

Context Signal-based forensic methods that rely on assumptions about the statistical distribution of quantized DCT coefficients are sensitive to subtle traces in the signal. JPEG codecs that employ trellis optimization produce images that require different assumptions and may compromise the reliability of forensic methods, leading to false alarms on benign images. We address this problem by modeling the artifacts that are introduced by trellis optimization and propose dedicated detection methods. We also conduct experiments that isolate all of *mozjpeg*'s signal altering optimizations and quantify their impact. Used as a pre-processor of forensic methods, our detectors could inform practitioners about potentially unreliable results and help prevent incorrect conclusions. Moreover, the detectors can serve forensics, as they indicate the use of codecs that employ trellis optimization.

Method To motivate our research, we train steganalysis detectors for three embedding methods on images compressed with *libjpeg–turbo* and evaluate them on images compressed with *mozjpeg* to

assess the sensitivity to trellis artifacts. To model these artifacts, we compress images at seven quality factors (50–100, in steps of five), and measure coefficient changes introduced by trellis optimization. Based on this analysis, we develop two analytic and three learning-based trellis detectors. The first analytic trellis detector is inspired by previous work from steganalysis [57] and uses calibration to estimate the quantized AC DCT coefficient histogram prior to trellis optimization. The second analytic detector models local histogram changes around coefficient values with shorter variable length encodings ("trellis candidate coefficients") to which the trellis algorithm shifts probability mass. The three learning-based trellis detectors differ in their feature sets and classify images using ensembles of Fisher linear discriminant analyses [54, 95]. The first uses Cartesian calibration features [57], the second uses local histogram features around trellis candidate coefficients, and the third uses JPEG rich model (JRM) features [56]. We evaluate the performance of the five detectors on in-distribution and out-of-distribution test images. Out-of-distribution test images include unseen quality factors, steganographic embedding, double JPEG compression, and *mozjpeg*'s overshoot deringing optimization.

Results Trellis optimization affected all tested images, modifying 10–18% of AC and 5–10% of DC coefficients. The overshoot deringing algorithm introduces only small changes in natural images (>1% of coefficients in 18% of the images), but substantially affects JPEG compressed images of computer graphics and text (40% of coefficients in 90% of the images). Finally, *mozjpeg* uses stronger quantization tables than the once defined in Annex K of the standard [161]. This modifies no DC, but 49–60% of AC coefficients of all tested images below quality factor 100. Trellis artifacts can affect state-of-the-art learning-based steganalysis detectors trained on images without trellis optimization. While the detectors achieve baseline accuracies of 91–99% and false-positive rates of 1–8% on clean images, they misclassify *mozjpeg* compressed images with trellis artifacts as stego images with probabilities between 43 and 99%. All five proposed trellis detectors achieve high detection accuracy (80–100%) for quality factors of 90 and above. At lower quality factors (85–50), performance degrades for the analytic detectors (to 71–82%), and, albeit less pronounced, for the detectors based on statistical learning (93–100%). The JRM features-based detector achieves near-perfect detection accuracy, also at low quality factors. We report out-of-distribution performance for the analytic and learning-based detectors that use trellis candidate coefficients and their local histogram neighbors. Both detectors are robust to overshoot deringing, and to steganographic embeddings in images without trellis artifacts. Detection accuracy decreases (by 0–9%-pts) when the images with trellis artifacts contain steganographic embedding, mainly due to increased missed detections (by 4–20%-pts). However, this scenario is rather hypothetical, as no practical stegaographic tool we are aware of uses trellis optimization. Both detectors fail to generalize to double compression, as it amplifies or washes out the trellis artifacts under certain quality factor configurations.

Contribution (95% / 95% / 100%) This is a single-author paper. I was responsible for the concept, the execution of the experiments, and the write-up. Benedikt Lorch supported me through discussions, and access to code infrastructure for statistical-learning-based steganalysis detectors. Benedikt Lorch and Rainer Böhme provided valuable feedback that improved the final manuscript.

4.3 Summary and Discussion of Results

Butora and Bas state that *mozjpeg* does not modify DCT coefficients. While this is true for the first version of the codec, things changed with version 2, which introduced codec optimizations. In an in-depth analysis of the latest *mozjpeg* compression codec, we investigate the internal workings of its

optimization algorithms and demonstrate that its neglect by the forensics community is no longer justified.

Mozjpeg differs compared to previous codecs by employing signal-modifying optimizations, including a deringing algorithm, trellis optimization, and custom quantization tables, as well as a lossless optimization of content-specific scan scripts with custom Huffman tables. The trellis optimization affects the distribution of quantized DCT coefficients. It improves the rate–distortion tradeoff by modifying quantized coefficient values, pushing them toward values of shorter variable length encoding and increasing the number of zero runs. This changes about 15% of AC coefficients in natural images, which is enough to impact the results of learning-based steganalysis detectors that rely on assumptions about a certain distribution of frequency coefficients in an image, resulting in false alarms for benign, *mozjpeg* compressed images.

Our research revealed that the share of progressive images on the web has increased, specifically since 2014, when Mozilla released *mozjpeg*. Moreover, we learned that the codec is deployed across major online platforms such as Facebook, Instagram, and WhatsApp.[4] It is therefore theoretically possible that images, currently under investigation, contain trellis artifacts, and may cause missed detections or false positives, potentially leading to incorrect forensic conclusions.

In his master's thesis, Christian Mayr built on our research and conducted an extensive evaluation of the sensitivity of forensic tools in the Amped Authenticate Software[5] on images compressed using *mozjpeg*. The software is actively used in forensics practice by law enforcement agencies. While most tools were robust, the thesis identified conditions under which a primary quantization table estimation algorithm based on statistical modeling [29] and a learning-based double-compression detector [129] were vulnerable to trellis artifacts. The primary quantization table estimation algorithm produced false results for single-compressed images containing trellis artifacts, and the double-compression detector was affected at specific quality factors. Recall that *mozjpeg* uses quantization tables that differ from the standard tables employed by *libjpeg* and *libjpeg–turbo*. To isolate the effect of trellis optimization, the experiments in the master's thesis compressed images with *mozjpeg* while using standard tables. Consequently, forensic tools found to be robust to trellis artifacts may still be affected by *mozjpeg*'s custom quantization tables.

While trellis optimization poses a risk for forensics, the traces that it leave behind can be modeled and detected, even under certain out-of-distribution scenarios. The trellis-detectors proposed in Paper *E)* should hence be employed as a preprocessing step to improve the reliability of forensic methods. Furthermore, the characteristic traces of codec optimizations might even provide opportunities, particularly for metadata-based forensic analyses. For example, custom scan scripts found in progressive images can indicate the use of *mozjpeg*.

Likewise, social network platforms are known to introduce distinctive traces in images that can serve as indicators that an image was distributed through the respective platform [132, 160]. Examples of such traces include platform-specific JPEG quantization tables, recompression signatures, characteristic metadata patterns, and consistent filename conventions [130]. As some platforms apply a fixed progressive JPEG encoding configuration for all images, their distinctive scan scripts constitute an addition platform-specific forensic trace.

[4] During a private conversation at *ICIP'24* in Abu Dhabi, two Meta employees responsible for image compression told me that *mozjpeg* is employed across their services.

[5] `https://ampedsoftware.com/authenticate`

4.4 Open Problems and Future Work

In this thesis, we demonstrated that differences in JPEG codec implementations can influence the traces present in an image under investigation. Future work may extend the analysis conducted in Paper *D)* to additional JPEG codecs, such as *Jpegli* [64], including multiple codec versions, following prior work [19]. Identifying relevant codecs requires a measurement study that reflects images encountered in practice. Such a study should include platforms that employ JPEG compression in their underlying infrastructure, including social networks, blogs, messaging systems, and image storage platforms. It should account for all usage scenarios, such as desktop and mobile browsers, mobile applications, and private direct communication versus public sharing. Measurements should be conducted in a controlled and continuous manner over an extended period.

Although JPEG forensics has been extensively studied, research on other lossy image compression formats, such as AVIF, HEIC, or WebP, remains comparatively sparse (*e.g.*, [68, 88, 117]). A comprehensive measurement study could therefore evaluate different image formats to assess whether the current focus of the forensic community is warranted or whether other formats merit increased attention. Future research may also examine scenarios in which a single image undergoes compression using multiple, different compression algorithms [102].

5. Conclusion

Current photography is undergoing a fundamental transformation. Imaging pipelines that once relied on deterministic signal processing operations to capture the light reflected from a scene are increasingly becoming entanglements of learning-based algorithms. These algorithms synthesize images from sensor input and data-driven priors learned from large image datasets during training. Current mobile phones already employ learning-based in-camera processing, such as denoising [128], color correction [65], image enhancement [9, 148], super-resolution zoom [143],[1] and aesthetic filters [126]. A technology that has emerged within this transformation is neural image compression. Although it has not yet been adopted in consumer devices and everyday applications, new codecs are proposed at a rapid pace in the literature, and the first industry standard has already been published. As we have shown, such learning-based processing operations may unintentionally alter the semantics of an image or its details. Consequently, this transition challenges long-standing assumptions about image authenticity. An image is considered authentic if it truthfully represents a captured scene, implying that processing operations do not alter the semantics of the image or its details. Given the changes introduced by neural image compression, established notions of image authenticity may therefore require reconsideration.

This shift is particularly relevant to digital image forensics, which deals with the verification of image authenticity. To ensure robust and reliable methods, the forensic community must remain aware of such developments, assess their associated risks, and adopt forensic methods accordingly. While this is especially critical for emerging technologies such as neural compression, it also applies to established compression algorithms. Although a substantial body of research on JPEG forensics has accumulated over the years, the number of new publications has recently declined. Meanwhile, the imaging industry continues to evolve, and new encoder optimizations are being deployed. As we have shown, forensic methods developed under lab assumptions can be vulnerable when applied to images produced by optimized codecs in real-world settings.

This dissertation contributes to strengthening the reliability of digital image forensics at a moment of technological transition. For JPEG, this involves keeping pace with evolving codec implementations encountered in practice. For neural image compression, it requires the development of new forensic methods, new datasets, and an increased awareness of risks that extend beyond signal-level artifacts. Most importantly, it may require revisiting concepts thought to be settled, most notably, the notion of image authenticity itself.

[1] Sebastian Rodriguez, a senior product marketing manager at Google, describes that the Pixel's Ultra Zoom feature uses "the power of AI to bend the rules of physics". `https://store.google.com/intl/en/ideas/articles/pixel-super-res-zoom/`, accessed Dec. 2025

References

[1] A. Adler and J. Tang. Synchronizing probabilities in model-driven lossless compression. *arXiv preprint arXiv:2601.10678*, 2026.

[2] S. Agarwal and H. Farid. Photo forensics from JPEG dimples. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2017.

[3] S. Agarwal and H. Farid. Photo forensics from rounding artifacts. In *Workshop on Information Hiding and Multimedia Security*, pages 103–114. ACM, 2020.

[4] E. Agustsson and R. Timofte. NTIRE 2017 Challenge on single image super-resolution: Dataset and study. In *Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[5] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, 23(1):90–93, 1974.

[6] E. Alshina. AI coding reality and perspectives – VCIP 2024 keynote presentation, 2024. `https://www.vcip2024.org/documents/VCIP2024_Keynote_E_Alshina.pdf`, (accessed: Nov 19, 2025).

[7] E. Alshina, J. Ascenso, and T. Ebrahimi. JPEG AI: The first international standard for image coding based on an end-to-end learning-based approach. *IEEE MultiMedia*, 31(4):60–69, 2024.

[8] J. Anger. vpv: Image viewer designed for image processing experts. (v0.8.2), 2023. `https://github.com/kidanger/vpv`, (accessed: Oct 15, 2025).

[9] Apple. Apple unveils iPhone 17 Pro and iPhone 17 Pro Max, the most powerful and advanced Pro models ever, 2025. `https://www.apple.com/newsroom/2025/09/apple-unveils-iphone-17-pro-and-iphone-17-pro-max/`, (accessed: Nov 21, 2025).

[10] J. Ascenso. JPEG AI: – keynote presentation at REMOVE Conference 2025, 2025. `https://www.insticc.org/Primoris/InvitedSpeakers/IMPROVE_2025/Keynotes/IMPROVE_2025_KS_2_Presentation.pdf`, (accessed: Nov 19, 2025).

[11] J. Ascenso, E. Alshina, and T. Ebrahimi. The JPEG AI standard: Providing efficient human and machine visual data consumption. *IEEE MultiMedia*, 30(1):100–111, 2023.

[12] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimization of nonlinear transform codes for perceptual quality. In *Picture Coding Symposium*, pages 1–5. IEEE, 2016.

[13] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017.

[14] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

[15] J. Ballé, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici. Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):339–353, 2020.

[16] K. Barnard, G. Finlayson, and B. Funt. Color constancy for scenes with varying illumination. *Computer Vision and Image Understanding*, 65(2):311–321, 1997.

[17] M. Barni and F. Bartolini. *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*. CRC Press, 2004.

[18] M. Beneš, N. Hofer, and R. Böhme. The effect of the JPEG implementation on the cover-source mismatch error in image steganalysis. In *European Signal Processing Conference*, pages 1057–1061. IEEE, 2022.

[19] M. Beneš, N. Hofer, and R. Böhme. Know your library: How the libjpeg version influences compression and decompression results. In *Workshop on Information Hiding and Multimedia Security*, pages 19–25. ACM, 2022.

[20] M. Beneš, B. Lorch, and R. Böhme. JPEG steganalysis using leaked cover thumbnails. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2023.

[21] S. Bergmann, D. Moussa, F. Brand, A. Kaup, and C. Riess. Frequency-domain analysis of traces for the detection of AI-based compression. In *International Workshop on Biometrics and Forensics*, pages 1–6. IEEE, 2023.

[22] S. Bergmann, D. Moussa, F. Brand, A. Kaup, and C. Riess. Forensic analysis of AI-compression traces in spatial and frequency domain. *Pattern Recognition Letters*, 180:41–47, 2024.

[23] S. Bergmann, D. Moussa, and C. Riess. Trustworthy compression? Impact of AI-based codecs on biometrics for law enforcement. *arXiv preprint arXiv:2408.10823*, 2024.

[24] S. Bergmann, F. Brand, and C. Riess. Three forensic cues for JPEG AI images. *arXiv preprint arXiv:2504.03191*, 2025.

[25] S. Bergmann, D. Moussa, and C. Riess. Polished pixels: impact of AI compression on image-based evidence. *Multimedia Tools and Applications*, 85(2):95, 2026.

[26] A. Berthet and J.-L. Dugelay. AI-based compression: A new unintended counter attack on JPEG-related image forensic detectors? In *International Conference on Image Processing*, pages 3426–3430. IEEE, 2022.

[27] A. Berthet, C. Galdi, and J.-L. Dugelay. On the impact of AI-based compression on deep learning-based source social network identification. In *International Workshop on Multimedia Signal Processing*, pages 1–6. IEEE, 2023.

[28] S. Bhardwaj, I. Fischer, J. Ballé, and T. Chinen. An unsupervised information-theoretic perceptual quality metric. *Advances in Neural Information Processing Systems*, 33:13–24, 2020.

[29] T. Bianchi and A. Piva. Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, 2012.

[30] R. Böhme. *Advanced Statistical Steganalysis*. Springer Publishing Company, Incorporated, 2010.

[31] N. Bonettini, L. Bondi, P. Bestagini, and S. Tubaro. JPEG implementation forensics based on Eigen-algorithms. In *International Workshop on Information Forensics and Security*, pages 1–7. IEEE, 2018.

[32] T. Boult and G. Wolberg. Correcting chromatic aberrations using image warping. In *Computer Vision and Pattern Recognition*, pages 684–687, 1992.

[33] A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[34] J. Butora and P. Bas. High quality JPEG compressor detection via decompression error. In *GRETSI*, 2022. `https://hal.science/hal-03697777`.

[35] F. W. Campbell and J. G. Robson. Application of Fourier analysis to the visibility of gratings. *The Journal of Physiology*, 197(3):551–566, 1968.

[36] E. D. Cannas, S. Mandelli, N. Popovic, A. Alkhateeb, A. Gnutti, P. Bestagini, and S. Tubaro. Is JPEG AI going to change image forensics? In *International Conference on Computer Vision*, pages 1564–1575. IEEE/CVF, 2025.

[37] J. P. Cardenuto, J. Krinsky, L. Nogueira, A. Bharati, and D. Moreira. Implications of neural compression to scientific images. In *Workshop on Information Hiding and Multimedia Security*, pages 80–85. ACM, 2025.

[38] M. Carnein, P. Schöttle, and R. Böhme. Forensics of high-quality JPEG images with color subsampling. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2015.

[39] M. Chen, J. Fridrich, M. Goljan, and J. Lukás. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90, 2008.

[40] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Routledge, 2013. 2^{nd} ed.

[41] V. Conotter, G. Boato, and H. Farid. Detecting photo manipulation on signs and billboards. In *International Conference on Image Processing*, pages 1741–1744. IEEE, 2010.

[42] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2007. 2^{nd} ed.

[43] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (5):2567–2581, 2020.

[44] B. Dornauer and M. Felderer. Web image formats: Assessment of their real-world-usage and performance across popular web browsers. In *International Conference on Product-Focused Software Process Improvement*, pages 132–147. Springer, 2023.

[45] Z. Duan, M. Lu, Z. Ma, and F. Zhu. Opening the black box of learned image coders. In *Picture Coding Symposium*, pages 73–77. IEEE, 2022.

[46] E. Dworetzky and J. Fridrich. JPEG compatibility attack revisited. *IEEE Transactions on Information Forensics and Security*, 2021.

[47] K. Egiazarian, J. Astola, N. Ponomarenko, V. Lukin, F. Battisti, and M. Carli. New full-reference quality metrics based on HVS. In *International Workshop on Video Processing and Quality Metrics*, volume 4, page 4, 2006.

[48] A. El Gamal and H. Eltoukhy. CMOS image sensors. *IEEE Circuits and Devices Magazine*, 21(3):6–20, 2005.

[49] S. Esenlik, Y. Wu, Z. Zhang, Y.-K. Wang, K. Zhang, L. Zhang, J. Ascenso, and S. Liu. An overview of the JPEG AI learning-based image coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2025.

[50] J. Fan, H. Cao, and A. C. Kot. Estimating EXIF parameters based on noise features for image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 8 (4):608–618, 2013.

[51] H. Farid. Digital image ballistics from JPEG quantization. *Dartmouth College*, 2006. Technical report.

[52] H. Farid. Exposing digital forgeries from JPEG ghosts. *IEEE Transactions on Information Forensics and Security*, 4(1):154–160, 2009.

[53] H. Farid. *Photo Forensics*. MIT Press, 2016.

[54] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[55] J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, 2010.

[56] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.

[57] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images: Breaking the F5 algorithm. In *International Workshop on Information Hiding*, pages 310–323. Springer, 2002.

[58] J. Fridrich, D. Soukal, J. Lukas, et al. Detection of copy-move forgery in digital images. In *Digital Forensic Research Workshop*, volume 3, pages 652–63, 2003.

[59] G. L. Friedman. The trustworthy digital camera: Restoring credibility to the photographic image. *IEEE Transactions on Consumer Electronics*, 39(4):905–910, 1993.

[60] H. Fu, X. Zhao, and X. He. Improving anticompression robustness of JPEG adaptive steganography based on robustness measurement and DCT block selection. *Security and Communication Networks*, 2021(1):9153468, 2021.

[61] O. Giudice, A. Paratore, M. Moltisanti, and S. Battiato. A classification engine for image ballistics of social data. In *International Conference on Image Analysis and Processing*, pages 625–636. Springer, 2017.

[62] T. Gloe. Forensic analysis of ordered data structures on the example of JPEG files. In *International Workshop on Information Forensics and Security*, pages 139–144. IEEE, 2012.

[63] T. Gloe, K. Borowka, and A. Winkler. Efficient estimation and large-scale evaluation of lateral chromatic aberration for digital image forensics. In *Media Forensics and Security II*, volume 7541, pages 62–74. SPIE, 2010.

[64] Google. Jpegli GitHub, 2024. `https://github.com/google/jpegli`, (accessed: Nov 14, 2025).

[65] Google. How Real Tone helps make a more equitable camera, 2025. `https://store.google.com/intl/en_uk/ideas/articles/inclusive-photography-real-tone/`, (accessed: Nov 24, 2025).

[66] P. Gupta, P. Srivastava, S. Bhardwaj, and V. Bhateja. A modified PSNR metric based on HVS for quality assessment of color images. In *International Conference on Communication and Industrial Application*, pages 1–4. IEEE, 2011.

[67] B. Hadwiger and C. Riess. The forchheim image database for camera identification in the wild. In *Pattern Recognition, Computer Vision, and Image Processing*, pages 500–515. Springer, 2021.

[68] M. Hafner, A. Radovic, M. Langer, S. Findenig, and A. Uhl. Forensic recognition of codec-specific image compression artefacts. In *Workshop on Information Hiding and Multimedia Security*, pages 131–136. ACM, 2024.

[69] N. Hofer. Increasing trust in image analysis by detecting trellis quantization in JPEG images. In *International Conference on Image Processing*, pages 3834–3840. IEEE, 2024.

[70] N. Hofer and R. Böhme. Progressive JPEGs in the wild: Implications for information hiding and forensics. In *Workshop on Information Hiding and Multimedia Security*, pages 47–58. ACM, 2023.

[71] N. Hofer and R. Böhme. A taxonomy of miscompressions: Preparing image forensics for neural compression. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2024.

[72] N. Hofer and R. Böhme. Challenging cases of neural image compression: A dataset of visually compelling yet semantically incorrect reconstructions. In *International Conference on Multimedia*, pages 13318–13324. ACM, 2025.

[73] N. Hofer and R. Böhme. SCLIC - semantic changes in learning based image compression (version 1.0.0.). `https://doi.org/10.5281/zenodo.16780952`, 2025. Dataset, published at ACM International Conference on Multimedia 2025.

[74] N. Hofer and R. Böhme. When the codec hallucinates: User perceptions of miscompressed images. In *CHI Conference on Human Factors in Computing Systems*. ACM, 2026. to appear.

[75] P. G. Howard, F. Kossentini, B. Martins, S. Forchhammer, and W. J. Rucklidge. The emerging JBIG2 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(7):838–848, 2002.

[76] G. Hudson, A. Léger, B. Niss, I. Sebestyén, and J. Vaaben. JPEG-1 standard 25 years: Past, present, and future reasons for a success. *Journal of Electronic Imaging*, 27(4):040901–040901, 2018.

[77] IJG. libjpeg, 1990. http://libjpeg.sourceforge.net/, (accessed: Nov 14, 2025).

[78] IJG. Recommendation ITU-T T.871, 1994. https://www.ijg.org/files/T-REC-T.871-201105-I!!PDF-E.pdf, (accessed: Nov 13, 2025).

[79] ISO. IEC 10918-1: Information technology - digital compression and coding of continuous-tone still images: Requirements and guidelines, 1994.

[80] ISO. IEC 6048-1:2025: Information technology – JPEG AI learning-based image coding system. part 1: Core coding system, 2025.

[81] ITU-T. ITU-T T.81: Digital compression and coding of continuous-tone still images, 1992.

[82] ITU-T. ITU-T T.840.1: Information technology – JPEG AI learning-based image coding system: Core coding system, 2025. https://handle.itu.int/11.1002/1000/16265.

[83] E. Jalilian, H. Hofbauer, and A. Uhl. Iris image compression using deep convolutional neural networks. *Sensors*, 22(7):2698, 2022.

[84] M. K. Johnson and H. Farid. Exposing digital forgeries by detecting inconsistencies in lighting. In *Workshop on Multimedia and Security*, pages 1–10. ACM, 2005.

[85] M. K. Johnson and H. Farid. Exposing digital forgeries through chromatic aberration. In *Workshop on Multimedia and Security*, pages 48–55. ACM, 2006.

[86] M. K. Johnson and H. Farid. Exposing digital forgeries through specular highlights on the eye. In *International Workshop on Information Hiding*, pages 311–325. Springer, 2007.

[87] O. R. Joubert, G. A. Rousselet, D. Fize, and M. Fabre-Thorpe. Processing scene context: Fast categorization and object interference. *Vision Research*, 47(26):3286–3297, 2007.

[88] V. Kadha and S. K. Das. Detecting image manipulation in lossy compression: A multi-modality deep-learning framework. In *Region 10 Conference (TENCON)*, pages 795–800. IEEE, 2023.

[89] H. R. Kang. *Color Technology for Electronic Imaging Devices*. SPIE, 1997.

[90] S. Katzenbeisser and F. Petitcolas. *Information Hiding*. Artech House information security and privacy series. Artech House, 2015.

[91] E. Kee and H. Farid. Exposing digital forgeries from 3-D lighting environments. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2010.

[92] E. Kee, M. K. Johnson, and H. Farid. Digital image authentication from JPEG headers. *IEEE Transactions on Information Forensics and Security*, 6(3):1066–1075, 2011.

[93] M. Kirchner. Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In *Workshop on Multimedia and Security*, pages 11–20. ACM, 2008.

[94] M. Kirchner. Notes on digital image forensics and counter-forensics, 2011. Excerpt from the Ph.D. thesis entitled "Forensic Analysis of Resampled Digital Signals". Available at https://ws.binghamton.edu/kirchner/papers/image_forensics_and_counter_forensics.pdf.

[95] J. Kodovsky, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444, 2011.

[96] J. D. Kornblum. Using JPEG quantization tables to identify imagery processed by software. *Digital Investigation*, 5:S21–S25, 2008.

[97] K. Krippendorff. *Content Analysis: An Introduction to its Methodology*. Sage Publications, 2018.

[98] C. Kumawat and V. Pankajakshan. A JPEG forensic detector for color bitmap images. *IEEE Open Journal of Signal Processing*, 2:280–294, 2021.

[99] S. Lai and R. Böhme. Block convergence in repeated transform coding: JPEG-100 forensics, carbon dating, and tamper detection. In *International Conference on Acoustics, Speech and Signal Processing*, pages 3028–3032. IEEE, 2013.

[100] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Annual Network and*

Distributed System Security Symposium, 2019. `https://tranco-list.eu/list/5Y5GN`, (accessed: Mar 05, 2023).

[101] E. Levecque, J. Butora, and P. Bas. Dual JPEG compatibility: A reliable and explainable tool for image forensics. *arXiv preprint arXiv:2408.17106*, 2024.

[102] B. Li, J. Shi, W. Li, and H. Li. WebP-JPEG transcoding detection by spotting re-compression artifacts with CNN-ViT for processing dual-domain features. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[103] J. Li, D. Li, S. Savarese, and S. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pages 19730–19742. PMLR, 2023.

[104] X. Li, B. Gunturk, and L. Zhang. Image demosaicing: A systematic survey. In *Visual Communications and Image Processing*, volume 6822, pages 489–503. SPIE, 2008.

[105] libjpeg–turbo. libjpeg–turbo Main Homepage, 2021. `https://libjpeg-turbo.org/`, (accessed: Nov 14, 2025).

[106] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[107] C. Loeffler, A. Ligtenberg, and G. S. Moschytz. Practical fast 1-D DCT algorithms with 11 multiplications. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 988–991. IEEE, 1989.

[108] H. Lohscheller. *Einzelbildübertragung mit wachsender Auflösung*. PhD thesis, Technische Hochschule Aachen, 1982. Dissertation.

[109] B. Lorch and R. Böhme. Landscape more secure than portrait? Zooming into the directionality of digital images with security implications. In *USENIX Security Symposium*, pages 6903–6920, 2024.

[110] B. Lorch and C. Riess. Image forensics from chroma subsampling of high-quality JPEG images. In *Workshop on Information Hiding and Multimedia Security*, pages 101–106. ACM, 2019.

[111] B. Lorch, A. Maier, and C. Riess. Reliable JPEG forensics via model uncertainty. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2020.

[112] W. Lu, J. Zhang, X. Zhao, W. Zhang, and J. Huang. Secure robust JPEG steganography based on autoencoder with adaptive bch encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7):2909–2922, 2020.

[113] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, 2006.

[114] R. G. Mani, R. Parthasarathy, S. Eswaran, and P. Honnavalli. A survey on digital image forensics: Metadata and image forgeries. In *Workshop on Applied Computing*, pages 27–28, 2022.

[115] D. Mari, S. Cavasin, S. Milani, and M. Conti. Effectiveness of learning-based image codecs on fingerprint storage. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2024.

[116] O. Mayer and M. C. Stamm. Forensic similarity for digital images. *IEEE Transactions on Information Forensics and Security*, 15:1331–1346, 2019.

[117] S. McKeown and G. Russell. Forensic considerations for the high efficiency image file format (HEIF). In *International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pages 1–8. IEEE, 2020.

[118] S. McKeown, G. Russell, and P. Leimich. Fingerprinting JPEGs with optimised huffman tables. *Journal of Digital Forensics, Security and Law*, 13(2):7, 2018.

[119] F. Mentzer, G. D. Toderici, M. Tschannen, and E. Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33:11913–11924, 2020.

[120] Q. Milliet, O. Delémont, and P. Margot. A forensic science perspective on the role of images in crime investigation and reconstruction. *Science & Justice*, 54(6):470–480, 2014.

[121] Mozilla Foundation. MozJPEG: Improved JPEG encoder, 2014. `https://github.com/mozilla/mozjpeg`, (accessed: Nov 14, 2025).

[122] P. Mullan, C. Riess, and F. Freiling. Forensic source identification using JPEG image headers: The case of smartphones. *Digital Investigation*, 28:68–76, 2019.

[123] S. Murdoch and M. Dornseif. Hidden data in internet published documents, 2004. `https://events.ccc.de/congress/2004/fahrplan/event/271.en.html`, (accessed: Nov 21, 2025).

[124] J. Norman and H. Farid. An investigation into the impact of AI-powered image enhancement on forensic facial recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 4306–4314. IEEE/CVF, 2024.

[125] G. Notess. The wayback machine: The web's archive. *Information Today Inc.*, 26(2):59–61, 2002.

[126] S.-C. Opitz. Beauty filters — when beauty is standardized, 2020. `https://www.fotomuseum.ch/wp-content/uploads/2024/06/Beauty_Filters_EN.pdf`, (accessed: Nov 24, 2025).

[127] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, pages 1–31, 2024.

[128] D. Packwood. Smarter smartphone photography: Unlocking the power of neural camera denoising with ARM SME2, 2025. `https://developer.arm.com/community/arm-community-blogs/b/ai-blog/posts/unlocking-the-power-of-neural-camera-denoising-with-arm-sme2`, (accessed: Nov 24, 2025).

[129] J. Park, D. Cho, W. Ahn, and H.-K. Lee. Double JPEG detection in mixed JPEG quality factors using deep convolutional neural network. In *European Conference on Computer Vision*, pages 636–652, 2018.

[130] C. Pasquini, I. Amerini, and G. Boato. Media forensics on social media platforms: a survey. *EURASIP Journal on Information Security*, 2021(1):4, 2021.

[131] W. B. Pennebaker and J. L. Mitchell. *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[132] Q.-T. Phan, G. Boato, R. Caldelli, and I. Amerini. Tracking multiple image sharing on social networks. In *International Conference on Acoustics, Speech and Signal Processing*, pages 8266–8270. IEEE, 2019.

[133] A. Piva. An overview on image forensics. *International Scholarly Research Notices*, 2013(1): 496701, 2013.

[134] A. Piva and M. Iuliani. Integrity verification through file container analysis. In H. T. Sencar, L. Verdoliva, and N. Memon, editors, *Multimedia Forensics*, pages 363–387. Springer, 2022.

[135] A. C. Popescu and H. Farid. Statistical tools for digital forensics. In *International Workshop on Information Hiding*, pages 128–147. Springer, 2004.

[136] A. C. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, 2005.

[137] C. A. Poynton. Rehabilitation of gamma. In *Human Vision and Electronic Imaging III*, volume 3299, pages 232–249. SPIE, 1998.

[138] T. Qiu, A. Nichani, R. Tadayontahmasebi, and H. Jeong. Gone with the bits: Revealing racial bias in low-rate neural compression for facial images. In *Conference on Fairness, Accountability, and Transparency*, pages 1862–1889. ACM, 2025.

[139] C. V. Ragaglia, L. Catania, F. Guarnera, D. Allegra, and S. Battiato. What if retrieval could work before decoding? The case of JPEG AI latents for deepfake source attribution. In *Deepfake Forensics Workshop at International Conference on Multimedia*, pages 21–28. ACM, 2025.

[140] E. Reinhard, E. A. Khan, A. O. Akyuz, and G. Johnson. *Color Imaging: Fundamentals and Applications*. CRC Press, 2008.

[141] R. Reininger and J. Gibson. Distributions of the two-dimensional DCT coefficients for images. *IEEE Transactions on Communications*, 31(6):835–839, 1983.

[142] L. Relic, R. Azevedo, M. Gross, and C. Schroers. Lossy image compression with foundation diffusion models. In *European Conference on Computer Vision*, pages 303–319. Springer, 2024.

[143] I. Reynolds. 10 things to know about the camera on Pixel 10, 2025. `https://blog.google/products/pixel/pixel-10-camera-features/`, (accessed: Nov 21, 2025).

[144] C. Riess. Physical integrity. In H. T. Sencar, L. Verdoliva, and N. Memon, editors, *Multimedia Forensics*, pages 207–234. Springer, 2022.

[145] C. Riess and E. Angelopoulou. Scene illumination as an indicator of image manipulation. In R. Böhme, P. W. L. Fong, and R. Safavi-Naini, editors, *Information Hiding International Conference*, volume 6387, pages 66–80. Springer, 2010.

[146] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[147] M. Safna Asiq and W. Sam Emmanuel. Colour filter array demosaicking: A brief survey. *The Imaging Science Journal*, 66(8):502–512, 2018.

[148] Samsung UK. How Samsung phone camera uses AI for moon photos & pictures, 2024. `https://www.samsung.com/uk/support/mobile-devices/how-galaxy-cameras-combine-super-resolution-technologies-with-ai-to-produce-high-quality-images-of-the-moon/`, (accessed: Nov 19, 2025).

[149] A. Schlögl, N. Hofer, and R. Böhme. Causes and effects of unanticipated numerical deviations in neural network inference frameworks. *Advances in Neural Information Processing Systems*, 36, 2024.

[150] I. Sebestyén. Some little-known aspects of the history of the JPEG still picture-coding standard, ITU-T T. 81 ISO/IEC 10918-1 (1986-1993). *ITU Journal: ICT Discoveries*, 3(1), 2020.

[151] H. T. Sencar and N. Memon. Identification and recovery of JPEG files with missing fragments. *Digital Investigation*, 6:88–98, 2009.

[152] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[153] C. Solomon and T. Breckon. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. John Wiley & Sons, 2011.

[154] T. H. Thai, R. Cogranne, F. Retraint, and T.-N.-C. Doan. JPEG quantization step estimation and its applications to digital image forensics. *IEEE Transactions on Information Forensics and Security*, 12(1):123–133, 2016.

[155] L. Theis, W. Shi, A. Cunningham, and F. Huszár. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations*, 2017.

[156] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar. Variable rate image compression with recurrent neural networks. In *International Conference on Learning Representations*, 2016.

[157] V. van der Meer and J. van den Bos. JPEG file fragmentation point detection using Huffman code and quantization array validation. In *International Conference on Availability, Reliability and Security*, pages 1–7, 2021.

[158] V. van der Meer, J. van den Bos, H. Jonker, and L. Dassen. Problem solved: A reliable, deterministic method for JPEG fragmentation point detection. *Forensic Science International: Digital Investigation*, 48:301687, 2024.

[159] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[160] S. Verde, C. Pasquini, F. Lago, A. Goller, F. De Natale, A. Piva, and G. Boato. Multi-clue reconstruction of sharing chains for social media images. *IEEE Transactions on Multimedia*, 25:9491–9505, 2023.

[161] G. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992. doi: 10.1109/30.125072.

[162] J. Wang, H. Wang, J. Li, X. Luo, Y.-Q. Shi, and S. K. Jha. Detecting double JPEG compressed color images with the same quantization matrix in spherical coordinates. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8):2736–2749, 2019.

[163] Z. Wang, E. Simoncelli, and A. Bovik. Multiscale structural similarity for image quality assessment. In *Conference on Signals, Systems & Computers*, pages 1398–1402. IEEE, 2003.

[164] J. Wen, M. Luttrell, and J. Villasenor. Trellis-based RD optimal quantization in H. 263+. *IEEE Transactions on Image Processing*, 9(8):1431–1434, 2000.

[165] D. White, P. J. Phillips, C. A. Hahn, M. Hill, and A. J. O'Toole. Perceptual expertise in forensic facial image comparison. *Royal Society B: Biological Sciences*, 282(1814):20151292, 2015.

[166] Wikimedia commons. Help:jpeg, historic version, 2013. `https://commons.wikimedia.org/w/index.php?title=Help:JPEG&oldid=88263460`, (accessed: Oct 21, 2025).

[167] Wikimedia Commons. Help:jpeg, 2017. `https://commons.wikimedia.org/wiki/Help:JPEG`, (accessed: Oct 21, 2025).

[168] R. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, pages 229–256, 1992.

[169] M. Yakubenko and M. Gashnikov. The influence of neural network image compression methods on digital watermarks. In *International Conference on Information Technology and Nanotechnology*, pages 1–5. IEEE, 2024.

[170] R. Yang and S. Mandt. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 36:64971–64995, 2023.

[171] Y. Yang, S. Mandt, and L. Theis. An introduction to neural data compression. *Foundations and Trends® in Computer Graphics and Vision*, 15(2):113–200, 2023.

[172] S. Ye, Q. Sun, and E.-C. Chang. Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In *International Conference on Multimedia and Expo*, pages 12–15. IEEE, 2007.

[173] G. J. Zelinsky. Understanding scene understanding. *Frontiers in Psychology*, 4:954, 2013.

[174] R. Zhang, P. Isola, A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern Recognition*, pages 586–595. IEEE, 2018.

[175] Z. Zhang, Y. Ren, X.-J. Ping, Z.-Y. He, and S.-Z. Zhang. A survey on passive-blind image forgery by doctor method detection. In *International Conference on Machine Learning and Cybernetics*, volume 6, pages 3463–3467. IEEE, 2008.

[176] R. Zou, C. Song, and Z. Zhang. The devil is in the details: Window-based attention for image compression. In *Conference on Computer Vision and Pattern Recognition*, pages 17492–17501. IEEE/CVF, 2022.

Part II

Papers

A. Taxonomy of Miscompressions

Authors

Nora Hofer, University of Innsbruck
Rainer Böhme, University of Innsbruck

Title

A Taxonomy of Miscompressions: Preparing Image Forensics for Neural Compression

Conference

IEEE International Workshop on Information Forensics and Security (WIFS '24)
Rome, Italy · December, 02–05, 2024

Abstract

Neural compression has the potential to revolutionize lossy image compression. Based on generative models, recent schemes achieve unprecedented compression rates at high perceptual quality, but they compromise semantic fidelity. Details of decompressed images may appear optically flawless, but semantically different from the originals, making compression errors difficult or impossible to detect. We explore the problem space and propose a provisional taxonomy of miscompressions. It defines three types of "what happens" and has a binary "high impact" flag indicating miscompressions that alter symbols. We discuss how the taxonomy can facilitate risk communication and research into mitigations.

(a) Lossless original (4.1 MB, 11.90 bpp)

(b) Neural compression (159 kB, 0.46 bpp, [30])

(c) JPEG QF 20 (158 kB, 0.46 bpp)

Figure A.1: State-of-the-art neural compression schemes can alter the semantic details of the decompressed images. The high fidelity and the lack of visible compression artifacts make false reconstructions look more authentic than JPEG, which introduces visible distortion. (Crop of image 0831 of DIV2K [3], 0.41% of the original.) All figures are best viewed on screen and magnified.

A.1 Introduction

A turning point in the investigation of the 2013 Boston Marathon bombing was a bystander's cellphone photo that allowed police to identify one of the suspects in a crowd [1] [2]. Remarkably, the relevant part of the image comprised just 0.2 % of all pixels. In this paper, we ask the question whether digital images will continue to serve as reliable sources in a future where neural compression becomes the default.

Neural image compression employs learning-based elements in the image compression pipeline, achieving high perceptual quality at unprecedented compression rates [40]. State of the art schemes use generative networks to synthesize parts of an image [30, 39]. However, a drawback of this approach is that the synthesized details appear plausible and of high perceptual quality, but may be semantically different from the original.

To illustrate this, Fig. A.1 compares a small crop (0.4 %) of an uncompressed image (left), to a version from the *HiFiC* neural compression scheme [30] (middle), and the JPEG compressed image at quality factor (QF) 20 (right). This factor was chosen to match the compression rate in bits per pixel between *HiFiC* and JPEG. While the neural compression retains clear readability of the numbers, closer inspection reveals that they differ: the upper row changes from $2264 - 7668$ in the original to $2254 - 7664$ in the *HiFiC* reconstruction. We confirmed that not only human observers, but also Google's Cloud AI optical character recognition came to this conclusion. Given the high apparent quality of the image, observers unaware of its processing history might be inclined to fully trust the image and its semantic content. By contrast, the visible compression artifacts in the JPEG image not only make the numbers difficult to read, but also signal low reliability and dissuade users from interpreting the numbers with confidence.

In this paper, we propose the term "miscompression" to describe semantic changes resulting from lossy compression. This new[1] phenomenon arose with neural compression and deserves the attention of researchers and forensic practitioners. To facilitate the conversation, we develop a taxonomy of miscompressions based on the explorative visual inspection of three benchmark datasets, examining

[1]The closest related work we are aware of is an attempt to introduce copy-evident marks into images which only appear after JPEG compression [26].

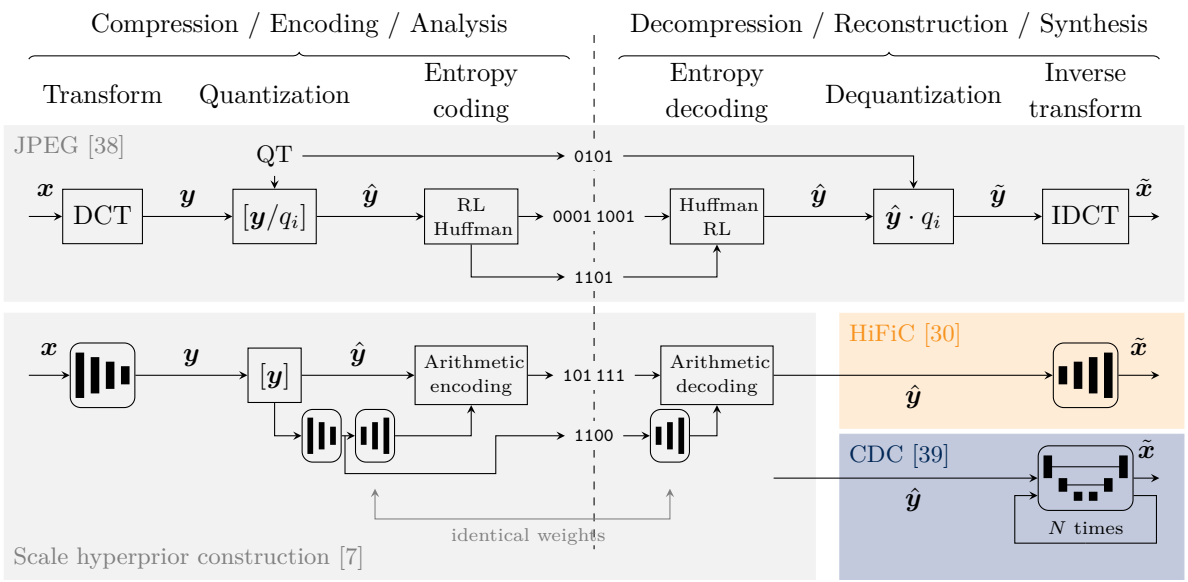


Figure A.2: Comparison of the conventional JPEG compression (top) and the neural compression schemes used in this paper (bottom). Elements with rounded corners are CNNs trained on datasets and used in inference mode during encoding and reconstruction.

five different neural compression schemes at different quality settings. The examples used for illustration in this paper are produced using two schemes of the latest generation. We derive implications for forensics and society at large, and outline objectives for research into mitigations.
We emphasize that all authors of recent neural compression schemes generally acknowledge the risk of hallucinations, warn against using their schemes for critical applications [30, p. 10], and occasionally point to examples of text becoming unreadable. However, miscompressions take many forms and pose a risk to society if left unaddressed. We see our work as a first step towards solutions.
The paper is structured as follows. Section A.2 introduces the principles of neural compression by comparing it to the JPEG pipeline. Section A.3 defines miscompressions, describes our empirical approach, and presents the proposed taxonomy using examples. Section A.4 discusses implications and outlines a research agenda. Section A.5 concludes our paper.

A.2 Primer on neural compression

The top row of Fig. A.2 shows the key components of the lossy image compression and decompression pipeline. The first component **transforms** an input image from the spatial domain into a domain where pixels are decorrelated and the variance is concentrated in fewer coefficients. A useful property of common transformations is that the distribution of the coefficients can be approximated parametrically. The second component, **quantization**, is a deliberately lossy process. It maps a range of values to a single discrete value. The quantization steps can be adjusted according to the relevance of each coefficient for the reconstruction of the signal. The final component of the compression pipeline is lossless **entropy coding**, which approximates Shannon's theorem [33] by encoding a sequence of quantized values into a sequence of bits. The better the input distribution is known, the tighter the approximation and the shorter the resulting bitstream.
Over decades, JPEG [38] has been the most popular lossy image compression scheme. It implements the transform by the linear discrete cosine transform (DCT) [4] on non-overlapping blocks of

8×8 pixels. The resulting coefficients are then quantized by dividing them by frequency-specific quantization factors and subsequent rounding to the nearest integer. The quantized coefficients are arranged in zigzag order and entropy-coded using run-length (RL) and Huffman encoding [21]. The resulting image file contains the quantization tables (QT), the quantized DCT coefficients, and the Huffman tables. JPEG decompression reverts this sequence of steps.

Neural compression replaces components of this pipeline with learnable elements, typically deep convolutional neural networks (CNN). This emerging field has its own jargon. Encoding and analysis are used to describe compression. Reconstruction and synthesis denote decompression (cf. Fig. A.2). Learning the transform promises that irrelevance in the input signal can be isolated better in the so-called latent space than with known structured transformations, such as block-wise DCT. While the networks have shown to derive basis functions similar to those in linear transforms [15], nonlinear transforms offer better adaptation to varying data distributions and can be optimized for specific distortion metrics. The loss function used for training has two terms: distortion and rate. By weighting these terms, different tradeoffs between image quality and file size can be achieved. Finding the right distortion metric for neural compression is an active field of research [12, 14]. Once trained, the weights are stored in the encoder and decoder. The CNNs are used in inference mode for the encoding and reconstruction of images.

Quantization in neural compression typically involves rounding and truncation [6, 36]. Unlike JPEG, it does not use and transmit QTs. The quantization step size is controlled by the scaling in last layer of the transform CNN and thus learned. Therefore, neural compression schemes commonly require a separately trained model for each target quality.

Also, entropy coding requires modifications. A drawback of learning a transform is the lack of a statistical model of the latent space. Here, the answer to machine learning is machine learning. The distribution of the latent space is modelled with a trained auto encoder. The prediction of this model is used to parameterize an arithmetic encoder. As the distribution is data dependent, the latent space of this prediction model must itself be transmitted to the decoder to enable reconstruction. Ballé et al.'s scale hyperprior construction [7] is the basis of the two schemes evaluated in this work. *HiFiC* [30] and *CDC* [39] improve on previous approaches to neural compression by using generative models for the inverse transform. *HiFiC* uses a **generative adversarial network** (GAN). GANs are trained with a rivaling discriminator network that regularizes the generator network towards producing outputs of high perceptual quality [18]. To control the content of the reconstruction, the generator is conditioned with the latent representation of the encoded image. *CDC* implements a **diffusion model** [35] for the inverse transform. It uses the latent representation to condition the denoising diffusion probabilistic model [19] (DDPM). Both schemes are trained end-to-end, allowing the variational autoencoder in the transform component to learn how to turn an input image to a condition. The rate estimate of the loss function is taken from the hyperprior model and the distortion estimate is a weighted sum of the perceptual loss and the mean squared error between the input and the reconstructed image. Increasing the weight of the perceptual metric gives the model the flexibility to deviate from the input signal and "make up" details during reconstruction. This enables high perceptual quality at unprecedented compression rates, but compromises semantic fidelity.

The digital image forensics community has only recently started to address the impact of neural compression: Berthet et al. revisit copy-move forgery detection [10] and source social network identification [11], Bergmann et al. use traces in the frequency and spatial domain for detection [8, 9], and Chen et al. show a vulnerability of different neural compression schemes to adversarial perturbations in the input image [13]. Jalilian et al. propose CNNs to compress biometric images [23]. However, to the best of our knowledge, nobody has yet investigated semantic changes and their implications.

A.3 Miscompressions

Semantic interpretation is the understanding of a perceived scene by applying domain terminology, *i.e.*, semantic concepts [29]. It is carried out by a human observer and is heavily influenced by their prior semantic and conceptual knowledge of the domain [20]. The *semantic meaning* of a scene or an image is the result of semantic interpretation.

We define **miscompressions** as reconstruction errors that occur when there is a discrepancy between the semantic meaning of an original image (detail) and its reconstructed version after neural compression. As a test, we require that a human observer asked to verbally describe the relevant part of the image would come up with a different description. Note that this definition applies to entire images as well as individual image details. Digital images used as evidence in forensic investigations often capture relevant details unconsciously, as can be seen from the Boston Marathon bombing example. Such photographs often preserve details that serve as objective representations of reality. Consequently, forensic investigators typically focus on analyzing the semantic meaning of specific image details, such as objects or individuals in the background, rather than interpreting the semantics of the entire image.

Miscompressions are a new phenomenon, requiring a precise terminology to describe, mitigate, or ultimately avoid them. This paper takes a first step in this direction and proposes a provisional taxonomy, systematically derived with three key objectives in mind. First, the taxonomy should facilitate research into the risks posed by miscompressions. Distinguishing between different forms of reconstruction errors that lead to miscompressions allows us to measure their prevalence, and compare compression schemes and the optimization metrics used. Second, it should facilitate research towards making neural compression safer. While ideal compression schemes would be completely immune to miscompressions, it is uncertain whether this is achievable at competitive compression rates. However, confidence in neural compression would be greatly improved if it could be ensured that certain types of miscompressions are extremely unlikely. The third intended application of our taxonomy is to deal with the remaining risk in practice. It should allow forensics experts to explain miscompressions using references to scientific evidence in order to convince a judge or jury.

A.3.1 Method

Our approach is exploratory. We focus on five relevant neural compression schemes [6, 7, 30, 31, 39], compress the test images of three widely-used benchmark datasets, *CLIC2020* [37], *DIV2K* [3], and *Kodak* [16] (full dataset), and manually inspect the reconstructions of a total of 552 images to identify miscompressions, using difference images for guidance, where necessary.

We have observed miscompressions in all tested schemes, but decided to shift our focus to *HiFiC* [30] and *CDC* [39]. These schemes employ generative networks and stand out, as their reconstructions are of such high fidelity that they appear deceptively authentic. We use the pre-trained *HiFiC* model with 180 million parameters for the three available compression intensities *high*, *mid*, and *low*,[2] and the pre-trained $\mathcal{X}$-parameterization model of *CDC* with 54 million parameters for the widely-adopted LPIPS loss at weights 0.0 and 0.9.[3] We varied the noise seeds for *CDC* and found that miscompressions prevailed.

[2] *HiFiC*: `https://github.com/tensorflow/compression/tree/master/models/hific`
[3] *CDC*: `https://github.com/buggyyang/CDC_compression`

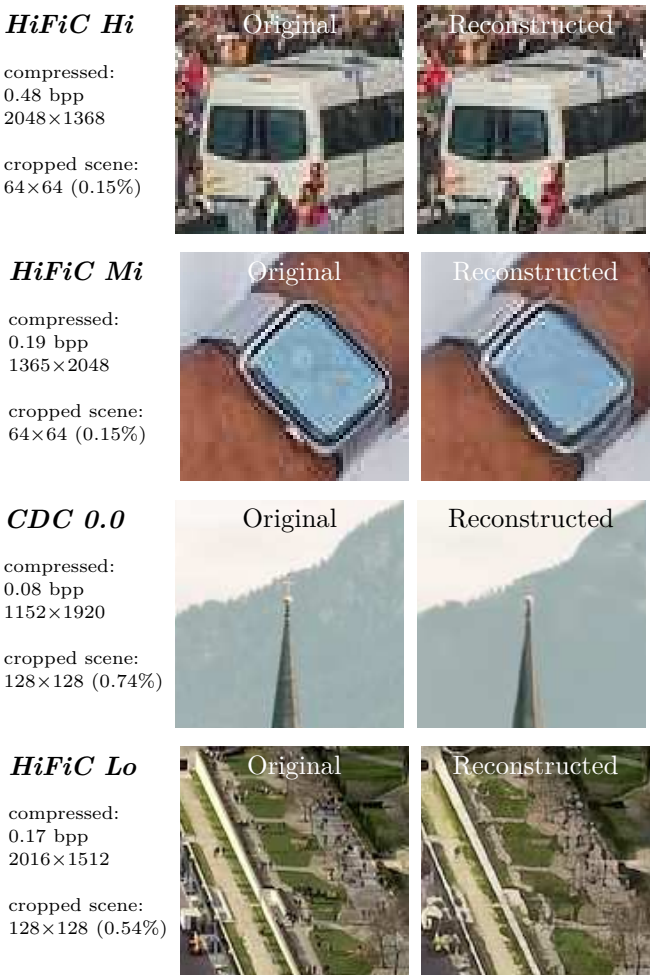


Figure A.3: Category AMPLITUDE: Reconstructions differ in the amplitude of spatial frequencies in the signal, affecting attributes such as brightness, color saturation, and the intensity of high frequency components.

A.3.2 Taxonomy

At a high level, our taxonomy separates the signal processing perspective ("What happens?") from the semantic impact ("How bad is the misinterpretation?"). Based on the apparent transformation of the signal, we define three categories. To illustrate each category we provide examples, cropped from compressed images of three datasets, and specify the compression model used, the bpp of the compressed image, the pixel dimensions of the original image, and the crop, as well as the percentage of the original represented by the displayed crop. The selected examples illustrate the characteristics of each type of miscompression. In practice, many miscompressions exhibit combined effects of multiple types.

Amplitude refers to reconstructions that differ in the amplitude of spatial frequencies in the image signal, such as changes in brightness, color saturation, or intensity of high frequency components. Attenuation seems to be more common, although we cannot rule out amplification. Unlike global signal processing operations, these effects tend to be local and content-dependent. Objects that we

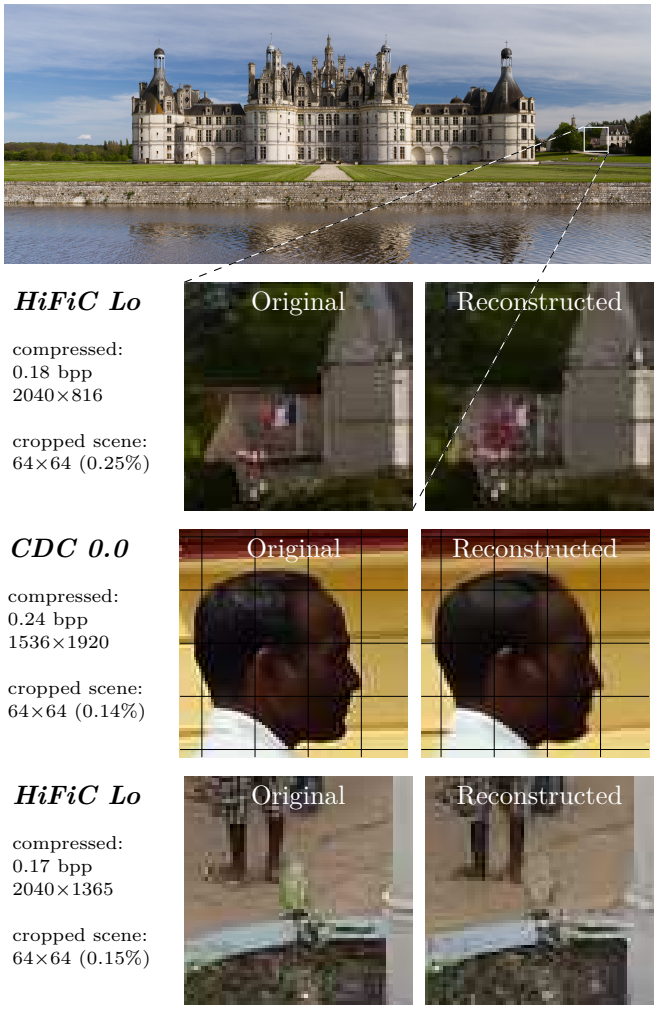


Figure A.4: Category GEOMETRY: Reconstructions contain geometric transformations, such as translation, rotation, scaling, and shearing, including shifted shapes and dissolved contours. (The top image illustrates the level of detail at which miscompressions occur. A grid was added to the middle crop.)

have found to be particularly susceptible to this type of miscompression include lights, colors of eye, hair, and skin, as well as birthmarks, and tattoos. Attenuation can result in altered colors or "disappearing" objects. Semantic changes occur when the amplitude carries meaning, as illustrated in the examples in Fig. A.3. In the top row, the fact that the car was braking, as indicated by the brake lights, is lost in the reconstruction. In the second row, a reconstructed watch appears to be turned off but is actually on and displaying the time. In the third row, the reconstructed image of a church tower does no longer include the Christian cross. In the bottom row, the reconstructed image of a park does not include the people present in the original image.

Geometry refers to reconstructions with geometric transformations such as translations, rotation, scaling, and shearing. This includes locally shifted shapes, dissolved contours, and imperfect representations of 3D scenes in 2D pixel matrices. Semantic changes occur when the geometry of an object carries semantic meaning, as illustrated in Fig. A.4. The top row shows a reconstruction of a

Figure A.5: Category SHAPE: Reconstructions contain changed contours.

Figure A.6: Miscompression of *symbols* such as body adornments or religious jewelry increase the risk of semantic misinterpretation.

flag that could be mistaken for graffiti on the wall in the background. The direction of the nose and chin of the person's profile in the second row is altered and differs from the original. The bottom row shows a leaf in the foreground that looks like a crack in the floor after reconstruction. Other susceptible objects, not illustrated here, include shadows and reflections. Semantic changes occur if the direction of a shadow changes and suggests a different positioning of the object casting the shadow, or if an object that is placed in front of a reflecting body of water or glass appears to be part of the reflection. This has implications for forensic methods that exploit inconsistencies in lighting direction and shadows for the detection of image manipulations [32, p. 14].

Shape refers to reconstructions that differ in shape, potentially caused by biases in the retrieval augmentation process. Semantic changes occur when the shape of an object conveys a semantic meaning, as illustrated in Fig. A.5. The top row shows a cropped image of a camera lens. The change in shape causes the number 8 in the original image to be mistaken as the number 6 in the reconstruction. In the bottom row, the shape of direction-specific traffic lights changes in the reconstruction from arrows to round lights. This results in a change of semantic meaning, as a round green light typically indicates that drivers can proceed in any direction.
Although we have observed several instances of altered textures, we do not consider them as a distinct type for now because the alterations did not align with our definition of miscompressions, *i.e.*, changes in semantic meaning. However, texture changes have been reported in the super-resolution literature [27, p. 25789], and we retain it as a potential extension to our taxonomy.

To classify the potential semantic impact of miscompressions across all categories, we define the SYMBOL modifier. The consequence of miscompressions is elevated when the affected objects portray *symbols*, *i.e.*, signs that carry specific meaning to human observers within a given social and cultural context [17]. Examples of symbols include letters, numerals, and signs, as well gestures, body adornments (*e.g.*, religious jewelry or clothing, wedding rings, etc.), traffic signs and lights, watch hands, logos, tattoos, graffiti, etc. This list is not exhaustive, and identifying a SYMBOL is subjective and can be challenging, especially without knowledge of the cultural and societal context of the captured scene. When symbols are involved, small changes in amplitude, geometry, or shape can completely alter the semantic meaning. For instance, the miscompression of a plant in the bottom row of Fig. A.4 is likely harmless, whereas missing jewelry, as shown in Fig. A.6 could lead to discord.
While screening our data, we have made a number of noteworthy qualitative observations. First, we found that miscompressions commonly occur in small image details (see top row in Fig. A.4), and a single image can contain multiple instances of miscompressions. Notably, not every image contains miscompressions. Images that depict single large objects against flat backgrounds are less susceptible. Moreover, we find that *CDC* is more likely to visibly destroy text, which reduces the risk of misinterpretation of incorrectly reconstructed text. In general, miscompressions occur seemingly unpredictably, and are difficult to distinguish from authentic image details.

A.4 Discussion

In this section, we briefly reflect on the decisions that have shaped our taxonomy, then outline how it can be applied, before discussing the wider implications of miscompressions and closing with selected avenues for future research.
The definition of miscompressions based on textual descriptions is naturally subjective. It depends on the observer with their experience and on the language, which defines concepts based on culture. Intersubjectivity can be improved by asking multiple observers [34]. The language dependency aligns the definition with relevance in the given cultural context.
The proposed taxonomy of miscompressions is a first qualitative step towards mitigations. The next step is to apply the taxonomy for the annotation of a larger dataset. This will pave the way towards quantifying the prevalence of miscompressions and identifying influencing factors with statistical methods. Importantly, such a dataset could be used to train models that detect and classify miscompressions, removing the human from the loop and allowing even greater scale.
Also image-to-text models can be useful tools to this end. To explore this option, we asked ChatGPT 4.0 to describe the original as well as the reconstructed image of the church tower in Fig. A.3. We presented it with a 256×256 crop (bigger than in Fig. A.3) that included the lower part of the tower and roofs of houses. ChatGPT's description changed from *"The image depicts a church steeple with a cross at the top, situated in a mountainous area. [...]"* for the original image to *"The image appears to be of a church steeple or a spire set against a mountainous backdrop. [...]"* But this approach did not work for all of our examples. For instance, there was no difference in the description of the watch in Fig. A.3. Using targeted object recognition methods to identify specific cases of miscompressions appears more promising, *e.g.*, the use of optical character recognition to identify miscompressed text.
Automatic detectors can be implemented as a safety net at encoding time to catch potential miscompressions. This would allow an increase in the number of bits allocated to areas of the image where miscompressions loom. The next step would be to use annotations of miscompressions as part of the training loss metric in order to harden future neural compression schemes against miscompressions. Our taxonomy can be used to tailor these metrics to the types of miscompressions

that should be avoided in a particular application, for example humans in the surveillance of public places and license plates in traffic surveillance. Conversely, neural compression could be tuned to *deliberately cause* miscompressions of, say, human faces as an integrated privacy-enhancing technology [22, 25, 28].

While research should strive to avoid miscompressions entirely, in the meantime it is crucial to deal with the existing risk in practice. It is imperative to acknowledge the existence of miscompressions and explain the associated risks of misunderstandings and false accusations to end users of the technology. Neural compression is not ready for use in safety and security critical applications, such as public surveillance or autonomous driving. The benefit of bandwidth savings is disproportionate to the risk of wrongful convictions and potentially fatal accidents. Worryingly, surveillance and autonomous driving are mentioned prominently in the motivation for the upcoming JPEG AI neural compression standard [5, p. 104]. In less critical applications, the use of neural compression should be documented. Suitable annotation could be stored in image metadata, where professionals, such as photo journalists and forensic investigators, can find and interpret them. A quantitative perceptual metric of miscompressions, similar to the metric for photo retouching [24], could be used in image captions, visible watermarks, or icons, and inform consumers about the potential presence of miscompressions. Reliable methods to detect neural compression are needed to enforce such policies [8, 9]. Interesting open research problems remain: Is it possible to detect instances of miscompressions in reconstructed images without access to the original? Can we develop forensic methods to distinguish uncontrolled miscompressions from malicious manipulations?

A.5 Conclusion

To our knowledge, this is the first study that compares multiple neural compression schemes for their susceptibility to produce semantically different reconstructions. We raise awareness of this novel problem in forensics, propose a provisional taxonomy of what we call *miscompressions*, and support it with existential evidence. We hope that as this taxonomy develops, it will enable quantitative studies of automated detection, prevalence, influencing factors, and mitigations.

Acknowledgment

We thank Benedikt Lorch, Martin Beneš, Judith Senn, and Verena Lachner for valuable discussions and comments. Computational results were achieved using the LEO HPC infrastructure at the University of Innsbruck.

References

[1] Boston Bombing Day 2, Apr 2016. `https://abcnews.go.com/US/boston-bombing-day-improbable-story-authorities-found-bombers/story?id=38375726`, last accessed: July 2024.

[2] Dzhokhar A. Tsarnaev, Mar 2024. `https://www.justice.gov/usao-ma/tsarnaev-exhibits-day-2` exh_29.pdf, last accessed: July 2024.

[3] E. Agustsson and R. Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *CVPR*, 2017.

[4] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, pages 90–93, 1974.

[5] J. Ascenso, E. Alshina, and T. Ebrahimi. The JPEG AI standard: providing efficient human and machine visual data consumption. *IEEE MultiMedia*, pages 100–111, 2023.

[6] J. Ballé, V. Laparra, and E. Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

[7] J. Ballé, D. Minnen, S. Singh, S. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. In *ICLR*, 2018.

[8] S. Bergmann, D. Moussa, F. Brand, A. Kaup, and C. Riess. Frequency-domain analysis of traces for the detection of AI-based compression. In *IWBF*, pages 1–6. IEEE, 2023.

[9] S. Bergmann, D. Moussa, F. Brand, A. Kaup, and C. Riess. Forensic analysis of AI-compression traces in spatial and frequency domain. *Pattern Recognit. Lett.*, pages 41–47, 2024.

[10] A. Berthet and J. Dugelay. AI-based compression: A new unintended counter attack on JPEG-related image forensic detectors? In *ICIP*, pages 3426–3430. IEEE, 2022.

[11] Alexandre Berthet, Chiara Galdi, and Jean-Luc Dugelay. On the impact of AI-based compression on deep learning-based source social network identification. In *MMSP*, pages 1–6. IEEE, 2023.

[12] S. Bhardwaj, I. Fischer, J. Ballé, and T. Chinen. An unsupervised information-theoretic perceptual quality metric. In *NeurIPS*, pages 13–24, 2020.

[13] T. Chen and Z. Ma. Towards robust neural image compression: Adversarial attack and model finetuning. *TCSVT*, 2023.

[14] K. Ding, K. Ma, S. Wang, and E. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *TPAMI*, pages 2567–2581, 2020.

[15] Z. Duan, M. Lu, Z. Ma, and F. Zhu. Opening the black box of learned image coders. In *PCS*, pages 73–77. IEEE, 2022.

[16] R. Franzen. Kodak photocd dataset, Nov 1999.

[17] A. Frutiger. *Signs and Symbols*. Weiss Verlag, 1989.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *NeurIPS*, 2014.

[19] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, pages 6840–6851, 2020.

[20] C. Hudelot, N. Maillot, and M. Thonnat. Symbol grounding for semantic image interpretation: From image data to semantics. In *ICCV*, pages 1875–1875. IEEE, 2005.

[21] D. Huffman. A method for the construction of minimum-redundancy codes. *IRE*, pages 1098–1101, 1952.

[22] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis. Face/off: Preventing privacy leakage from photos in social networks. In *CCS*, pages 781–792. ACM, 2015.

[23] E. Jalilian, H. Hofbauer, and A. Uhl. Iris image compression using deep convolutional neural networks. *Sensors*, 22(7), 2022.

[24] E. Kee and H. Farid. A perceptual metric for photo retouching. *PNAS*, pages 19907–19912, 2011.

[25] K. Lander, V. Bruce, and H. Hill. Evaluating the effectiveness of pixelation and blurring on masking the identity of familiar faces. *JARMAC*, pages 101–116, 2001.

[26] A. Lewis and M. Kuhn. Towards copy-evident JPEG images. *GI Jahrestagung*, pages 1582–1591, 2009.

[27] B. Li, X. Li, H. Zhu, Y. Jin, R. Feng, Z. Zhang, and Z. Chen. SeD: Semantic-aware discriminator for image super-resolution. In *IEEE/CVF CVPR*, pages 25784–25795, 2024.

[28] Y. Li, N. Vishwamitra, B. Knijnenburg, H. Hu, and K. Caine. Effectiveness and users' experience of obfuscation as a privacy-enhancing technology for sharing photos. *PACM HCI*, pages 1–24, 2017.

[29] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

[30] F. Mentzer, G. Toderici, M. Tschannen, and E. Agustsson. High-fidelity generative image compression. *NeurIPS*, 2020.

[31] D. Minnen and S. Singh. Channel-wise autoregressive entropy models for learned image compression. In *ICIP*, pages 3339–3343. IEEE, 2020.

[32] A. Piva. An overview on image forensics. *International Scholarly Research Notices*, page 496701, 2013.

[33] C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, pages 379–423, 1948.

[34] A. Smaling. Varieties of methodological intersubjectivity - the relations with qualitative and quantitative research, and with objectivity. *Quality and Quantity*, pages 169–180, 1992.

[35] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265. PMLR, 2015.

[36] L. Theis, W. Shi, Q. Cunningham, and F. Huszár. Lossy image compression with compressive autoencoders. *ICLR*, 2017.

[37] G. Toderici, W. Shi, R. Timofte, L. Theis, J. Balle, E. Agustsson, N. Johnston, and F. Mentzer. Workshop and challenge on learned image compression (clic2020). In *CVPR*, 2020.

[38] G. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, pages 30–44, 1991.

[39] R. Yang and S. Mandt. Lossy image compression with conditional diffusion models. *NeurIPS*, 2024.

[40] Y. Yang, S. Mandt, and L. Theis. An introduction to neural data compression. *Found. Trends Comput. Graph. Vis.*, pages 113–200, 2023.

B. User Perceptions of Miscompressions

Authors

Nora Hofer, University of Innsbruck
Rainer Böhme, University of Innsbruck

Title

When the Codec Hallucinates: User Perceptions of Miscompressed Images

Conference

ACM CHI Conference on Human Factors in Computing Systems (CHI '26)
Barcelona, Spain · April, 13–17, 2026

Abstract

People exchange images every day. New methods for image compression leverage neural networks to save bandwidth, but they can undermine the semantic integrity. The term *miscompression* refers to unintended semantic changes of image details, introduced by generative AI during neural (de)compression. Although prior work has speculated about the resulting risks, no empirical evidence exists on how people perceive these novel compression artifacts. In this study, 115 human subjects compared original images with conventionally compressed, neurally compressed, and miscompressed images. Participants perceive that miscompressions elevate the risk of misunderstandings when communicating with images. They also frequently attribute miscompressions to intentional editing, whereas conventional JPEG artifacts are more often recognized as distortions. This paper proposes a method to study this new phenomenon, provides the first empirical evidence of user perceptions of miscompressions, and derives implications for trust in images, as well as interface designs that mitigate the risk.

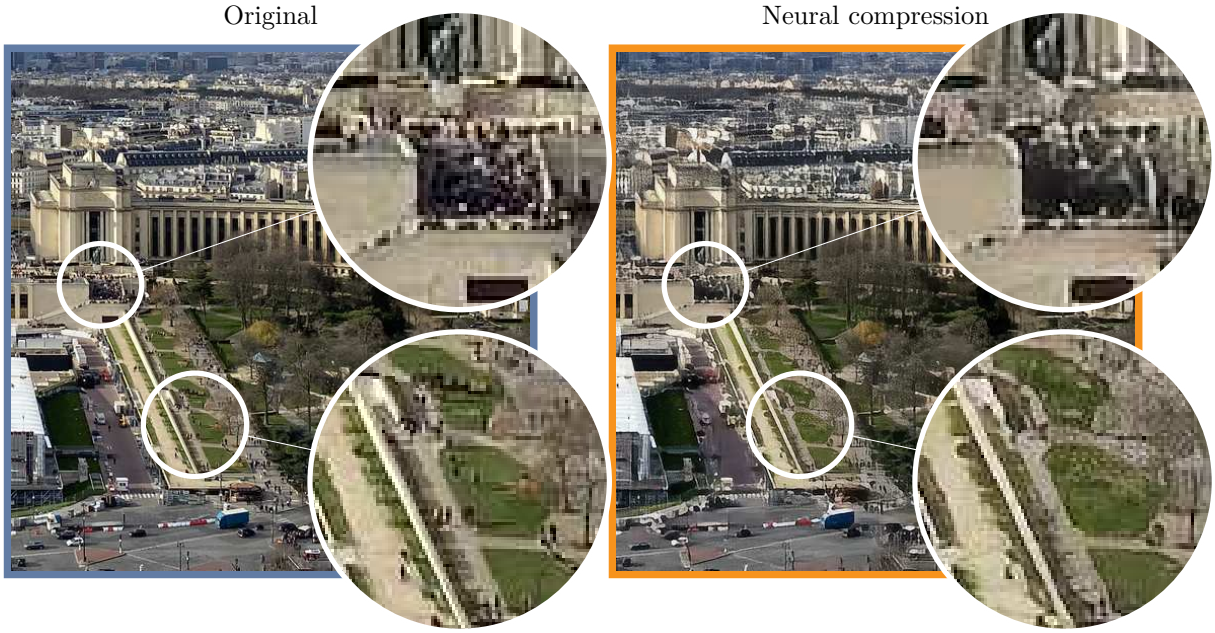


Figure B.1: M-Paris. Neural image compression can introduce subtle but sometimes semantically relevant changes to image details. Here, the crowd on the stairs is no longer recognizable and people on the grass disappear (after compression with HiFiC [62] at 0.17 bits per pixel). In our fictional introduction story, such *miscompressions* lead to misunderstandings. In this paper, we study how people perceive these novel compression artifacts. Rather than recognizing them as distortions, participants often interpret them as intentional edits and report an elevated risk of misunderstandings. All figures in this paper are best viewed on screen and in color.

B.1 Introduction

Imagine Cassandre, a photojournalist, who covers a breaking news story about a protest in the Trocadéro district of Paris. She takes photos of the barriers, the police presence, and the crowds heading towards the site. She uses a messenger app to send the photos to her colleague Tiresias, who is writing the story in the London newsroom. Cassandre later reads the published article and is shocked. Tiresias describes a small crowd and does not mention the police. The public is outraged, and activists accuse them of misrepresentation. Confused, Tiresias insists that he reported exactly what he saw in the photos. When they compare them with Cassandre's originals, they realize that the versions received are different from the ones sent. What was supposed to be an objective report has become a controversy, caused by *neural image compression*. Our example in Figure B.1 suggests that the next generation of image compression methods may make this fictional story a reality.

Compression is essential for the efficient transmission and storage of digital images. The most common methods are lossy. This means they remove information imperceptible to the human eye in order to reduce the file size. While JPEG, a standard from the 1990s, still dominates the web [22], researchers have turned their attention to *neural* image compression. The idea is to replace conventional signal processing operators in the image compression and decompression pipeline with trained neural networks. Early proposals leverage variational autoencoders [6, 7]. State-of-the-art methods use image transformers [5, 91] for the encoder and generative adversarial networks [62] (GANs) or diffusion models [88] for the decoder. These methods achieve unprecedented compression

rates at comparable or superior perceptual quality. Wide deployment in consumer devices may be just a matter of time,[1] thanks to new standards like JPEG AI [5].

This development raises concern. Such low bitrates can inconspicuously undermine the integrity of image details. Recently, the signal processing literature has proposed the term *miscompressions* [38] to describe semantic changes of image details caused by neural compression. Conventional methods, such as JPEG, often introduce visible indicators of compression, like blocking [68], blurring [60], or ringing [31, 36] that allow viewers to judge the reliability of an image. Neurally compressed images, however, lack such indicators and tend to appear visually flawless. Thereby they can create a false sense of trust.

While the cause of miscompressions is technical, the consequences are social and may pose risks to humans in various ways. First, miscompressions can lead to the uncontrolled and unintended spread of misinformation, especially when the reconstruction is realistic [46]. Second, miscompressions can change the semantics of images in a way that resembles intentional editing. Therefore, comparisons between the original and the reconstructed images, *e.g.*, in court, by the media, or by insurers, could result in false accusations. A recent CHI paper discovered that viewing images enhanced by artificial intelligence (AI) can alter one's memory of a scene [70]. Eyewitnesses to a scene may fall victim to this effect as miscompressions can resemble the semantic changes of AI enhancements. Third, the intended applications of neural compression include downstream computer vision tasks in critical domains, such as public surveillance and autonomous driving [5, p. 103]. Clearly, potential classification errors can cause severe and irreversible harm.

Researching miscompressions is difficult because the definition of a semantic change and its severity are subjective. The existing examples are based on the subjective view of a few researchers [38, 72]. However, semantic understanding differs between individuals, based on different experiences and cultural backgrounds. To illustrate this point, not all authors are in full agreement on how unexpected and severe the missing people in Figure B.1 are. This calls for the involvement of a broader set of users to reduce the reliance on individual subjective opinions.

The research question of our empirical study is to find out if a wider population perceives miscompressions of state-of-the-art neural compression methods as concerning. We also want to understand whether the aforementioned risk of miscompressions being mistaken for intentional image editing or manipulation is supported by real users. Finally, we want to measure whether the users are familiar with the visible distortions produced by conventional lossy compression and can attribute the differences to JPEG artifacts. We operationalize the research question in three hypotheses that can be tested with quantitative methods:

H1 Miscompressed images are *more likely* to cause misunderstandings than other similar images.

H2 The differences between a miscompression and its original are *more likely* attributed to intentional editing than for other similar images.

H3 The differences between a miscompression and its original are *less likely* attributed to uncontrollable distortion than for other similar images.

The term "other similar images" refers to images representing the same scene that have been compressed using conventional JPEG or other neural compression codecs which do not result in miscompression for this input. Note that H2 and H3 are not two ends of the same continuum. The hypotheses are conceptually different as they relate to different causes (intention vs. accident), op-

[1] "[T]he first ever implementation [...] of JPEG AI encoder and decoder on their mobile phone" `https://www.linkedin.com/posts/touradjebrahimi_wearejpeg-activity-7346065622880976896-Izoh` (posted: July 2025; accessed: August 2025)

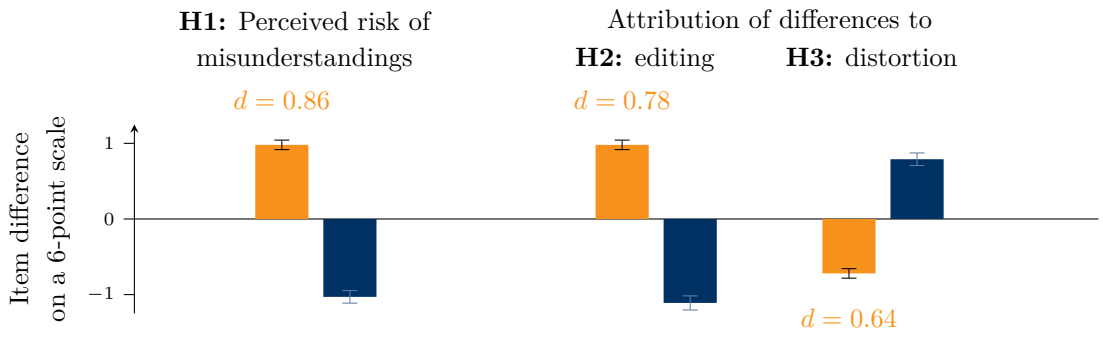


Figure B.2: Users perceive that miscompressions (orange bars, left) carry an increased risk of causing misunderstandings. They attribute differences between originals and reconstructions more to intentional editing and less to uncontrollable distortion. The same metrics for JPEG images (blue bars, right) have the opposite direction. Standard errors and Cohen's d for effect size are shown. All results are statistically significant at the $p < 0.001$ level after accounting for the panel data structure with subject and image fixed effects.

erations (editing vs. compression), and actors (human vs. machine). Moreover, they are neither mutually exclusive, as images can be edited and compressed, nor complete, as images can differ for other reasons, *e.g.*, slightly different camera angle or acquisition time.

To test the hypotheses, we collect standardized responses in a lab study on a sample of 115 users, presenting them a curated set of stimulus images compressed with state-of-the-art neural compression codecs. Using a combined between-subjects and within-subjects design, we get a total of 1 380 image views and 1 131 ratings on image differences. Figure B.2 shows the results in a nutshell, supporting all three hypotheses. In summary, we make the following contributions. i) We design and test an empirical method for a user study on a new phenomenon which can serve as a baseline for future studies on miscompressions. ii) We present evidence confirming that miscompressions are perceived differently from conventional compression artifacts and that they increase the risk of misunderstandings. iii) We discuss implications for the design of future interfaces that enable users to assess and react to the risks posed by miscompressions.

The remainder of this paper is structured as follows. Section B.2 reviews research on user perception of digital images. Section B.3 documents our method and instrument, Section B.4 presents the results, both aggregated and broken down by images. Section B.5 discusses the findings and derives implications for interface designs. Lastly, Section B.6 concludes.

B.2 Related Work

The perception of digital images has long been of interest to the research communities of human-computer interaction (HCI), signal processing, and imaging. However, no comprehensive theoretical framework exists: "people's perception of image distortion is complex" [66, p. 3235]. Intervening factors include the image content and the viewer's cultural background. Here, we highlight selected human-factor works that have influenced our design or will help to interpret our results. A summary of the technology behind neural image compression is provided in Appendix B.1.1.

B.2.1 User Involvement in Image Compression Research

The role of user studies in the lossy image compression literature is to measure how humans perceive the quality of images after compression. It distinguishes between subjective and objective image quality metrics.

Subjective metrics are collected by asking human observers. They are extensively researched and standards exist [41, 42, 43]. In *reference studies*, human observers compare the original (reference) to the compressed image. In *no-reference studies*, the observers are asked to judge the image quality of the compressed image alone. A popular method is called two-alternative forced-choice [33] (2AFC). Participants are shown two images and asked to choose their preferred one. In no-reference studies, 2AFC is applied using different quality settings or codecs.

Objective metrics are mathematically defined. The spectrum ranges from simple distance metrics, such as the signal-to-noise ratio, to perceptually weighted reference metrics, such as SSIM [84], MS-SSIM [85], and FSIM [89]. Recently, learning-based metrics have gained popularity as both reference and, predominantly, no-reference quality metrics [35, 90]. They are part of the loss functions used to train generative AI and neural compression models. While technically "objective", their learned parameters depend on human input, trying to mimic "subjective" human perception. This alignment remains challenging and is an active field of research [14, 34, 71]. One possible cause of miscompressions is the overemphasis on such no-reference quality metrics that improve realism at the expense of fidelity.

In the neural compression literature, user studies typically accompany proposals for new codecs [62, 73]. Their goal is to demonstrate the codec's performance rather than understanding people's perception of neurally compressed images. They are often based on a a small number of viewers and images, and do not include control variables. Semantic differences are acknowledged occasionally, *e.g.*, [73, p. 316], [62, p. 10], but we are not aware of any user study investigating human perception of miscompressions. Tsereh et al. [82] present a dataset of machine-detected JPEG AI compression artifacts with crowdsourced human verification. Their focus is to improve compression performance rather than safeguarding semantic integrity.

B.2.2 Users' Ability to Detect Authentic Images

An image is considered *not* authentic if the original has been edited to alter its semantics, or if it has been entirely generated by AI [15, Fig. 2]. Therefore, we review the literature on human performance in detecting *image editing*, which relates to our Hypothesis 1, and on detecting *AI-generated images*, since neural compression relies on the same technology.

Image editing Ostrovsky et al. [67] explore how different lighting configurations in the image influence participants' ability to detect irregularities. Farid and Bravo [24] study the ability to detect forgeries based on irregularities in geometric shades and reflections. Carvalho et al. [21] involve human subjects to validate a forensic forgery detection method based on color classification of scene illuminants. Sun et al. [77] propose a dataset to benchmark the detectability of AI editing operators and involve 35 human subjects to assess the task difficulty. Schetinger et al. [75] crowdsource by asking 400 non-experts to localize suspected forgeries in images. Their participants often relied on contextual cues. Relevant for the selection of our stimuli, they find that images with high structural complexity are harder to evaluate, whereas the size of the edited area does not matter. Experience with digital imaging improved detection capability. Across these studies, one consistent finding emerges: humans' ability to detect image editing is poor, with detection accuracies between 40 and 60 %.

AI image generation Early image generation technologies were still detectable by humans, as demonstrated repeatedly by Farid and colleagues [23, 25, 40, 58]. A turning point came with advances in deep learning, especially GANs [30]. Lago et al. [55] crowdsource a comparison of GAN-based face generators. The participants were not only unable to detect generated images, but they also misclassified generated images as real more often than the real control images. Follow-up studies have confirmed this negative result and investigated influencing factors. Nightingale and Farid [65] vary the gender and ethnicity of the stimuli and find that white male faces are the least distinguishable, presumably due to biases in the GAN's training data. Frank et al. [27] test a subset of the same face images on a representative sample, confirming the results obtained from convenience samples. Mink et al. [63] explore the effect of identity-based biases. They find that viewers are better at detecting artificial faces if they share the gender or racial identity of the portrayed subject. Wöhler et al. [87] use eye tracking to study the cues participants use to detect face swapping in videos. They find that participants decide based on artifacts, such as blur or unnatural expressions and eye movements. Lu et al. [56] generalize the experiment from faces to all kinds of AI-generated images and ask participants to select from eight cues that may have influenced their judgement. "Detail" and "smoothing", two artifacts common in neural compression, are mentioned most frequently. They also control for participants' experience with generative AI, but the effect is not significant. Finally, Kamali et al. [51] present the probably most comprehensive study. Their collection of 750k data points from 50k subjects, including 35k qualitative comments, allows them to dig deep into potential cues people use to make a decision. General image quality is mentioned most often as a cue, supporting the concern that high realism may create a false sense of trust [46]. Some of their findings may transfer to the perception of miscompressions. Note that neither Lu et al. [56] nor Kamali et al. [51] consider conventional JPEG compression artifacts as cues, which highlights the novelty of our Hypothesis 3.

B.2.3 Risks of Semantically Distorted Images

The authenticity of images and their effect on the credibility of information has been studied for decades, *e.g.*, for news articles [32, 64], web pages [52], and user-generated content [29, 46, 63]. In the 1990s, researchers even suspected that a mere demonstration of Photoshop, an image editor, could erode viewers' trust in images [54]. While this hypothesis was rejected, it reinforces the idea that experience with editing technologies should be a control variable.

All this research focuses on intentional and controlled edits to the semantics of an image. Miscompressions are a new phenomenon. They compromise the semantic integrity in so far unpredictable and undetectable ways. Research on miscompressions is sparse. Hofer and Böhme define them as discrepancies "between the semantic meaning of an original image (detail) and its reconstructed version after neural compression." [38, p. 3] and collect a human-annotated dataset of miscompressions from different compression codecs [39]. Agustsson et al. [2] propose a technical remedy. Their decoder has a parameter to select between blurry outputs that are closer to the input, and outputs with synthesized image details that are more realistic. Qiu et al. [72] take a closer look at one specific type of miscompression, revealing that the semantic distortions are subject to bias: African–American faces are commonly reconstructed to appear more Caucasian, while Caucasian faces largely retain their original features. The authors demonstrate the bias using a few examples and measure it by passing reconstructed test images to a learned phenotype classifier.

The machine learning community has studied potential challenges for neural compression, regardless of their impact on visible semantics. Chen and Ma [18] warn that malicious actors could scramble image content by exploiting a vulnerability to adversarial perturbations. Madden et al. [57] show that

it is possible to gain control over the output by triggering bitstream collisions. Both attacks require control over the input image and detailed information on the codec and its implementation. In non-malicious use cases, the performance of downstream computer vision tasks may degrade [10, 45, 59], especially for iris recognition [10]. Learning-based image forensics can also be affected, including detectors of image manipulation [11, 16, 17], provenance [12], and deepfakes [9, 16]. By contrast, Cardenuto et al. [17] report that the evidential value of medical images in science does not deteriorate at an equal bitrate compared to conventional compression. All of these works process images with machines and do not involve any human viewers. Because the risks of semantically distorted images may affect human perception more than machines, a study on the users' perspective appears overdue.

B.3 Method

To our knowledge, this is the first user study on miscompressions. Our aim is to validate whether the concept of miscompressions used by researchers matches the understanding of users. To this end, we collected ratings of multiple human subjects on images depicting multiple scenes. This allows us to generalize from the subjective understanding of individuals as well as from the distinctive characteristics of a scene. As miscompressions are defined by differences between images, we opted for a **full-reference study** with one test image displayed next to the reference image at a time. To ensure external validity while not requiring to introduce our subjects to the topic of neural compression, we came up with a **scenario** set in the context of social media. This scenario is handy because it is plausible for an image to undergo unknown processing in transit, including lossy compression, retouching, or manipulation. To eliminate any confounding effect of the display device or light conditions [83], we conducted the study in a **controlled lab environment**. To make the effect of miscompressions measurable, we included test images that were not miscompressed for comparison. We created these **control images** of the same scenes by compressing the source images with a different neural compression codec or JPEG. This allows us to control for the effect of scene content (using scene fixed effects in the analysis). At the same time, we wanted to ensure that each subject saw each scene only once. This requires a **between-subjects** design. To increase the number of ratings and the diversity of scenes, we have combined it with a **within-subject** design (making subject fixed effects necessary to account for repeated measurements).
Our power estimation suggested that we should have at least 15 ratings for each image pair.[2] As we could not expect every subject to spot all the differences, we added a margin and aimed for 25 views of each image pair. Our pretests revealed that twelve ratings per subject would fit into the time frame of 15 minutes. With a conservative estimate of 100 participants, we decided to have four groups. Section B.3.2 provides details on the stimulus selection and placement within the instrument.

B.3.1 Instrument

Our instrument had four parts: introduction, demographics, image comparisons, and control variables.

Part 1: Introduction After receiving a briefing and signing a consent form, participants entered the instrument. The start page informed them that the study aimed to measure their perception

[2]Pre-data, our rationale was to not miss a "large" ($f = 0.36$) effect with more than 50 % probability at the $p \leq 0.05$ significance level when analysed individually (Section B.4.3). Since we had multiple image pairs, we could tolerate missing effects in half of them. We used the `R` package `pwr` to calculate the sample size for a balanced one-way analysis in two groups (miscompressed vs other similar image).

of distortions in digital images introduced during the transmission over the internet, and that they would compare pairs of images and answer questions about the differences. They were not told about neural image compression or miscompressions.

Part 2: Demographics We asked for the participants' age, gender identity, and whether they had any visual impairment, and used visual aids during the study.

Part 3: Image comparisons In the main part of the instrument, participants were asked to imagine a scenario where they had taken an image and uploaded it to a social media platform. The image had gone viral, and another person discovered it in their feed on another social media platform. Two images were displayed next to each other on the same screen, the original (reference image) on the left and the received image (test image) on the right. Figure B.3 shows a screenshot of the stimulus presentation in the instrument. The same page stated that the two images were *not* identical and asked whether participants could see at least one difference. If the answer was "no", the study proceeded to the next image pair.
Participants who responded with "yes" were asked three follow-up questions. To investigate Hypothesis 1, we asked whether the differences could lead to misunderstandings between them and the person receiving the image. We deliberately left the terms "difference" and "misunderstanding" vague because we feared that providing a definition with examples could bias the interpretation. Participants could express their certainty on a rating scale annotated with the labels "certainly", "very likely", "likely", "unlikely", "very unlikely", and "certainly not". We chose six points to prevent undecided respondents from choosing a neutral midpoint [13]. To investigate Hypothesis 2, we asked participants whether they would attribute the differences to intentional editing, mentioning retouching, filters, or manipulation as examples. We recorded their answer on the same scale as before. To test Hypothesis 3, we asked whether they would attribute the differences to uncontrolled distortions, using transmission errors or compression as examples, again on the same scale. The description of the scenario and the two images remained visible for all three questions.
After the first image pair, we informed the participants that the same scenario will be repeated for multiple images. We also reminded them to stay focussed and examine each pair carefully. We did not mention the number of image pairs (twelve), but a progress bar gave indications.

Part 4: Control variables The literature documents various factors that may influence image perception, such as previous experience with digital image processing [47], photography [75], generative AI [46], and media literacy [46, 49]. We therefore included control questions for those variables, asking about the participants' previous exposure to generative and conventional image processing technologies and how often they verify the authenticity of images online.
To assess the external validity of the data collected in our hypothetical scenario, we also asked participants about their experience with image sharing and how realistic they thought the scenario was. The study concluded with an open-text field for feedback on image selection, question clarity, and any difficulties encountered.

The instrument was implemented using an online survey tool[3] hosted by the university. We disabled navigation and recorded responses as well as response times. All questions required responses, except for the final open-text feedback question. As display time restrictions can impair participants' ability to detect anomalies in generated images [51], we did not impose any time limits.
An English translation of the original German questionnaire can be found in Appendix B.2.2.

[3] `https://www.limesurvey.org/`

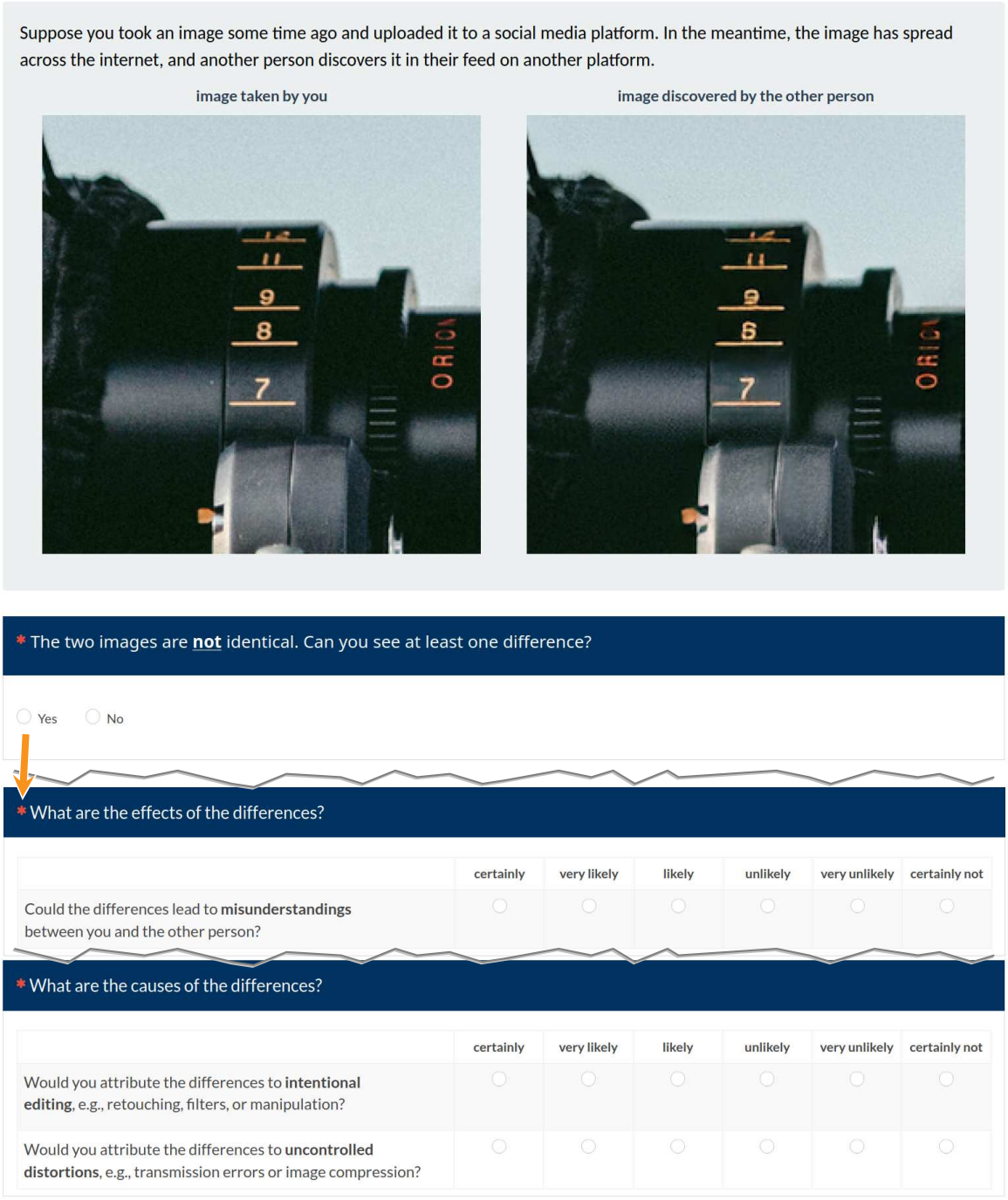


Figure B.3: Screenshot of the standardized stimulus presentation (translated from German). The images remained displayed above of all three question blocks. The last two blocks were skipped if the first question was answered "no." The stimulus here is CAMERA: the original on the left, and the miscompression on the right, where the engraved number 8 has been altered to a 6.

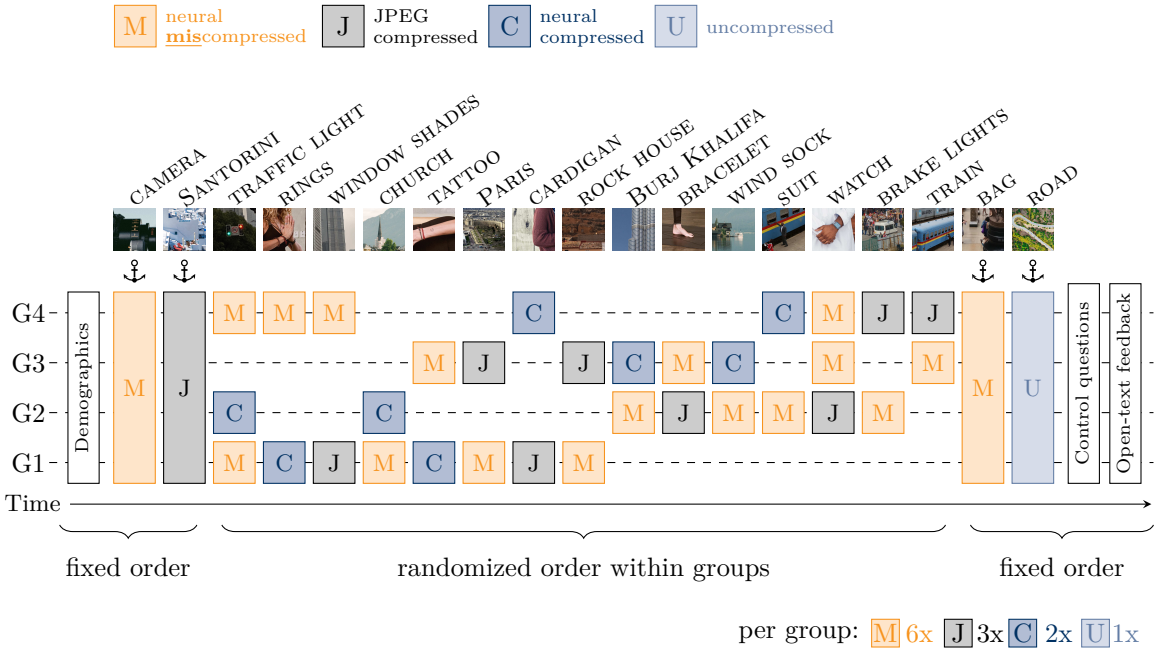


Figure B.4: Flowchart of the instrument. Participants were split into four groups, G1–G4, depicted in rows. Image pairs (in columns) were assigned so that each group viewed six control images and six miscompressed images. The color of a box indicates in which version a test image was shown to all members of the respective group. For example, WINDOW SHADES was shown to Group 1 as JPEG and to Group 4 as miscompression. Groups 2 and 3 have not rated this stimulus. The order of the presentation was randomized for each participant in the indicated range. All groups viewed the remaining four anchor image pairs (positionally fixed as the first and last two pairs). The uncompressed anchor image pair at the end was an attention check.

B.3.2 Stimuli

Because miscompressions are rare, and no method exists to deliberately craft targeted examples, we used four state-of-the-art neural compression codecs [5, 7, 62, 88], including the reference model for the upcoming JPEG AI standard, to compress and then manually inspect images that are suitable as stimuli. We use the definition of miscompressions in the literature, which requires that a human observer would use a different verbal description for the compressed image than for the original [38, p. 3]. We report all details of the compression, including codecs, settings, and bit rates, in Appendix B.2.3 and Table B.2.

Selection From a shortlist of manually collected miscompressions, we selected 19 scenes using criteria designed to ensure diversity and maintain participant engagement. Specifically, we varied image properties and content known to influence image perception [24, 51, 67, 81], including texture, scene complexity, detail, global brightness, contrast, structural complexity, and the presence of dominant edges or flat areas. We also varied indoor and outdoor environments of differing perspectives (near objects, wider scenes). The selected scenes cover objects, buildings, vehicles, identifiable and non-identifiable persons, and body parts. We further varied the *types* [38] and *severity* of miscompressions, with severity determined by the authors' subjective perception. For instance, we include a miscompression where a crescent moon tattoo is turned into a full moon tattoo (M-TATTOO) as a severe example, and a miscompression where open window shades appear closed (M-WINDOW

SHADES) as a less severe example. We also included cases of miscompressions of objects with strong semantic meaning. Examples are the miscompression where the number 8 is turned into the number 6 on a camera objective (M-CAMERA), and the arrow-shaped traffic light that is turned into a regular round traffic light (M-TRAFFIC LIGHT (ARROW)). For contrast, one scene was intentionally chosen for its semantic irrelevance (CARDIGAN). All test images can be viewed in Appendix B.2.5 and are provided in the supplemental material.

Placement in the instrument Figure B.4 documents how and in which order the stimuli were assigned to the four groups, thereby balancing content, properties, and control type. Boxes with letters indicate which test image the respective group compared to the original reference image. We included six control images and six miscompressions (M) per group. The control images consisted of three JPEGs (**J**), two neurally compressed images (**C**) that were not miscompressed, and one uncompressed test image (U) that did not differ from the reference, as an attention check.
While selecting the stimuli, we found two instances of multiple miscompressions from different neural compression codecs for the same scene. The first scene is TRAFFIC LIGHT, with the modified shape (M-TRAFFIC LIGHT (ARROW)) and a "No Cars" sign, that resembles a "No Photography" sign (M-TRAFFIC LIGHT (SIGN)). The second scene is WATCH. In one version the smartwatch display appears to be turned off (M-WATCH (OFF)) and in the other version, the watch appears physically broken (M-WATCH (BROKEN)). We decided to include both versions to gain insights into different forms of miscompressions for the same scene. The semantically irrelevant stimuli, CARDIGAN, is the only scene without a corresponding miscompressed variant. This resulted in a total of 36 image pairs.
To better compare responses between groups, we selected four *anchor images* that were identical for all groups (*cf.* anchor symbols in Fig. B.4). Two anchor images were placed at the beginning to give all groups an equal opportunity to familiarize themselves with the task. The first anchor (M-CAMERA) was a motivating miscompression and the second (J-SANTORINI) was a JPEG control image containing visible compression artifacts. The two last images were again anchor images with a miscompression (M-BAG) and the uncompressed control image (U-ROAD). We decided to place U-ROAD at the very end, as we were concerned that participants might get demotivated when they could not find a single difference. Between the anchors, all participants in the same group viewed the same images in random order, to reduce potential bias caused by order or fatigue effects.

Presentation All participants used identical desktop computers with 23-inch monitors (1920×1080 resolution) and accessed the instrument via Firefox. The images were displayed at 512^2 pixels (13.5 cm side length). As some miscompressions were very small (*e.g.*, ROCK HOUSE, BRAKE LIGHTS), we used smaller crops (128^2 or 256^2 pixels) and scaled them up to 512^2 pixels with nearest neighbor upsampling to ensure constant size and avoid uncontrolled upsampling by the browser. The increasing visibility of individual pixels also signals a low resolution to the participant.

B.3.3 Statistical Analysis

We test our main hypotheses by fitting linear regressions with the ordinary least squares method. The specifications we consider take the form,

$$y_{i,k} = b_0 + b_1 \cdot x_{\text{mc},k} + b_2 \cdot x_{\text{jpeg},k} + \cdots + d_i + s_j + \varepsilon_{i,k} \ ,$$

where $y_{i,k}$ is the rating of the i-th subject on the k-th image pair in the range $\{1, \ldots, 6\}$, $x_{\text{mc},k} \in \{0, 1\}$ is an indicator for miscompressions, $x_{\text{jpeg},k} \in \{0, 1\}$ is an indicator for control images with

conventional JPEG compression, "..." are placeholders for additional control variables, and $\varepsilon_{i,k}$ is the residual. The coefficients d_i and s_j are the estimated subject and scene fixed effects, respectively, and b_l are the estimated coefficients we report and interpret. Control images that were compressed with neural compression have $x_{\text{mc},k} = x_{\text{jpeg},k} = 0$. The fixed effects aim to capture the panel structure in our mixed within and between-subject design, reducing the likelihood that the residuals are unduly correlated (as confirmed by regression diagnostics). All anchor images share one scene fixed effect to prevent collinearity. We report four specifications of three dependent variables, one for each hypothesis: the perceived risk of misunderstanding (H1), the perceived likelihood of intentional editing (H2), and the perceived likelihood of uncontrollable distortion (H3). The specifications differ in which coefficients are forced to zero. We exclude image pairs where the subject did not report seeing any differences.

We use Cohen's d [19] to report the effect size. The numerator is the estimated mean difference, b_1, and the pooled standard deviation in the denominator is calculated from the residuals.

B.3.4 Ethics and Data Protection

The study was IRB approved. All participants consented to their participation and to the collection and processing of their personal data for the purpose of scientific research. They also agreed to the publication of the data in a way that would not allow them to be identified. Participants were informed that participation was voluntary and that they could withdraw from the study at any time. After completing the study, we offered them a printed debriefing sheet to inform them about the research project. They also had the option of leaving a contact address to be informed of the study results. Participants received no financial compensation. When selecting stimuli, we took into account potential triggers of negative emotions in order to avoid any psychological or social harm.

B.3.5 Limitations

Our method has limitations. First, our sample of participants consists of German-speaking undergraduate computer science students. Although their perception of the risk of misunderstanding and attribution of image differences may not be representative of the general population, this sample is arguably similar to early adopters of new technology. Moreover, previous research found that the cultural background can influence the perception of image distortion [66]. Future research should consider a more heterogeneous sample. Second, our stimuli are hand-selected. While there is no commonly agreed way of sampling representative images, our selection strategy was geared towards showcasing a spectrum of miscompressions of varying type, visibility, and severity (according to the authors' subjective perception). The main results are averages over all scenes, and could have been much stronger if we had included more scenes like Rock house or Watch, or much weaker if all our scenes were like Window shades. We report breakdowns and interpret individual scenes to increase confidence in our findings. Third, we did not collect which specific differences participants noticed and referred to. We considered asking for a written description of the detected differences [51, 75], but decided against it for fear of fatigue effects. We also considered visually highlighting predefined differences in the images [70] or providing hints, but also decided against this, as it has been shown to influence decisions [75]. There may also be hidden limitations that will only become apparent as the body of work in this area grows. As with any first empirical study of a new phenomenon, this one relies on some decisions that are essentially educated guesses.

B.4 Results

We will now present our results. Section B.4.1 describes the sample, Section B.4.2 presents the main results on the aggregate level, and Section B.4.3 offers a breakdown by stimulus.

B.4.1 Descriptive Statistics

Our sample has 115 participants (31 female, 80 male, 4 non-binary or other) in the age range 19 to 40 (median 21). The median response time for the whole instrument was $12\,m15\,s$ (quartiles $10\,m29\,s$ and $14\,m16\,s$). 45% of the participants report a visual impairment and 75% of them used optical aids to compensate. The participants are balanced across groups $(25, 28, 31, 31)$ with the expected random variation. Each image pair in a group has an average of 25 ratings, with a minimum of 11 for J-Paris. Recall that participants only rated images in which they noticed differences. The anchor images have 97 or more ratings, except for U-Road, which is the uncompressed check image and does not have any difference (6 participants report having seen one).

Figure B.5 shows the proportion of participants who noticed a difference (rightmost bars) for all images along with the median response time to answer the yes/no question. For most image pairs, it took participants less than 50 seconds to notice a difference. The anchor images M-Camera and J-Santorini stand out on the slow end as they were the first stimuli of the instrument and participants had to familiarize themselves with the scenario and questions. It also took some time to check that there are no differences in U-Road. Differences were most often overlooked for J-Cardigan and for both M-Paris and J-Paris.

Tables reporting the descriptive statistics of our control questions are provided in Appendix B.3.2. We emphasize that the self-reported attitudes or behaviors in our control questions are intended to contextualize the sample. With regard to conventional image processing (retouching, montage, and digital photography), a majority report having tried them and being able to explain how they work. This contrasts with AI-supported techniques (generation, generative inpainting). Most participants have only heard of them and the practical experience is limited to having tried image generation, but not inpainting. While most participants are experienced and report a good understanding of conventional compression, they have no practical experience with neural compression. Only 32% have heard of it. A non-existing technique, "virtual image compression," was included as an attention check [61]. Most participants are honest and some think they have heard about it (Tab. B.3). We also asked how often our participants try to verify images across different platforms and contexts (Tab. B.4). A majority verifies images from unknown social network profiles at least occasionally, but tends to trust images from private contacts. Images on reputable news sites are verified slightly more often, at around the level of social network profiles of public persons or organizations. Our final control question aimed to measure the external validity by asking how realistic the scenario of a photo going viral is (Tab. B.5). 18% state that they have already experienced a situation like this, and 68% find it realistic that an image could get modified while sharing. We interpret this to mean that our scenario is not too artificial.

From the answers to the open feedback question we extracted three categories of repeating comments: 11% of participants report uncertainties about the term *misunderstanding* and would like more context, 4% are uncertain about what constitutes a *difference*, and 2% inform us that they have heard about the research project before, which may introduce bias. We use this for a robustness check.

Table B.1: Regression results supporting our hypotheses in the panel data

	Dependent variable											
Predictor	Misunderstanding (H1)				Intentional editing (H2)				Uncontrollable distortion (H3)			
Specification	(i)	(ii)	(iii)	(iv)	(i)	(ii)	(iii)	(iv)	(i)	(ii)	(iii)	(iv)
Miscompression	0.90^{***}	0.98^{***}	0.98^{***}		0.81^{***}	0.99^{***}	1.00^{***}		-0.57^{***}	-0.71^{***}	-0.73^{***}	
JPEG				-1.03^{***}				-1.11^{***}				0.79^{***}
Size 128^2			0.20				-1.34^{*}				1.47^{**}	
Size 256^2			-0.33				-1.09^{*}				1.14^{*}	
Visually impaired			0.43				0.22				-0.57	
Scene fixed eff.	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes
Subject fixed eff.	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes
Within R^2 (adj.)		0.35	0.35	0.35		0.30	0.30	0.30		0.24	0.24	0.24
Total R^2 (adj.)	0.09	0.46	0.46	0.43	0.06	0.40	0.40	0.38	0.04	0.30	0.31	0.29

Coefficients normalized to one unit of the 6-point rating scale from "certainly not" to "certainly."
$N = 1131$. Significance levels: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$.

B.4.2 Main Results

Table B.1 shows the regression results analyzing how different predictors influence users' perceived risk of how certain image differences can lead to misunderstandings (H1), are attributed to intentional editing (H2), and uncontrollable distortion (H3). The coefficients show the effect of the predictors on the user ratings measured in units of a 6-point scale.

For each dependent variable, we present four specifications: (i) a null model without fixed effects (which should not be interpreted), (ii) a baseline model with scene and subject fixed effects, (iii) an extended model including selected binary control variables, and (iv) a variant of the baseline model where the predictor is replaced with the indicator for JPEG compression. The latter is not directly related to our hypotheses and included as a contrast when the familiar JPEG artifacts are present. The estimated coefficients from specifications (ii) and (iv) are visualized as bars in the headline results (Fig. B.2 in Sect. B.1).

In the baseline model, the main effect for miscompressions on misunderstanding is positive, close to 1.0, and statistically significant at the $p < 0.001$ level. This means that, after controlling for all scene and subject-specific variation, the average participant sees the risk of a misunderstanding one step closer to "certainly" if the image is a miscompression. **This supports Hypothesis 1.** The effect does not change when control variables for the crop size and a visual impairment are included. We also included control variables derived (by approximate median splitting) from the questions on theoretical knowledge and practical experience with both conventional and AI-based image processing techniques. We observed no significant effect and refrained from reporting these specifications. We also get null results for the control variables on image verification behavior and for all dependent variables. Interpreting the R^2 measures, we observe that about 11% of the variance is explained by the presence of a miscompression, compared to 35% explained by differences between scenes and subject-specific level shifts.

We see almost the same effect for miscompressions on intentional editing. After controlling for heterogeneous subjects and scenes, the average participant attributes the difference to intentional editing one step closer to "certainly." **This supports Hypothesis 2.** The variance explained by the predictor is about the same, but the fixed effects explain slightly less for intentional editing (30%) than for misunderstandings. Unlike before, small patches with visible pixelation due to upscaling

tend to offset the attribution to intentional editing. The statistical significance level is lower due to the small number of images created from small crops.

The main effect for miscompressions on the attribution to uncontrollable distortion is negative, at around -0.7, and statistically significant at the $p < 0.001$ level. This means that after controlling for heterogeneous subjects and scenes, the average participant responds 70% of a step closer to "certainly not" when asked whether they attribute the differences to uncontrollable distortion, thus **supporting Hypothesis 3**. This compares to more than one step towards "certainly" if the image is a 128^2 crop, indicating that some participants attribute pixelation to distortion. They apparently find it hard to distinguish what distortion is already present in the reference and what is added in the test image.

Regression diagnostics do not reveal anything of concern. The stability of the coefficients across the specifications indicates the absence of excessive collinearity or suppression effects. We also explored whether the number of images a subject had rated or the position of the image in the instrument has an effect. The former has no effect and the latter has a very small learning effect, which we do not interpret because the order is confounded with the scene. As additional robustness checks, we re-estimated all regressions on two subsets of the data, first excluding the 19 subjects who mentioned one of the three concerns in the open feedback question, resulting in $N_1 = 954$ ratings, and second excluding the six subjects who saw a difference in the U-Road image, failing the attention check ($N_2 = 1064$). The signs, magnitudes, and significance levels of the main effects are unchanged.

To better interpret what a difference of one step on a 6-point scale from "certainly" to "certainly not" means for the noise level in our data, we calculate Cohen's d as a measure of effect size [19]. We obtain a "large" effect, $d = 0.86$, for misunderstanding (H1), and "moderate" effects for the other dependent variables, $d = 0.78$ (H2) and $d = 0.64$ (H3), respectively. An exploratory analysis of potential gender effects revealed that noticing differences in image pairs does not depend on gender. We observe a tendency for male participants to have slightly stronger effects in the hypothesized direction than females, especially for H1. As the interaction terms are statistically less significant ($0.01 < p < 0.1$) than our main results and the sample is imbalanced, we and refrain from reporting and interpreting quantitative results.

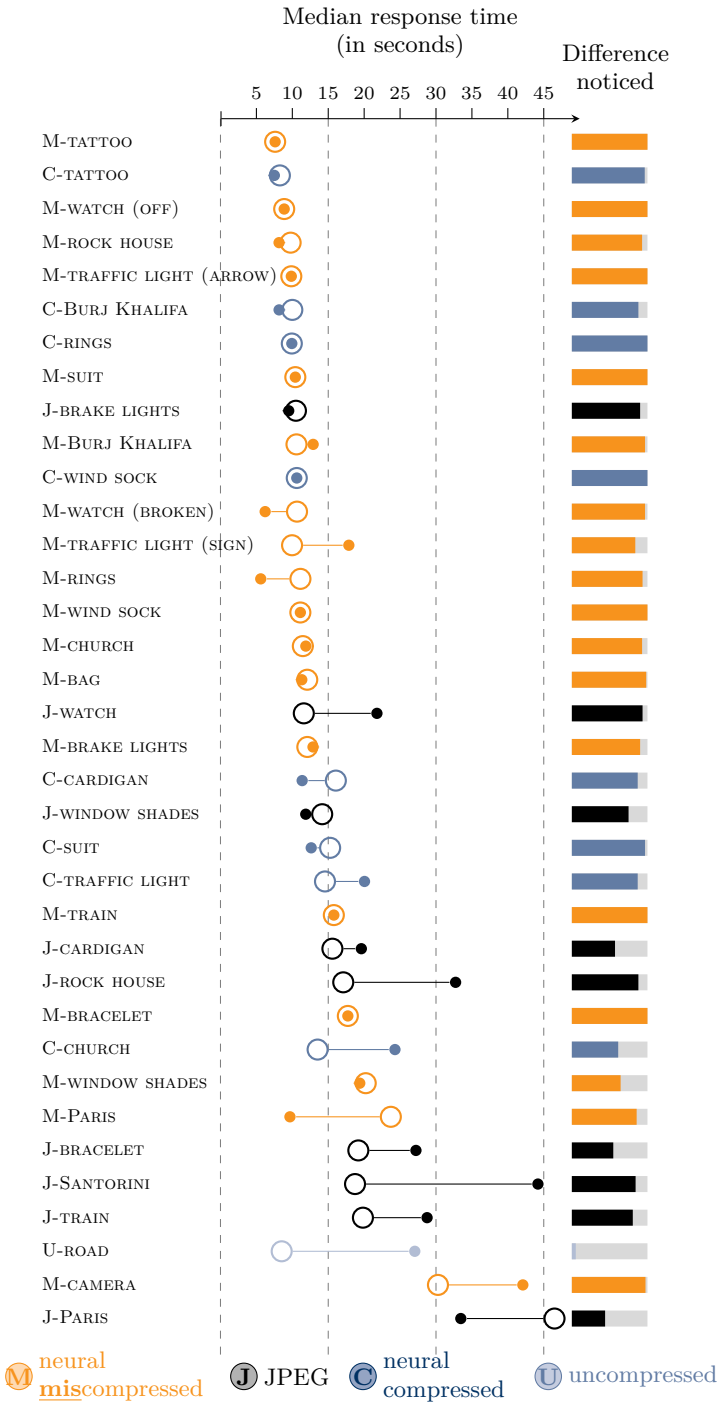


Figure B.5: Time needed to spot differences. Rings ◯ for respondents who noticed a difference, bullets • for respondents who did not. The share of respondents who noticed a difference is indicated on the right. Sorted by increasing median response time over all respondents and color coded by image type.

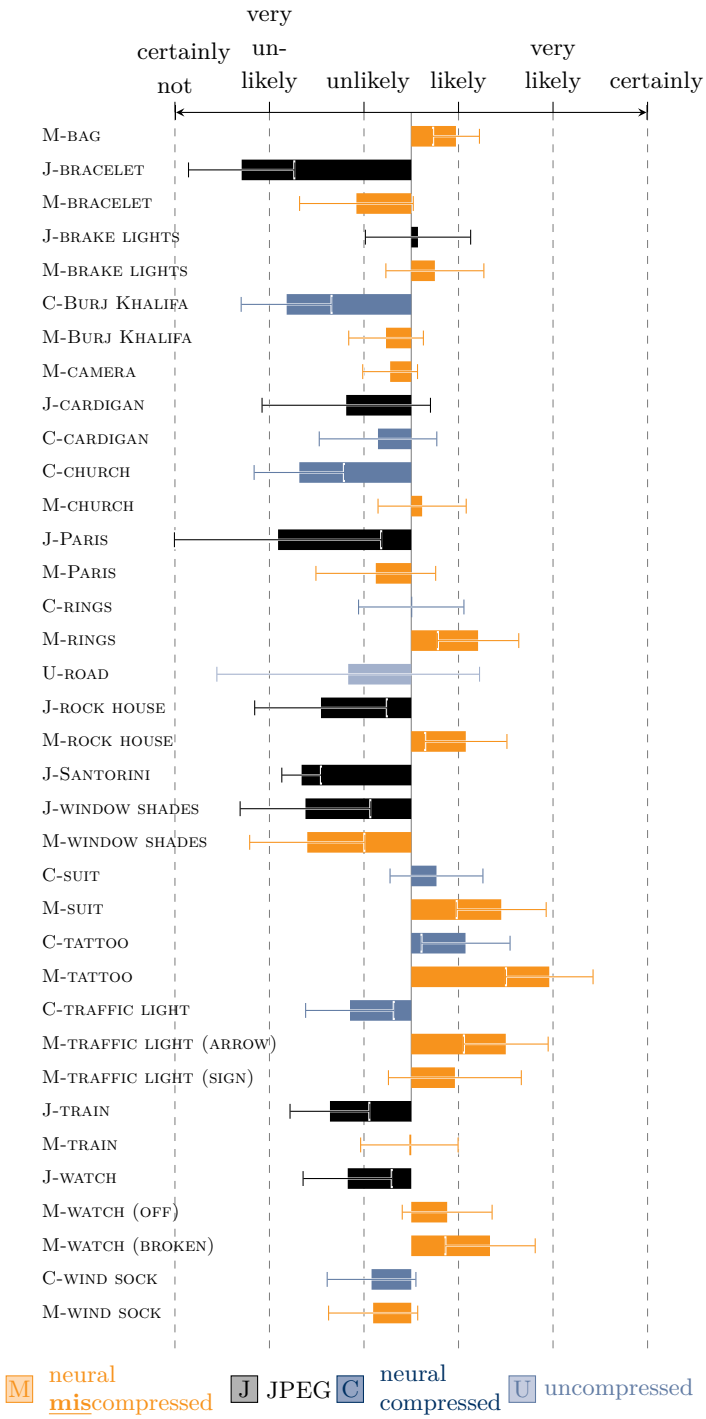


Figure B.6: Perceived risk of misunderstanding after image transmission. Means of a 6-level scale with 95 % confidence intervals, broken down by stimulus image and color coded by image type. Miscompressions (M, orange) dominate on the side of elevated risk. See text for important exceptions.

B.4.3 Breakdown by Image

While the main results give a consistent message, it is important to note that this is the average effect across a diverse set of images curated by us. Figure B.6 unpacks this by showing participants' perceived risk of misunderstanding for each image. Although most of the miscompressed images (orange bars) point towards "certainly" on the right, and the control images (all other colors) tend to point to the left, there are a few exceptions. We will discuss these in an exploratory way to learn more about users' perceptions and potential reasoning.

Paris This scene surprised us by the relatively small proportion of participants who noticed the difference: 85% for the M-PARIS and only 44% for J-PARIS. Unlike the reader in Figure B.1, the participants did not have a magnifying glass to zoom into this 512^2 image. While there is a difference in the reported susceptibility to misunderstandings in the hypothesized direction, the level is shifted to the "unlikely" half of the scale and the confidence intervals overlap, indicating that there is no statistically significant difference for this scene alone. However, this scene allows us to interpret the differences in the response time between participants who did and did not notice the differences (Fig. B.5). Observe that participants who did not notice any differences were much faster, suggesting that they overlooked the missing people. Prior work has shown that recognizing the high-frequency image content, like the people in this image, requires more effort [75].

Figure B.7: M-WIND SOCK. Participants did not detect the vanishing color of the wind sock or did not perceive it as a risk of misunderstandings. A reason could be the increased smoothness of the background.

Wind sock The differences were almost unanimously found for both M-WIND SOCK and C-WIND SOCK, and participants perceived the risk of misunderstandings as equally "unlikely" for both versions. An explanation could be that participants primarily notice the global blur, rather than the absence of the red wind sock in the miscompressed image (Fig. B.7). While we tried to select miscompressions of universally known objects, it could also be that not all participants recognized the wind sock.

Rings This scene was the only case where more participants noticed a difference in the neurally compressed control image than in the miscompression (100 vs. 94%). This can be explained by the smoothing, as shown in Figure B.8, which may resemble popular beauty filters [4]. Interestingly, while the participants attribute the differences in the control image *"very likely"* to intentionally editing (see Fig. B.11, below), they do not perceive an increased risk of misunderstandings.

Figure B.8: C-Rings: Participants attributed differences in the neurally compressed control image to intentional editing.

Tattoo This is the only scene where the neurally compressed control image is perceived as causing misunderstandings. (The confidence intervals are right of the midpoint.) Miscompression and control image still differ in the hypothesized direction, but the confidence intervals overlap. We conjecture that tattoos are perceived as sensitive features that allow one to draw inferences about a person's personality or identify individuals. This may incline participants to flag an issue (Fig. B.9). While testing this explanation would require a tailored study, we see a similar tendency in the Suit scene, where the person's face is disfigured.

Window shades This stimulus was included as an example for a miscompression of low severity. 64% of the participants notice differences in the miscompression, the lowest share of all miscompressions, and only a few believe that these could lead to misunderstandings. Participants may overlook the closed window shades because they are hard to spot (Fig. B.10). Alternatively, they may not perceive window shades as semantically relevant enough to cause misunderstandings. Moreover, the differences were attributed to uncontrollable distortion rather than intentional editing (Fig. B.11). Of course, this is context dependent [75]. If the scene was presented to experts who review construction faults, the results could be very different. This alludes to a general challenge for miscompression research. The number of domains of expertise to cover is huge.
We present the breakdown for the two remaining dependent variables in a combined scatter plot. Figure B.11 shows the mean and confidence intervals for the attribution of the differences to intentional editing on the horizontal axis and to uncontrollable distortion on the vertical axis. We

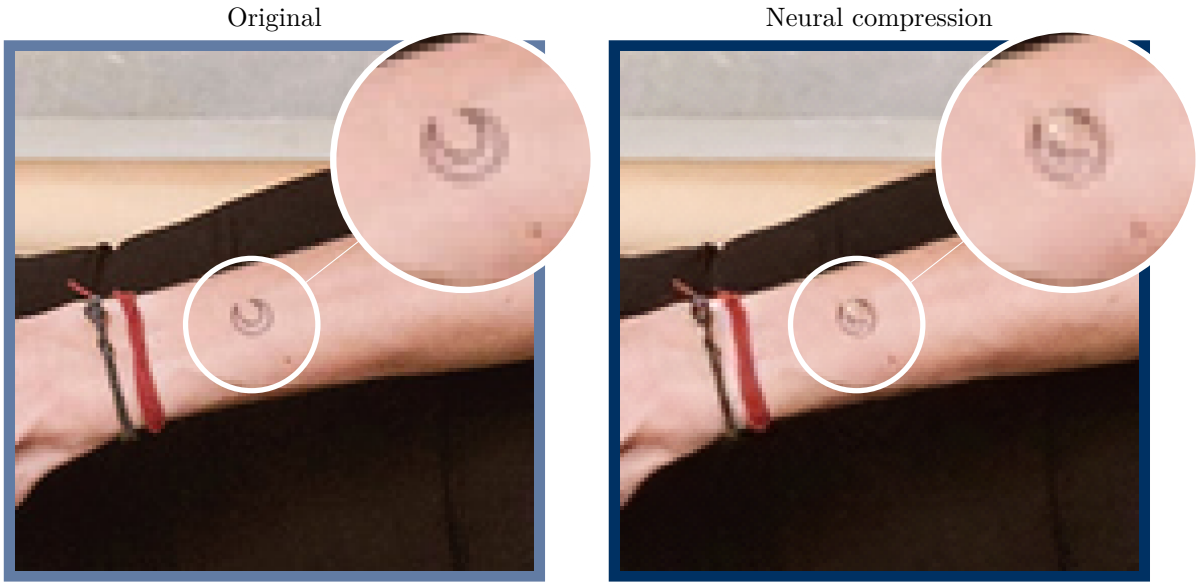


Figure B.9: C-Tattoo: Participants perceive the neurally compressed control image as a likely cause of misunderstandings.

have annotated extreme points and interesting scenes discussed above. Observe that while the data span a large range of the intentional editing scale, the responses for uncontrollable distortion are much more concentrated on the side of "certainly." This means that our participants can identify conventional JPEG compression artifacts. The negative correlation between the means on both axes shows how H2 and H3 are related across diverse scenes, and even more so between types. However, at the level of individual responses for any given scene, the two causes are not perceived as mutually exclusive (see Fig. B.14 in the Appendix).

B.5 Discussion

Image-based communication is part of everyday life and commonly relies on lossy compression. Today's codecs work invisibly for end users, fostering the expectation that mere compression does not alter the semantics of an image. Neural compression, if adopted, will disrupt this expectation for the sake of bandwidth savings. Even if codecs improve further, any information not transmitted will need to be synthesized, leaving room for hallucinations by the generative AI in the decoder. Understanding how such disruptions affect people's perception and use of images for communication is therefore critical.

This study is the first to confirm that concerns previously raised by individual researchers about miscompressions in neural compression codecs are shared by a wider set of people. This has several implications. We begin by discussing specific implications of our results for each hypothesis, before turning to broader implications for the design of future image communication interfaces.

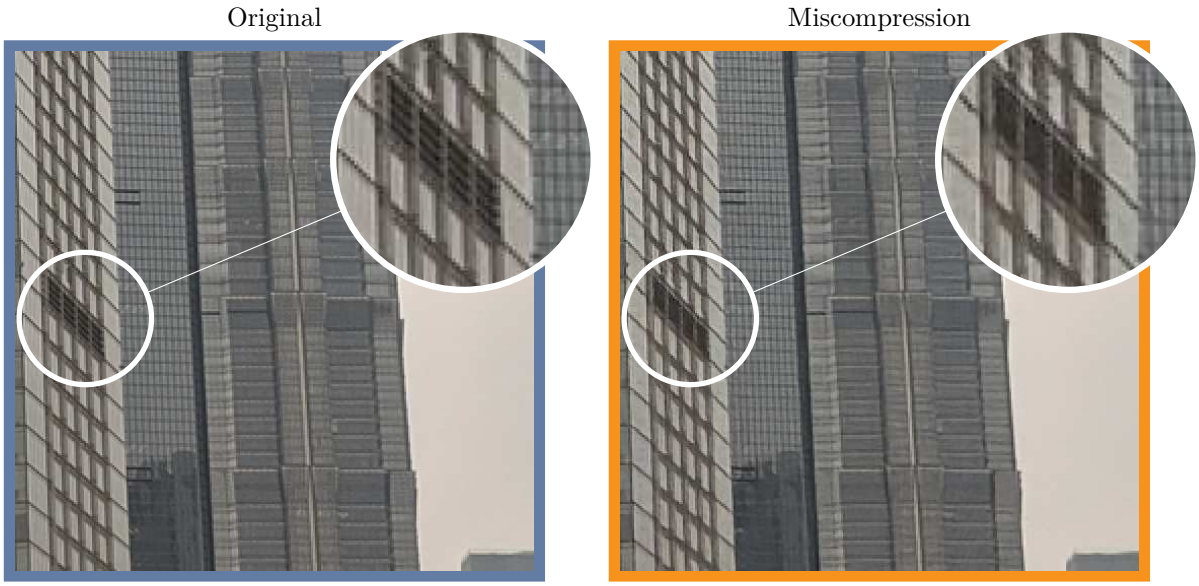


Figure B.10: M-WINDOW SHADES: Participants perceived a low risk of misunderstanding.

B.5.1 Implications of Our Results

B.5.1.1 Miscompressions Elevate the Risk of Misunderstandings (H1)

The full-reference design allowed us to probe how users perceive and evaluate semantic changes at the level of detail. Our results indicate that even subtle differences can have an effect and lead to misunderstandings, a catch-all term for a range of potential social consequences. One can easily envision the resulting risks: a changed arrow in a traffic light could cause accidents (TRAFFIC LIGHT (SIGN)), a missing wedding ring might spark false accusations (RINGS), interpreting miscompressed images in science and engineering can lead to false conclusions (WINDOW SHADES), and law enforcement could fail if biometric identifiers are misrepresented (TATTOO).
However, the current real-world impact of miscompressions is limited because the technology is not yet rolled out. This gives the community time for more human subjects research, which could inform the development of better codecs. It can include direct end-to-end tests for complete codecs, similar to our study. Moreover, data from user studies can be used to train learnable image quality metrics, which in turn can be used to train neural compression codecs.

Future research Whether a miscompression poses a risk of a misunderstanding might depend on the context. A follow-up study should test other scenarios than social media. Studies involving experts in journalism, law enforcement, and forensics can complement the picture for high-stakes applications. Concerning the instrument design, we believe that our participants had examples of misunderstandings in mind, but we did not ask for them. Future work could ask users for this information for each scene. This information would make it possible to distinguish between the likelihood and the expected severity of misunderstandings, linking these judgments to established categories in risk management [76]. On a broader level, these insights can guide a responsible deployment of neural compression across application areas. More narrowly, they can inform the calibration of future neural compression codecs to enable a risk-adjusted allocation of bits. Details which are prone to be miscompressed and cause potentially severe consequences should be preserved.

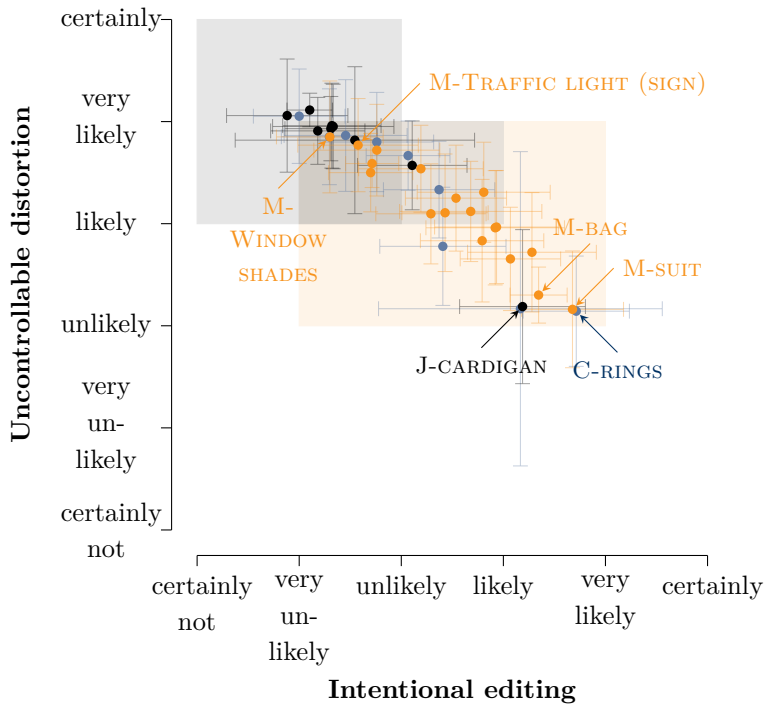


Figure B.11: Suspected causes of the image differences: scatter plot of means over the responses for intentional editing (H2) versus uncontrollable distortion (H3) aggregated for each image pair (with 95 % confidence intervals as bars). The Pearson correlation between image means is $\rho = -0.95$ ($p < 0.001$). The shaded areas visualize the empirical interquartile ranges over all responses in each type (color coding as in Fig. B.4).

B.5.1.2 Miscompressions Are Confused With Intentional Editing (H2)

Until now, the only way semantics of parts of images can change is when they are edited. This explains why people who are unaware of neural compression wrongly attribute the local differences of miscompressions to editing. This confusion can be problematic: mistaking compression artifacts for editing might hinder attempts to resolve simple misunderstandings (H1) through image comparison, especially if the parties involved (or even an independent third party) are unaware of the true cause of the differences. Even worse, it has been shown that neural compression can mislead "deep fake" detectors [16]. Accusations of malicious manipulation are quickly raised.
Moreover, previous research has shown that people are better at detecting manipulations than at identifying generated images (see Sect. B.2.2). If neural compression becomes more widely adopted, people may lose this ability as they get used to neural compression artifacts and can no longer use them as indicators of manipulation. Likewise, as the decoders of neural compression leverage generative AI, more images will appear "synthetic", diminishing users' (already limited) ability to detect generated images. Both effects further contribute to the erosion of trust in images [26].

Future research This finding paves the way for a number of follow-up research questions. Future work could contrast miscompressions with actual manipulations, both manual and AI-supported (*e.g.*, inpainting). It should also involve participants who are familiar with neural compression (*e.g.*, after passing a training phase) and experts who deal with image manipulations professionally. Our results may also prompt further research in the legal domain. The EU's AI Act [69] requires

providers of AI systems to embed machine-readable marks in generated content. Systems that "do not substantially alter the input data [...] or the semantics thereof" are exempt [69, Art. 50 (2)]. It must be clarified whether potential miscompressions would constitute an alteration to the semantics as defined by this law.

B.5.1.3 Miscompressions Are Not Recognized as Compression Artifacts (H3)

The generative networks in neural compression tend to produce visually appealing, photorealistic images. Our neurally compressed test images do not show typical compression artifacts regardless of whether they are miscompressed or not. The absence of these indicators can lead to misplaced trust in images and overconfidence in their content.
This interpretation is supported by our finding that people are indeed familiar with JPEG artifacts and interpret them as signs of compression. JPEG test images are associated with a significantly lower risk of misunderstanding and differences are less often explained with editing. In contrast, for neurally compressed control images, participants often resorted to uncontrollable distortion as a fallback explanation when no plausible cause for the differences was apparent. We conclude this from the strong negative correlation of the image means in Figure B.11.

Future research To date, little is known about how compression-induced cues interact with people's perception and trust in images, likely because researchers assumed the effect to be marginal. Our findings challenge this assumption. Follow-up studies should examine whether specific personal factors, such as experience and education draw people's attention to such cues. These studies should vary the type and strength of compression artifacts (*e.g.*, blocking, blurring, and ringing) and control for image content, device, and resolution. More than 30 years of JPEG compression could have affected how people perceive digital images.

B.5.2 Possible Interventions

Miscompressions challenge long-standing assumptions about the integrity of image communication. Given that research on neural compression is still at an early stage, the first widely adopted codecs will likely surpass those we can currently evaluate. Nevertheless, it remains highly uncertain whether it will be possible to fully avoid miscompressions with technical means. This calls for the development of strategies to mitigate potential risks. In the following, we consider approaches that allow users to recognize and respond to the risks posed by miscompressions. We distinguish passive strategies that notify users and raise awareness (Sect. B.5.2.1) and active strategies that involve user interaction and feedback (Sect. B.5.2.2).

B.5.2.1 Notifications and Awareness

Users who are aware of potential semantic changes are, in principle, better positioned to evaluate how much to trust an image. Since miscompressions cannot be detected reliably with current technology, flagging individual instances is not feasible. The risk is present in every neurally compressed image. Therefore, users should always be informed about the use of neural compression. This can be done with labels or visible watermarks in the images or image captions. Technical initiatives, such as C2PA [1, 80] and JPEG Trust [78] may serve as sources of provenance information.
Prior work provides guidance on effective labeling strategies for edited [53], AI generated [26, 28, 86], and identified misinformative content[44, 50]. While these works are excellent starting points, their findings may not translate directly to neural compression. Labels for generated content typically

signal that an image is not authentic. By contrast, a label flagging the use of neural compression only indicates that there is a possibility that an otherwise authentic image contains altered details. More research is needed to design and test effective image labeling systems adapted to the neural compression context. This should not be studied in isolation. For labeling systems to be usable in practice, they must consider the entire user experience, spanning all kinds of content sources and level of trustworthiness. The abstract risk of a miscompression must be weighed against the certain presence of completely artificial material, possibly generated with the intent to mislead. A comprehensive image labeling systems must integrate all this information and ensure that users understand it and facilitate a reaction that corresponds to the risk.

B.5.2.2 Interaction

Tailored user interfaces can support users to detect and react to instances of miscompressions. Drawing inspiration from progressive encoding [37], one approach is to deliver images at a low bitrate by default while retaining a high-quality version that can be requested on demand, for instance, when a user zooms into an image or activates a "view-as-sent" function. A user interface could also allow viewers to select whether the decoder should produce an appealing, realistic looking version of an image, or rather a version that is of lower quality but closer to the original. This is possible by transmitting a single bitstream [2]. Senders concerned about potential miscompressions could similarly benefit from a "view-as-received," option, enabling them to verify what recipients will see. Although such features cannot eliminate risk entirely, maintaining access to the original version for some period of time (by storing it on the server) could be valuable for resolving potential misunderstandings after they arise.

These techniques allow users to detect miscompressions. Once detected, users should be able to report them to the platform operator, similar to existing interfaces to report harmful content or misinformation [20]. Reactions to these reports include notifying all affected users of the specific miscompression, censoring the problematic region, replacing the image with a high-quality version, and collecting a database of miscompressions that can be used to improve future neural compression codecs. Several HCI studies are needed to determine the suitability of these measures and to identify effective ways to design and integrate such interaction mechanisms into real-world image communication platforms.

B.6 Conclusion

With billions of images exchanged through messaging apps every day, reliable image compression has become central to digital communication. This study takes a new direction of research that investigates how people perceive artifacts of conventional and future image processing technologies, and how they interpret them as cues for reliability. While the controversy involving the journalists in the introduction was fictional, our findings suggest that similar misunderstandings could arise in the real world. As compression shifts toward neural methods that can inadvertently alter the meaning of images, it is important to understand the personal and social consequences of this transition. Addressing these challenges requires not only technical advances but also sustained HCI research into how users perceive, interpret, and respond to the risks introduced by neural compression. Only in this way can we, as a society, be put in a position to decide how much bandwidth is worth saving at the expense of trust in images.

B.7 Acknowledgements

We thank all participants for taking part in our user study. We also thank Max Ninow for his help in setting up the instrument; Pascal Knierim, Florian Alt, and the members of Florian's research group for helpful advice on writing a CHI paper; and Simon Koch and Kristina Magnussen for valuable comments on the draft. This work received funding by the state of Tyrol (F.50541/6-2024). Computational results were achieved using the LEO HPC infrastructure at the University of Innsbruck.

References

[1] Rafal Ablamowicz and Bertfried Fauser. Content credentials: C2PA technical specification, 2025.

[2] Eirikur Agustsson, David Minnen, George Toderici, and Fabian Mentzer. Multi–realism image compression with a conditional generator. In *Conference on Computer Vision and Pattern Recognition*, pages 22324–22333. IEEE/CVF, 2023.

[3] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super–resolution: dataset and study. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135. IEEE, 2017.

[4] Markus Appel, Fabian Hutmacher, Theresa Politt, and Jan-Philipp Stein. Swipe right? Using beauty filters in male Tinder profiles reduces women's evaluations of trustworthiness but increases physical attractiveness and dating intention. *Computers in Human Behavior*, 148:107871, 2023.

[5] Joao Ascenso, Elena Alshina, and Touradj Ebrahimi. The JPEG AI standard: Providing efficient human and machine visual data consumption. *IEEE Multimedia*, 30(1):100–111, 2023.

[6] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017.

[7] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

[8] Martin Beneš, Nora Hofer, and Rainer Böhme. Know your library: How the libjpeg version influences compression and decompression results. In *Workshop on Information Hiding and Multimedia Security*, pages 19–25. ACM, 2022.

[9] Sandra Bergmann, Denise Moussa, Fabian Brand, André Kaup, and Christian Riess. Forensic analysis of AI-compression traces in spatial and frequency domain. *Pattern Recognition Letters*, pages 41–47, 2024.

[10] Sandra Bergmann, Denise Moussa, and Christian Riess. Trustworthy compression? impact of ai-based codecs on biometrics for law enforcement. *arXiv preprint arXiv:2408.10823*, 2024.

[11] Alexandre Berthet and Jean-Luc Dugelay. AI-based compression: A new unintended counter attack on JPEG-related image forensic detectors? In *International Conference on Image Processing*, pages 3426–3430. IEEE, 2022.

[12] Alexandre Berthet, Chiara Galdi, and Jean-Luc Dugelay. On the impact of AI-based compression on deep learning-based source social network identification. In *International Workshop on Multimedia Signal Processing*, pages 1–6. IEEE, 2023.

[13] George Bishop. Experiments with the middle response alternative in survey questions. *Public Opinion Quarterly*, 51(2):220–232, 1987.

[14] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate–distortion–perception tradeoff. In *International Conference on Machine Learning*, pages 675–685. PMLR, 2019.

[15] Rainer Böhme and Matthias Kirchner. Counter-forensics: Attacking image forensics. In Husrev T. Sencar and Nasir D. Memon, editors, *Digital Image Forensics: There is More to a Picture Than Meets the Eye*, pages 327–366. Springer, 2012.

[16] Edoardo Daniele Cannas, Sara Mandelli, Natasa Popovic, Ayman Alkhateeb, Alessandro Gnutti, Paolo Bestagini, and Stefano Tubaro. Is JPEG AI going to change image forensics? *arXiv preprint arXiv:2412.03261*, 2024.

[17] João Phillipe Cardenuto, Joshua Krinsky, Lucas Nogueira, Aparna Bharati, and Daniel Moreira. Implications of neural compression to scientific images. In *Workshop on Information Hiding and Multimedia Security*, pages 80–85. ACM, 2025.

[18] Tong Chen and Zhan Ma. Toward robust neural image compression: Adversarial attack and model finetuning. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(12):7842–7856, 2023.

[19] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Routledge, 2nd edition, 2013.

[20] Kate Crawford and Tarleton Gillespie. What is a flag for? social media reporting tools and the vocabulary of complaint. *New Media & Society*, 18(3):410–428, 2016.

[21] Tiago José De Carvalho, Christian Riess, Elli Angelopoulou, Helio Pedrini, and Anderson de Rezende Rocha. Exposing digital image forgeries by illumination color classification. *Transactions on Information Forensics and Security*, 8(7):1182–1194, 2013.

[22] Benedikt Dornauer and Michael Felderer. Web image formats: Assessment of their real-world-usage and performance across popular web browsers. In *International Conference on Product-Focused Software Process Improvement*, pages 132–147. Springer, 2023.

[23] Hany Farid and Mary J Bravo. Photorealistic rendering: How realistic is it? *Journal of Vision*, 7(9):766–766, 2007.

[24] Hany Farid and Mary J Bravo. Image forensic analyses that elude the human visual system. In *Media Forensics and Security II*, pages 52–61. SPIE, 2010.

[25] Hany Farid and Mary J Bravo. Perceptual discrimination of computer generated and photographic faces. *Digital Investigation*, 8(3-4):226–235, 2012.

[26] KJ Kevin Feng, Nick Ritchie, Pia Blumenthal, Andy Parsons, and Amy X Zhang. Examining the impact of provenance-enabled media on trust and accuracy perceptions. *Human-Computer Interaction*, 7(CSCW2):1–42, 2023.

[27] Joel Frank, Franziska Herbert, Jonas Ricker, Lea Schönherr, Thorsten Eisenhofer, Asja Fischer, Markus Dürmuth, and Thorsten Holz. A representative study on human detection of artificially generated media across countries. In *Symposium on Security and Privacy*, pages 55–73. IEEE, 2024.

[28] Dilrukshi Gamage, Dilki Sewwandi, Min Zhang, and Arosha K Bandara. Labeling synthetic content: User perceptions of label designs for AI-generated content on social media. In *CHI Conference on Human Factors in Computing Systems*, pages 1–29, 2025.

[29] Gina Gayle. *The Perceived Credibility of Professional Photojournalism Compared to User-Generated Content among American News Media Audiences*. PhD thesis, Syracuse University, 2020.

[30] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[31] David Gottlieb and Chi-Wang Shu. On the gibbs phenomenon and its resolution. *Society for Industrial and Applied Mathematics Review*, 39(4):644–668, 1997.

[32] Jennifer D Greer and Joseph D Gosen. How much is too much? assessing levels of digital alteration of factors in public perception of news media credibility. *Visual Communication Quarterly*, 9(3):4–13, 2002.

[33] Michael J Hautus, Neil A Macmillan, and C Douglas Creelman. *Detection Theory: A User's Guide*. Routledge, 2021.

[34] Dailan He, Ziming Yang, Hongjiu Yu, Tongda Xu, Jixiang Luo, Yuan Chen, Chenjian Gao, Xinjie Shi, Hongwei Qin, and Yan Wang. PO-ELIC: perception-oriented efficient learned image coding. In *Conference on Computer Vision and Pattern Recognition*, pages 1764–1769. IEEE/CVF, 2022.

[35] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time–scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

[36] Nora Hofer. Increasing trust in image analysis by detecting trellis quantization in JPEG images. In *International Conference on Image Processing*, pages 3834–3840. IEEE, 2024.

[37] Nora Hofer and Rainer Böhme. Progressive JPEGs in the wild: Implications for information hiding and forensics. In *Workshop on Information Hiding and Multimedia Security*, pages 47–58. ACM, 2023.

[38] Nora Hofer and Rainer Böhme. A taxonomy of miscompressions: Preparing image forensics for neural compression. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2024.

[39] Nora Hofer and Rainer Böhme. Challenging cases of neural image compression: A dataset of visually compelling yet semantically incorrect reconstructions. In *International Conference on Multimedia*, pages 13318–13324. ACM, 2025.

[40] Olivia Holmes, Martin S Banks, and Hany Farid. Assessing and improving the identification of computer-generated portraits. *ACM Transactions on Applied Perception*, 13(2):1–12, 2016.

[41] International Telecommunication Union. Methodology for the subjective assessment of the quality of television pictures. *Recommendation ITU-R BT.500-14*, 2012. Available at: `https://www.itu.int/rec/R-REC-BT.500-14-201910-I/en`.

[42] International Telecommunication Union. Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment. *Recommendation ITU-T P.913*, 2016. Available at: `https://www.itu.int/rec/T-REC-P.913/en`.

[43] International Telecommunication Union. Subjective video quality assessment methods for multimedia applications. *Recommendation ITU-T P.910*, 2023. Available at: `https://www.itu.int/rec/T-REC-P.910/en`.

[44] Farnaz Jahanbakhsh and David R Karger. A browser extension for in-place signaling and assessment of misinformation. In *CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2024.

[45] Ehsaneddin Jalilian, Heinz Hofbauer, and Andreas Uhl. Iris image compression using deep convolutional neural networks. *Sensors*, 22(7):2698, 2022.

[46] Eunchae Jang, Hui Min Lee, Sangwook Lee, Yongnam Jung, and S Shyam Sundar. Too good to be false: How photorealism promotes susceptibility to misinformation. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2025.

[47] Se-Hoon Jeong, Hyunyi Cho, and Yoori Hwang. Media literacy interventions: A meta-analytic review. *Journal of Communication*, 62(3):454–472, 2012.

[48] Panqi Jia, A Burakhan Koyuncu, Jue Mao, Ze Cui, Yi Ma, Tiansheng Guo, Timofey Solovyev, Alexander Karabutov, Yin Zhao, Jing Wang, Elena Alshina, and Andre Kaup. Bit rate matching algorithm optimization in JPEG-AI verification model. In *Picture Coding Symposium*, pages 1–5. IEEE, 2024.

[49] S Mo Jones-Jang, Tara Mortensen, and Jingjing Liu. Does media literacy help identification of fake news? Information literacy helps, but other literacies don't. *American Behavioral Scientist*, 65(2):371–388, 2021.

[50] Ben Kaiser, Jerry Wei, Eli Lucherini, Kevin Lee, J Nathan Matias, and Jonathan Mayer. Adapting security warnings to counter online disinformation. In *USENIX Security Symposium*, pages 1163–1180, 2021.

[51] Negar Kamali, Karyn Nakamura, Aakriti Kumar, Angelos Chatzimparmpas, Jessica Hullman, and Matthew Groh. Characterizing photorealism and artifacts in diffusion model-generated images. In *CHI Conference on Human Factors in Computing Systems*, pages 1–26, 2025.

[52] Mona Kasra, Cuihua Shen, and James F O'Brien. Seeing is believing: How people fail to identify fake images on the web. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2018.

[53] Eric Kee and Hany Farid. A perceptual metric for photo retouching. *National Academy of Sciences*, 108(50):19907–19912, 2011.

[54] James E Kelly and Diona Nace. Digital imaging & believing photos. *Visual Communication Quarterly*, 1(1):4–18, 1994.

[55] Federica Lago, Cecilia Pasquini, Rainer Böhme, Hélène Dumont, Valérie Goffaux, and Giulia Boato. More real than real: A study on human visual perception of synthetic faces. *Signal Processing Magazine*, 39(1):109–116, 2021.

[56] Zeyu Lu, Di Huang, Lei Bai, Jingjing Qu, Chengyue Wu, Xihui Liu, and Wanli Ouyang. Seeing is not always believing: Benchmarking human and model perception of AI-generated images. *Advances in Neural Information Processing Systems*, 36, 2023.

[57] Jordan Madden, Lhamo Dorje, and Xiaohua Li. Bitstream collisions in neural image compression via adversarial perturbations. *arXiv preprint arXiv:2503.19817*, 2025.

[58] Brandon Mader, Martin S Banks, and Hany Farid. Identifying computer-generated portraits: The importance of training and incentives. *Perception*, 46(9):1062–1076, 2017.

[59] Daniele Mari, Saverio Cavasin, Simone Milani, and Mauro Conti. Effectiveness of learning-based image codecs on fingerprint storage. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2024.

[60] Pina Marziliano, Frederic Dufaux, Stefan Winkler, and Touradj Ebrahimi. A no-reference perceptual blur metric. In *International Conference on Image Processing*, volume 3, pages III–III. IEEE, 2002.

[61] Adam W Meade and S Bartholomew Craig. Identifying careless responses in survey data. *Psychological Methods*, 17(3):437, 2012.

[62] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, pages 11913–11924, 2020.

[63] Jaron Mink, Miranda Wei, Collins W Munyendo, Kurt Hugenberg, Tadayoshi Kohno, Elissa M Redmiles, and Gang Wang. It's trying too hard to look real: Deepfake moderation mistakes and identity-based bias. In *CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2024.

[64] Tara Marie Mortensen, Brian P McDermott, and Khadija Ejaz. Measuring photo credibility in journalistic contexts: Scale development and application to staff and stock photography. *Journalism Practice*, 17(6):1158–1177, 2023.

[65] Sophie J Nightingale and Hany Farid. AI-synthesized faces are indistinguishable from real faces and more trustworthy. *National Academy of Sciences*, 119(8):e2120481119, 2022.

[66] Yuzhen Niu, Feng Liu, Xueqing Li, and Michael Gleicher. The complexity of perception of image distortion: an initial study. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 3235–3240. 2010.

[67] Yuri Ostrovsky, Patrick Cavanagh, and Pawan Sinha. Perceiving illumination inconsistencies in scenes. *Perception*, 34(11):1301–1314, 2005.

[68] Basak Oztan, Amal Malik, Zhigang Fan, and Reiner Eschbach. Removal of artifacts from JPEG compressed document images. In *Color Imaging XII: Processing, Hardcopy, and Applications*, volume 6493, pages 60–68. SPIE, 2007.

[69] European Parliament and the Council. Regulation (EU) 2024/1689 of 13 june 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), 2024.

[70] Pat Pataranutaporn, Chayapatr Archiwaranguprok, Samantha WT Chan, Elizabeth Loftus, and Pattie Maes. Synthetic human memories: AI-edited images and videos can implant false memories and distort recollection. In *CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2025.

[71] Yash Patel, Srikar Appalaraju, and R Manmatha. Human perceptual evaluations for image compression. *arXiv preprint arXiv:1908.04187*, 2019.

[72] Tian Qiu, Arjun Nichani, Rasta Tadayontahmasebi, and Haewon Jeong. Gone with the bits: Revealing racial bias in low-rate neural compression for facial images. In *Conference on Fairness, Accountability, and Transparency*, pages 1862–1889. ACM, 2025.

[73] Lucas Relic, Roberto Azevedo, Markus Gross, and Christopher Schroers. Lossy image compression with foundation diffusion models. In *European Conference on Computer Vision*, pages 303–319. Springer, 2024.

[74] JPEG (ISO/IEC SC29/WG1). JPEG AI reference software, 2024. `https://gitlab.com/wg1/jpeg-ai/jpeg-ai-reference-software`, version 7.0.

[75] Victor Schetinger, Manuel M Oliveira, Roberto da Silva, and Tiago J Carvalho. Humans are easily fooled by digital images. *Computers & Graphics*, 68:142–151, 2017.

[76] Michael Warren Skirpan, Tom Yeh, and Casey Fiesler. What's at stake: Characterizing risk perceptions of emerging technologies. In *CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

[77] Zhihao Sun, Haipeng Fang, Juan Cao, Xinying Zhao, and Danding Wang. Rethinking image editing detection in the era of generative AI revolution. In *International Conference on Multimedia*, pages 3538–3547. ACM, 2024.

[78] Frederik Temmermans, Sabrina Caldwell, Deepayan Bhowmik, and Touradj Ebrahimi. JPEG Trust: an international standard facilitating the assessment of trustworthiness of digital media assets. In *Applications of Digital Image Processing XLVII*, volume 13137, pages 99–104. SPIE, 2024.

[79] George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Balle, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer. Workshop and challenge on learned image compression (CLIC2020). In *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2020.

[80] Christoph Trattner, Svenja Lys Forstner, Alain D Starke, and Erik Knudsen. C2PA provenance labels increase trust in news platforms across western countries. 2024.

[81] Sophie Triantaphillidou, Elizabeth Allen, and R Jacobson. Image quality comparison between JPEG and JPEG2000. II. scene dependency, scene analysis, and classification. *Journal of Imaging Science and Technology*, 51(3):259–270, 2007.

[82] Daria Tsereh, Mark Mirgaleev, Ivan Molodetskikh, Roman Kazantsev, and Dmitriy Vatolin. JPEG AI image compression visual artifacts: detection methods and dataset. *arXiv preprint arXiv:2411.06810*, 2024.

[83] Dhanraj Vishwanath, Ahna R Girshick, and Martin S Banks. Why pictures look right when viewed from the wrong place. *Nature Neuroscience*, 8(10):1401–1410, 2005.

[84] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[85] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Conference on Signals, Systems & Computers*, pages 1398–1402. IEEE, 2003.

[86] Chloe Wittenberg, Ziv Epstein, Adam J Berinsky, and David G Rand. Labeling AI-generated content: promises, perils, and future directions. *An MIT Exploration of Generative AI*, 2024.

[87] Leslie Wöhler, Martin Zembaty, Susana Castillo, and Marcus Magnor. Towards understanding perceptual differences between genuine and face-swapped videos. In *CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2021.

[88] Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, pages 64971–64995, 2024.

[89] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. FSIM: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011.

[90] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern Recognition*, pages 586–595. IEEE, 2018.

[91] Renjie Zou, Chunfeng Song, and Zhaoxiang Zhang. The devil is in the details: Window-based attention for image compression. In *Conference on Computer Vision and Pattern Recognition*, pages 17492–17501. IEEE/CVF, 2022.

B.A Background

B.1.1 Principles of Neural Image Compression

The first step of the compression pipeline is the analysis transform, a neural network that encodes the input image into a latent representation. This representation is quantized by rounding and then coded into a bit stream using arithmetic encoding. The most popular construction is called "hyperprior" [7] because it parametrizes the arithmetic encoder by another autoencoder, which predicts the distribution of the latent representation produced by the transform network. This autoencoder's latent space is transmitted for arithmetic decoding. Finally, a synthesis transform network reconstructs the image from the decoded representation. All steps are trained end to end by minimizing a rate–distortion function. To generate the stimulus images in our user study, we used the variational autoencoder used in the original **Hyperprior** proposal by Ballé et al. [7] and three follow-up works. The **HiFiC** model by Mentzer et al. [62] differs in the last step. It uses the decoded latent representation to condition a Generative Adversarial Network (GAN) which reconstructs the image. By contrast, Yang et al.'s **CDC** codec [88] conditions a diffusion model on the decoded latent representation for the reconstruction. Also the **JPEG AI** [5] codec uses

the hyperprior construction and feeds the decoded latent representation into a synthesis transform network to reconstruct the image. Unlike other codecs, JPEG AI converts the image into the YUV color space before transformation.

B.1.2 Acknowledgement of Semantic Changes

In their limitation section, Relic et al. [73, p. 316] mention the challenge of "*misgeneration of content*". They remark: "*Specifically at very low bitrates, identities, text, or lower-level content can vary from the original image, and thus may raise ethical concerns in specific scenarios.*" Also, Mentzer et al. [62, p. 10] mention failure cases, such as small text or faces, and emphasize that "in theory" their generator can "*[. . .] produce images that are very different from the input*" and that their method is "*not suitable for sensitive image contents, such as, e.g., storing medical images, or important documents.*" Most notably, Agustsson et al. note that "*Since the realism constraint might produce reconstructions that are far away from the input, these systems might be looked at with suspicion because it is not clear which details are in the original and which were added by the architecture*" [2, p. 22325]. None of the above studies measure the perception of the semantic differences.

B.B Method

B.2.1 Participant Briefing

[Translated from German and *[anonymized]*]

Welcome to the study **Perception of disturbances in digital images** conducted by *[institute]*!

During the transmission of images on the internet, visible and invisible "distortions" can occur. Researchers worldwide are working on new standards to improve the quality of transmitted images. In this study, you will compare original images with transmitted ones and evaluate any differences that may appear.

The participation is voluntary, and you have the right to withdraw from the study at any time. Your responses and the answering time per question will be recorded. All data will be fully anonymized. The study will take approximately 15 minutes to complete. It is part of the *[research project and funding organization]*. The participation in the study involves no risks.

If you have questions during the study, please contact the pro seminar teacher. For questions after completing the study, you can reach out to *[name and contact information of the project leader]*. At the end of the study, you will have the opportunity to provide your email address to receive updates on the study's findings.

Your participation helps us improve future image formats. Thank you for your contribution!

[Declaration of Consent]

B.2.2 Questionnaire

[Translated from German. *Answer options are listed in italics.*]

Part 1: Introduction

I1: [Same as printed Briefing]

Part 2: Demographics

D1: What year were you born? *Answer must be between 1925 and 2006.*

D2: Which gender do you identify with? *female, male, non-binary, would not like to specify*

D3: Eyesight *yes, no*

1. Is your vision impaired, *e.g.*, due to long/short-sightedness or color blindness?

2. Do you use optical aids to compensate for this while completing this study, *e.g.*, glasses or contact lenses?

Part 3: Image Comparison

Suppose you took a picture some time ago and uploaded it to a social media platform. In the meantime, the image has spread across the internet and a second person discovers it in their feed on another platform.
[Stimulus 1] image taken by you
[Stimulus 2] image discovered by the other person

S1: The two images are not identical. Can you see at least one difference? *yes (condition for S2 and S3), no*

S2: What are the effects of the differences? *certainly, very likely, likely, unlikely, very unlikely, certainly not*

1. Could the differences lead to **misunderstandings** between you and the other person?

S3: What are the causes of the differences? *certainly, very likely, likely, unlikely, very unlikely, certainly not*

1. Would you attribute the differences to **intentional editing**, *e.g.*, retouching, filters, or manipulation?

2. Would you attribute the differences to **uncontrolled distortions**, *e.g.*, transmission errors or image compression?

A1: Action page
[Appeared once after the first image comparison block.]
Thank you – We will now repeat this for different images.
Please don't get distracted and evaluate each picture carefully.

Part 4: Control Variables

C1: Please indicate your **theoretical knowledge** of the following technologies *never heard of it, have heard of it, can explain how it works, profound knowledge*

1. Image retouching, *e.g.*, color corrections, removal of "blemishes", smoothing, or sharpening of textures
2. Image montage, *e.g.*, adding, modifying, or removing objects with the cloning tool
3. AI-supported image generation, *e.g.*, DALL · E, MidJourney, or Stable Diffusion
4. Generative inpainting, *e.g.*, with DeepFill, Adobe Firefly Generative Fill, or DALL · E
5. Digital photography, *e.g.*, with a smartphone or digital camera
6. Conventional lossy image compression, *e.g.*, JPEG or WEBP
7. Neural image compression, *e.g.*, with algorithms like JPEG AI or HiFiC
8. Virtual image compression, *e.g.*, with codecs like VBC Optimizer or SpectraZip

C2: Please indicate your **practical experience** with the following technologies. *no practical experience, have tried it, use occasionally, use regularly*
[*cf.* enumeration from C1]

C3: Please indicate how often you verify the authenticity of images from various sources, *e.g.*, by zooming in on the image. I verify in image ... *... almost always., ... often., ... occasionally., ... almost never., not applicable*

1. Social network profile of a public person or organization I know
2. Social network profile of a public person or organization I **don't** know
3. Social network profile of a private person I know
4. Social network profile of a private person I **don't** know
5. Private direct message (e.g., SnapChat, Instagram, WhatsApp)
6. Reputable online news site

C4: What is your experience with the distribution of images on the internet? *happens to me regularly, has happened to me, could happen to me, could rather not happen to me, don't know*

1. How realistic is the scenario with the photo that gets widely distributed online?
2. And how realistic is it that the photo gets modified in this process?

F1: Please take a moment to share your feedback on the study, *e.g.*, the clarity of the questions, the selection and presentation of the images, any difficulties in answering, *etc.*

[Open question field]

End page

Thank you for your participation!
If you are interested in the results of the study, please provide us with your email address. The address will be stored separately from your responses. [Link to survey]

B.2.3 Stimuli Compression

The source images were taken from two widely-used image compression benchmark datasets [3, 79]. For the compression of our test images we selected four state-of-the-art neural image compression codecs that cover the range of technologies applied in the neural compression literature. For the compression with the Hyperprior [7] we used the hierarchical mode optimized for MSE at compression intensity 3 out of 8. For the compression with the GAN based HiFiC model [62] we used the intensities Hi or Lo. We used TensorFlow Compression library[4] (TFC) for the compression with these two models and pretrained weights. For the compression with JPEG AI [5], we used the verification model of its Reference Software [74] at version 7.0, commit 50EC1478.
During encoding we use high operation point (HOP)[48] and disable all tools. We include images for the target bits per pixel (BPP) values 0.25 and 0.75. Lastly, we use the code repository of the authors and their pretrained weights to compress images with the diffusion based CDC model [88] using x-parameterization for LPIPS weight 0.9 and Lagrangian multiplier to control the compression quality of 2048 and 0512. All codecs were executed on a shared GPU cluster using a 64-core Nvidia A100 GPU.
To compress the JPEG control images we used the Python Imaging Library (PIL) version 11.1.0 with `libjpeg-turbo`, version 3.0 [8] at the default quality factor 75.

B.2.4 Stimuli Overview

[4] `https://github.com/tensorflow/compression`

Table B.2: Compression specification and verbal description of semantic changes of stimulus images

Test image	Crop Size	Compression codec	BPP[1)]	Position in instrument	Source dataset	Semantic change
Bag–M	256^2	CDC-2048x09	0.30	11	CLIC	The color of the bag changes from purple to blue.
Bracelet–M	256^2	CDC-2048x09	0.29	3	CLIC	The texture changes from a beaded to a knotted look.
Bracelet–J	256^2	JPEG-75	1.03	5	CLIC	-
Brake lights–M	128^2	HiFiC-Hi	0.47	9	DIV2K	The van's brake lights are turned off.
Brake lights–J	128^2	JPEG-75	1.27	9	DIV2K	-
Burj Khalifa–M	256^2	Hyper-MSE 3	0.19	3	DIV2K	Color appears on the façade that could resemble paint.
Burj Khalifa–C	256^2	JPEG AI-0.25	0.22	10	DIV2K	-
Camera–M	256^2	HiFiC-Lo	0.09	1	DIV2K	The engraved number 8 changes into the number 6.
Cardigan–C	512^2	CDC-2048x09	0.28	4	CLIC	-
Cardigan–J	512^2	JPEG-75	1.35	9	CLIC	-
Church–M	256^2	HiFiC-Lo	0.08	6	CLIC	The steeple's cross changes into a star.
Church–C	256^2	CDC-0512x09	0.47	10	CLIC	-
Paris–M	512^2	HiFiC-Lo	0.17	8	CLIC	The people in the grass disappear, and the crowd on the stairs turns into a black blur.
Paris–J	512^2	JPEG-75	1.25	4	CLIC	-
Rings–M	256^2	HiFiC-Lo	0.14	10	CLIC	The thumb ring and the necklace pendant disappear.
Rings–C	256^2	Hyper-MSE 3	0.20	4	CLIC	-
Road–U	512^2	-	-	12	CLIC	-
Rock house–M	128^2	HiFiC-Lo	0.14	10	DIV2K	The house built into the rock is unrecognizable because its door, windows, and roof disappear.
Rock house–J	128^2	JPEG-75	1.06	5	DIV2K	-
Santorini–J	256^2	JPEG-75	1.24	2	DIV2K	-
Suit–M	256^2	HiFiC-Lo	0.17	7	DIV2K	The man's hand and facial features disappear.
Suit–C	256^2	JPEG AI-0.25	0.26	3	DIV2K	-
Tattoo–M	256^2	CDC-2048x09	0.29	9	CLIC	The tattoo turns from a crescent moon into a full moon.
Tattoo–C	256^2	CDC-0512x09	0.69	7	CLIC	
Traffic l.–M (arrow)	128^2	CDC-2048x09	0.22	3	CLIC	The arrow-shaped traffic light has turned into a regular, round traffic light.
Traffic l.–M (sign)	128^2	HiFiC-Hi	0.36	5	CLIC	The "No Cars" sign has turned into a "No Cameras" sign.
Traffic l.–C	128^2	JPEG AI-0.25	0.26	4	CLIC	
Train–M	256^2	HiFiC-Lo	0.17	6	DIV2K	The head of a person leaning out the window disappears.
Train–J	256^2	JPEG-75	1.18	6	DIV2K	
Watch (broken)–M	256^2	CDC-2048x09	0.19	8	CLIC	The smartwatch looks broken.
Watch (off)–M	256^2	HiFiC-Lo	0.09	7	CLIC	The display of the smartwatch is turned off.
Watch–J	256^2	JPEG-75	0.72	8	CLIC	-
Wind sock–M	256^2	JPEG AI-0.25	0.22	6	CLIC	The red wind sock disappears.
Wind sock–C	256^2	HiFiC-Lo	0.08	8	CLIC	-
Window shades–M	128^2	HiFiC-Hi	0.36	7	CLIC	The open window shades look closed.
Window shades–J	128^2	JPEG-75	0.61	5	CLIC	-

[1]) measured for the full image.

B.2.5 Images

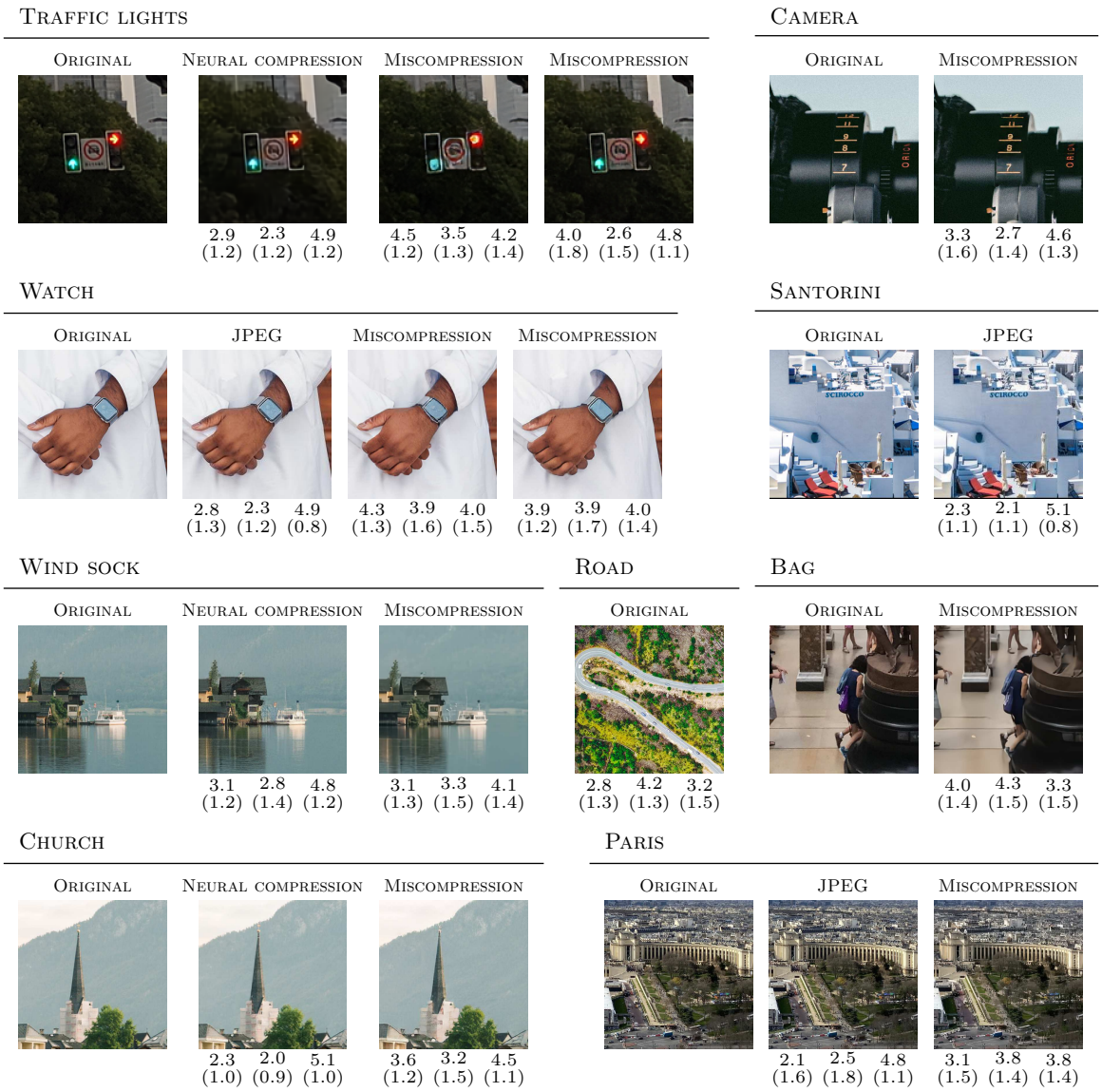


Figure B.12: Stimulus material with per-image descriptive statistics. Camera, Bag, Santorini and Road were shown to all groups. Images are best viewed on screen and magnified. See Figure B.13 on the next page for the legend.

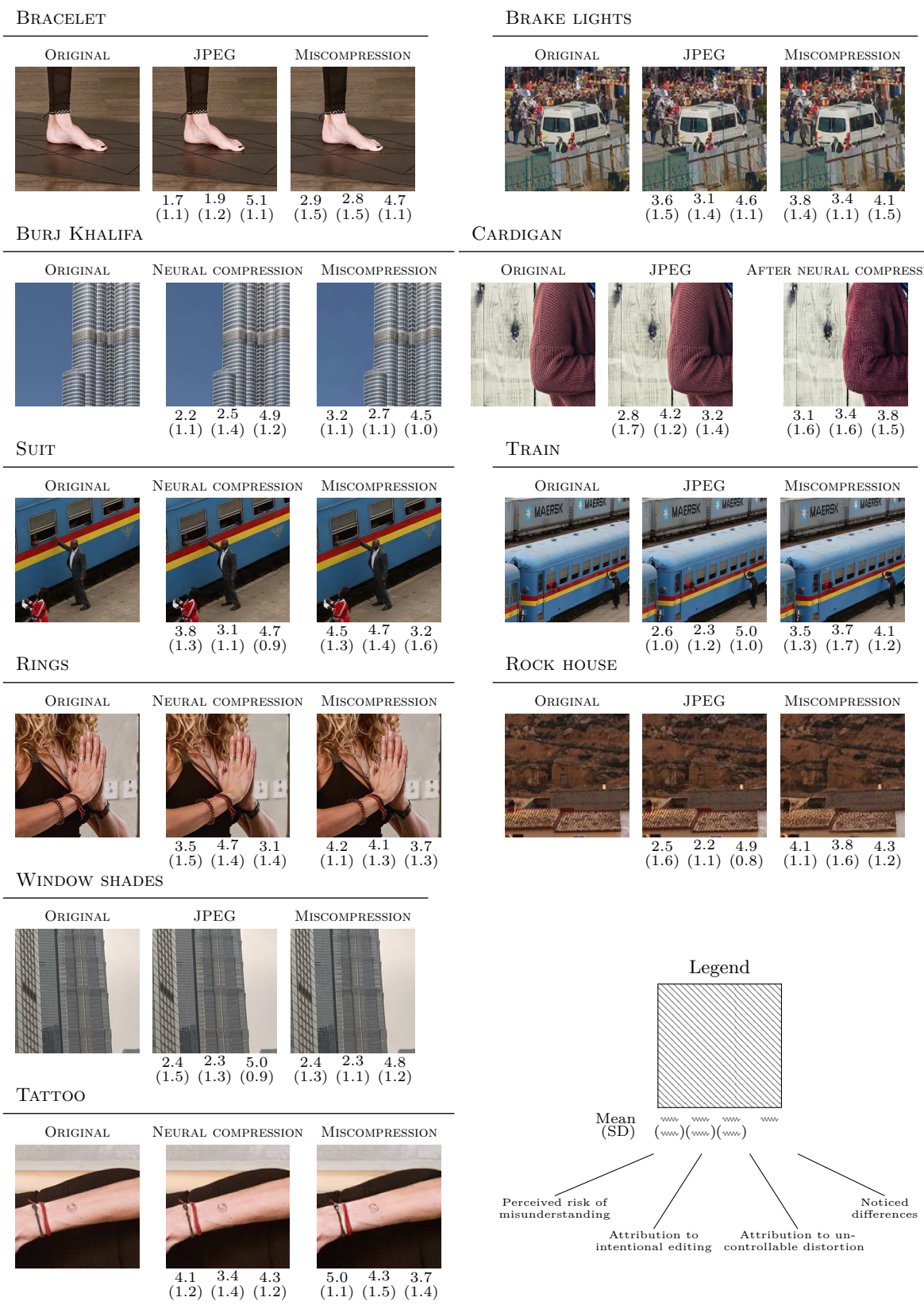


Figure B.13: Stimulus material with per-image descriptive statistics (continued).

B.C Results

B.3.1 Supplemental Figure

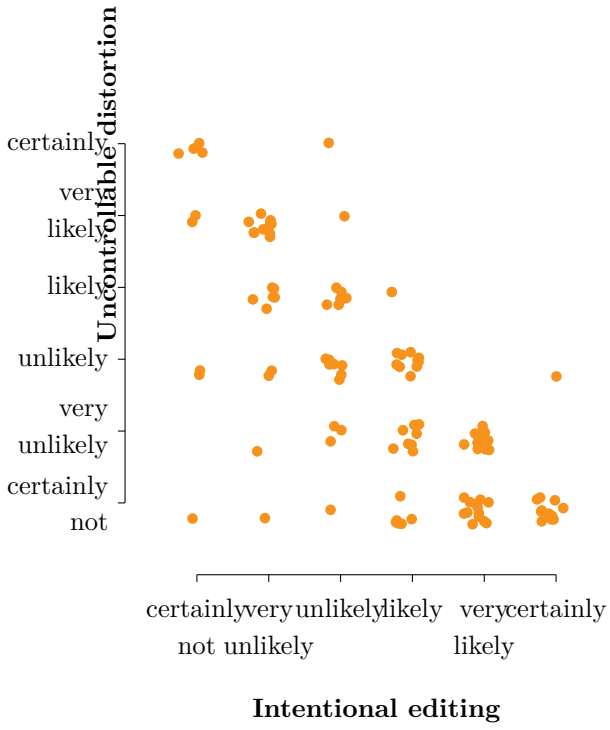


Figure B.14: Suspected causes of the differences in the M-Bag. Unlike Figure B.11, this scatter plot shows responses of individual participants (with jitter applied for visibility). The negative correlation is not perfect as rejecting one cause does not imply the support of the other.

B.3.2 Supplemental Tables of Control Questions

Table B.3: Descriptive statistics of the control questions

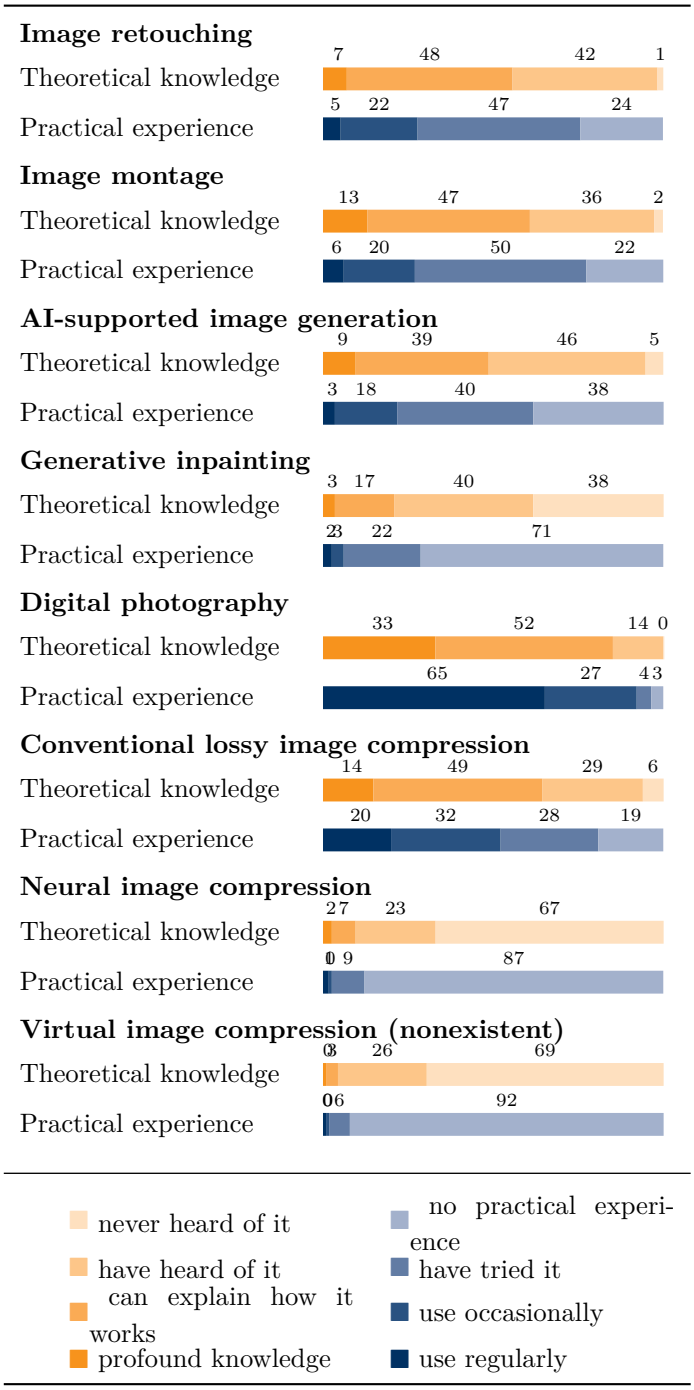


Table B.4: Descriptive statistics of the control questions (cont'd)

Verification of images from different sources (%)

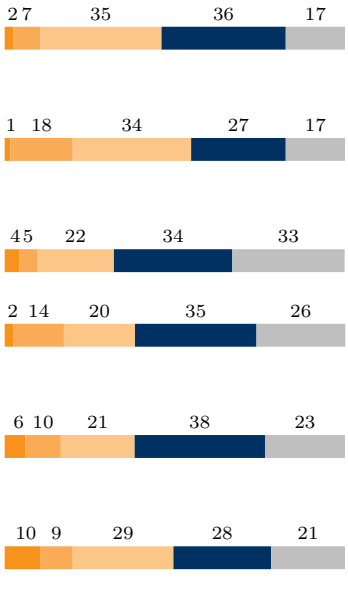
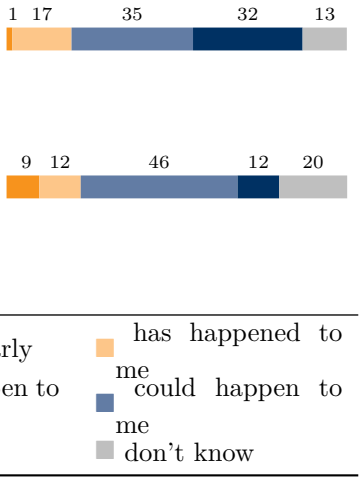


Table B.5: Descriptive statistics of the control questions (cont'd)

Experience with the hypothetical scenario (%)



C. A Research Dataset of Miscompressions

Authors

Nora Hofer, University of Innsbruck
Rainer Böhme, University of Innsbruck

Title

Challenging Cases of Neural Image Compression: A Dataset of Visually Compelling Yet Semantically Incorrect Reconstructions

Conference

ACM International Conference on Multimedia (MM '25)
Dublin, Ireland · October, 27–31, 2025

Abstract

Preserving the semantic integrity of image details is difficult in neural image compression. Failure to do so can result in *miscompressions*: reconstruction errors that change the meaning between the original and reconstructed images. Undetected miscompressions can compromise the reliability of reconstructed images and potentially reduce the accuracy of downstream computer vision tasks. To advance research on this problem, we present SCLIC, a curated dataset of 18k human-annotated miscompressions generated by 12 neural compression models. It includes images from three common benchmark datasets, compressed and reconstructed using codecs based on CNNs, GANs, diffusion models, and image transformers for different perceptual metrics and rate–distortion settings. We envision that this dataset will facilitate the development of strategies to mitigate miscompressions and enable more reliable neural image compression codecs.

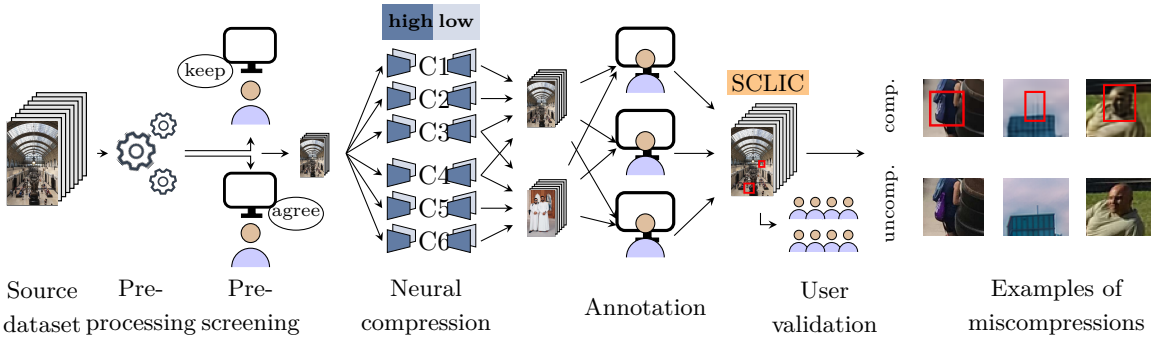


Figure C.1: The SCLIC dataset: Human labelers search and annotate reconstruction errors in image details that change the semantic, creating a dataset of 18k uniquely identified *miscompressions*. Each source image was compressed with four (of six) codecs at two quality settings. Images were prescreened for content and selected miscompressions validated in a user study.

C.1 Introduction

Machine learning is about to transform lossy image compression. Research in *neural compression* demonstrates that replacing conventional signal processing steps in the compression and decompression pipeline with learned elements achieves unprecedented levels of visual reconstruction quality, especially at low bit rates [6, 24, 35]. However, prior work has pointed out the risk of *miscompressions* [14]. Miscompressions are reconstruction errors in which the semantics of image details change after lossy compression. The examples shown to the right of Figure C.1 include a purple bag that has turned blue, an additional antenna on a roof, and a darkened skin tone. Unlike conventional lossy compression, neural reconstructions can mislead viewers because they lack cues indicating poor compression quality. These reconstructions tend to appear clean and compelling, which can create a false sense of trust.

Prior work has mentioned the problem [24, 35], proposed a taxonomy [14], and documented biases in the shift of semantics [27]. However, research into mitigations remains scarce, presumably for the lack of a suitable dataset. Building such a dataset is not straightforward as it requires assessing semantics at the level of image details, a task machines have yet to learn [20]. In this paper we present SCLIC,[1] a dataset of 18 019 annotated miscompressions identified by three trained human labelers spanning 10 828 images from popular benchmark datasets.

Figure C.1 shows the generation process described in this paper. We have defined four objectives for the dataset:

- *Scale*: The dataset should be large enough to train vision transformer models with examples of miscompressions.
- *Coverage*: The dataset should include images covering a variety of architectures, codecs, and compression rates.
- *Diversity*: The dataset should include a variety of scenes, perspectives, light conditions, exposures, and capturing devices to measure potential influencing factors of the input.

[1] Semantic Changes in Learned Image Compression

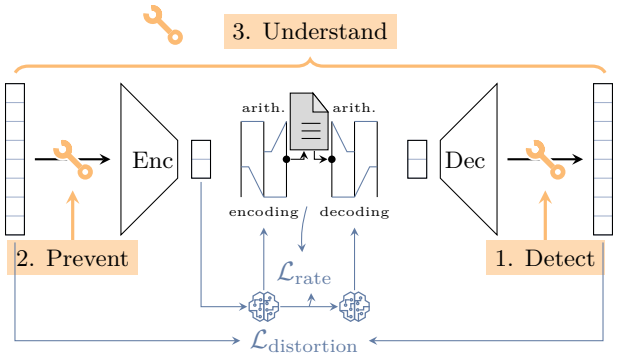


Figure C.2: Diagram of a neural compression codec. Wrenches show potential applications of our dataset (*cf.* Sect. C.2.3).

- *Reproducibility*: The dataset should facilitate reproducible research and its generation should be repeatable.

The remainder of this paper is structured as follows: Section C.2 recalls the concepts of neural compression, reviews existing work on miscompressions, and outlines mitigation approaches, from which we derive the requirements for our dataset. This links to the intended uses of the dataset. Section C.3 describes our processing and annotation pipeline, access instructions, as well as legal and ethical aspects. Finally, Section C.4 reports first insights using statistical analyses of the annotations before Section C.5 concludes our paper.
The full dataset, all supplemental material, and download instructions are accessible via: `https://zenodo.org/records/16780952`.

C.2 Primer on neural image compression

The conventional lossy compression pipeline has three components. First, an input image is **transformed** from the spatial domain into a domain where pixels are decorrelated and the variance is concentrated in fewer coefficients. A useful property of conventional transforms is that the distribution of the coefficients is known. The second component, **quantization**, is deliberately lossy. The quantization steps can be adjusted according to the relevance of each coefficient. Finally, **entropy coding** compresses the quantized coefficients into a bit stream. The better the input distribution is known, the shorter the resulting bit stream. Conventional codecs like JPEG [32] employ a blockwise discrete cosine transform (DCT) [2], followed by quantization using frequency-dependent tables, and entropy coding via run-length and Huffman schemes [15].
Figure C.2 shows the pipeline of *neural* image compression. It replaces fixed signal processing operators with learnable elements, such as deep convolutional neural networks (CNN). These encoders are optimized to capture nonlinear signal structures and yield compact latent representations [13]. Quantization is commonly done with simple rounding [6, 29]. Entropy coding has to deal with the unknown distribution of the latent space. Therefore, almost all neural compression codecs use a trained hyperprior autoencoder network [7] to learn the distribution of the latent space. The prediction of this model is then used to parameterize an arithmetic coder. The entire pipeline is trained end-to-end with a rate–distortion loss, where the rate component is derived from the entropy model and distortion is measured using pixel-wise and perceptual metrics. At inference time, the trained encoder and decoder are fixed. Different compression qualities require separate models for different

rate–distortion tradeoffs.

C.2.1 Codecs

The existing codecs differ in the architecture of the encoder and decoder networks. Early codecs implement variational autoencoders with CNNs on both sides. One branch of research focuses on using vision transformers for encoding. This approach is presumably inspired by the success of the attention mechanism in natural language processing (NLP) and computer vision tasks. The idea is to find an embedding for image blocks such that the encoder allocates more bits to more challenging areas (*i.e.*, edges, textures). The STF codec [38] uses transformers with window-attention modules to better capture local features in the input signal. Another example for a transformer encoder is the reference implementation of the draft JPEG AI standard [5], which is currently under development. Its high operation point (HOP) transform mode uses two transformer attention modules (one for luminance and one for chrominance) with attention blocks for adaptive channel-wise weighting.
Another branch of research focuses on the decoder side, refining generative networks to create visually appealing reconstructions. For example, the HiFiC codec [24] replaces the decoding network with a generative adversarial network (GAN) conditioned on the hyperprior. Similarly, the CDC codec [35] uses a diffusion variational autoencoder, also conditioned on the hyperprior.

C.2.2 Miscompressions

Modelling human perception is not trivial and has become an active field of research [10, 12]. Incorporating perceptual metrics in the loss function helps retain high perceptual quality at low bit rates. However, if perceptual metrics are given too much weight, networks may deviate from the input signal and tend to "make up" details during reconstruction. This puts semantic fidelity at risk. For example, the left crop in Figure C.1 shows a visitor at the Musée d'Orsay whose bag has changed from purple in the original (bottom) to blue after neural compression (top). Such changes match the definition of *miscompressions*, *i.e.*, discrepancies "between the semantic meaning of an original image (detail) and its reconstructed version after neural compression." [14, p. 3].
Miscompressions pose new risks that were absent in conventional compression. While visible artifacts in JPEG images indicate low reliability and may cause viewers to distrust the images, neurally compressed images often appear visually authentic, even if they convey false information. This can lead to (unintentional) misinformation and may cause safety and security risks. Previous work also points out ethical concerns: Qiu et al. [27] find that racial bias in neural codecs can cause miscompressions of specific ethnicity groups. African–American faces tend to be reconstructed to appear more Caucasian, while Caucasian faces largely retain their original features. The risks are not limited to human observers, but may potentially compromise the accuracy of downstream computer vision tasks. For instance, some evidence suggests that biometric features are vulnerable to miscompressions [9, 16, 23]. The risk of detection errors has been reported especially for iris images [9]. Worryingly, the proposed JPEG AI standard prominently mentions downstream tasks in public surveillance and autonomous driving scenarios as an intended application area for this codec [5, p. 104]. Risks increase further when adversaries can strategically modify the input signal to trigger bit stream collisions and gain control over the output [22].

C.2.3 Mitigations and requirements for a dataset

Mitigating the risk of miscompressions requires a dataset that captures many instances of the issue. We derive the requirements for our SCLIC dataset by discussing its potential applications in

mitigation efforts. Starting from the lower-hanging fruits, we first focus on detection, then discuss prevention, before moving to the challenge of understanding full causal relationships. The steps relate to the stages in the compression pipeline, as annotated in Figure C.2.

C.2.3.1 Detection

The first approach is to use a post-processing module to automatically detect miscompressions in reconstructed images. Such a detector could filter misleading outputs, recompress with higher bit rate, or notify the viewer of the reduced reliability [14]. This approach is related to the work of Tsereh et al. [31], who compile a dataset of 47k images containing compression artifacts that were annotated by human subjects on a crowdsourcing platform. Their dataset is used to train a detector for similar artifacts. Although it may help reduce compression artifacts in general, their dataset does not focus on semantically relevant reconstruction errors. This calls for a similar effort tailored to address the risk of miscompressions. Arguably, miscompressions should be addressed first because reducing general artifacts further might increase the risk of creating a false sense of trust. To be most useful, the dataset of annotated miscompressions should allow comparisons of miscompressed regions to their spatial neighborhood ("context") as well as to contrast regions that are *not* annotated as miscompressed but share similar features. More broadly, the idea to detect semantic changes in image details connects to Jiang et al. [17], who study event hallucinations in vision–language models. They propose using large language models (LLM) to detect invented narratives in image descriptions. A similar approach, refined to the level of image details, could compare text descriptions of original and reconstructed images to detect miscompressions. The SCLIC dataset should allow us to benchmark these approaches.

C.2.3.2 Prevention

While the detection of miscompressions in reconstructed images is currently the most feasible approach, it does not necessarily mitigate the risks for downstream computer vision tasks. The JPEG AI standard supports a region of interest (ROI) feature [3], which controls the allocation of bits to image regions. It should be explored whether this feature can prevent miscompressions, assuming that the positions where they occur are known or predictable. The SCLIC dataset should support experiments with the ROI mask, independent of the ability to predict miscompressions. Inspiration could also be taken from special image sources. For example, text integrity has been studied for learning-based compression of screen content. Zhou et al. [36] propose a codec that uses an external prior guidance module to improve the structural fidelity and preserve text. They identify relevant image regions and add weights to the loss function to guide the bit allocation to regions of interest during compression. Their approach should be tested on natural images, which differ substantially from screen content. Again, a ground-truth dataset is required to evaluate whether this approach will make miscompressions less likely.

C.2.3.3 Understanding

In order to develop neural compression that is immune to miscompressions and preserves details, we need to understand what causes miscompressions. Lieberman et al. [21] study the out-of-distribution performance of neural compression codecs by introducing low, mid, and high-frequency augmentations in the input images. Employing their tools to our dataset should allow for a better understanding of codec behavior. A first step would be to study the susceptibility to miscompressions of different decoder architectures. For example, Qiu et al. [27] report that diffusion models show the biggest bias for skin types, followed by VAEs and then GANs. By contrast, GANs exhibit the

strongest bias for eye types. While it is in principle possible to compare the existing pretrained models using the SCLIC dataset, this comparison should be interpreted with caution. The effect of the architecture is confounded with many other parameters. Finally, our annotated dataset could be a starting point to design new perceptual metrics tailored to specific kinds of miscompressions (*e.g.*, text, faces, color).

C.3 Method

Here we describe the dataset generation process shown in Figure C.1.

C.3.1 Preparation

To generate our dataset we took 2491 uncompressed images from the three benchmark **source datasets** CLIC [30], DIV2K [1], and RAISE [11]. **Preprocessing** was necessary to match the input dimensions of all codecs and to fit within the available GPU memory. We center-cropped the images to the largest multiples of 16 pixels and downscaled them to a maximum dimension of 2304 pixels using ImageMagick's `resize` tool. RAISE images came as TIFF and were converted to PNG with ImageMagick's `convert` tool. We removed 15 images that did not have three channels or were corrupted.

To reduce the number of images that have to be viewed by our labelers and ensure diversity of the dataset, we **prescreened** the images. We excluded any image that did *not* contain the following: humans or depictions of humans (*e.g.*, statues, drawings), symbols and signs (*e.g.*, text, religious, cultural, traffic, *etc.*), vehicles, buildings, other human-made structures (*e.g.*, fences, patterns, cables, *etc.*), and discernible reflections and shadows of objects. We also excluded images that portrayed large, close objects if they did *not* contain any small details, as they are very unlikely to be miscompressed. 912 images that were rated to be (borderline) excluded, by *both* researchers independently, were removed. Prescreening would have allowed us to filter out any content that could potentially trigger negative emotions or cause psychological harm to our labelers, but the source datasets did not contain any such image. Moreover, we hoped that excluding "boring" images would keep the labelers engaged and increase the quality of our dataset.

To **compress** the resulting dataset of 1563 images, we chose six codecs from the literature (**C1**–**6**) and selected two compression settings per codec such that they approximately matched the target bit rates of 0.25 (**low**) and 0.75 (**high**) bit per pixel (bpp).

C1:	Hyperp. MSE [7]	**low:** $\alpha = 3$	**high:** $\alpha = 7$
C2:	Hyperp. MS-SSIM [7]	**low:** 0.25 bpp	**high:** 0.75 bpp
C3:	HiFiC [24]	**low:** *HiFiC-lo*	**high:** *HiFiC-hi*
C4:	STF [38]	**low:** $\lambda = 0.067$	**high:** $\lambda = 0.025$
C5:	CDC xparam 0.9 [35]	**low:** $\lambda = 2048$	**high:** $\lambda = 512$
C6:	JPEG AI *HOB* [5]	**low:** 0.25 bpp	**high:** 0.75 bpp

The selection of codecs was based on our objective to cover a variety of different architectures and the availability of pretrained models. To strike a balance between codec variety, annotation workload, and the ability to compare codecs on the same images, we split the dataset in half and compressed each half using four different codecs. We selected two codecs (C3 and C6) to compress both halves. All other codecs were used to compress only one half, resulting in a total of 12 504 compressed images ($1563 \times 4 \times 2$). They were split into 8 *bulks* of 64 *batches* each. Batches contained 25 compressed

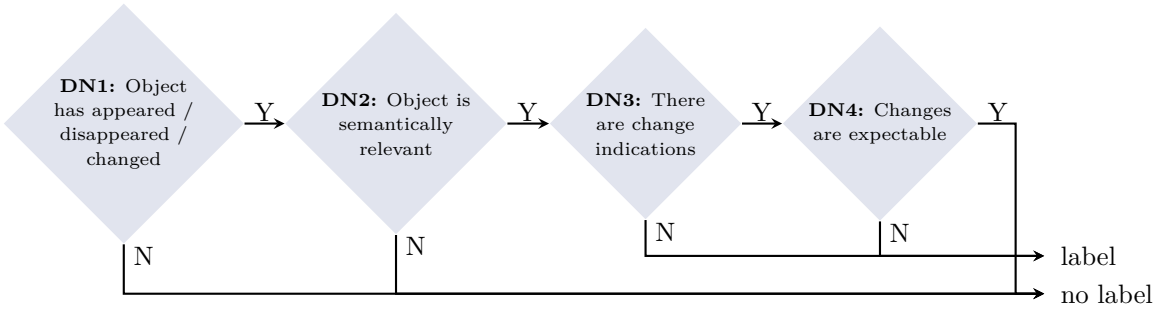


Figure C.3: Decision tree to disambiguate what constitutes a miscompression. A modification (DN1) of a semantically relevant object (DN2) is labeled as miscompression if there are *no* visible indications of the modification (DN3) or one would *not* expect the modification given the visible quality of the surrounding area (DN4).

images and were the smallest unit of work assigned to the labelers. Each batch was assigned to 1 of 8 bulks in a way that all compressed versions of one image appeared in the same bulk, but images within bulks (and batches) were in random order.

C.3.2 Instrument

The images were annotated batch-wise in a controlled lab environment by one of three labelers in the course of ten months. Each labeler viewed approximately the same number of images.

C.3.2.1 Interface

We used the `VPV` image viewer [4], which we extended to record the coordinates of annotated miscompressions. The labelers were instructed to draw tight bounding boxes around miscompressed objects or areas. They could toggle between the compressed and uncompressed version of the same image and zoom in or out as wanted. Images were shown full screen. Labelers could display the pixelwise difference in a color map on half of the screen.

C.3.2.2 Task

The labelers were provided with instructions that introduced them to *miscompressions*, described the goal of the project, the setup including `VPV`, and annotation instructions. Due to the subjective nature of semantics, annotating miscompressions is not a straightforward task. We conducted multiple training sessions to standardize the labelers' annotation behavior as much as possible.. The training included joint annotation sessions and in-depth evaluations of annotations in selected training images.

The core of the instructions was a decision tree, depicted in Figure C.3, which standardized the definition of miscompressions and guided labelers' decisions. We describe the tree using the example of the miscompression of the bag in Figure C.1 that changed from purple to blue: First, Decision Node 1 (DN1) filters for image areas that are visibly modified after compression. DN1 applies (Y) for blue bag. The second node, DN2, filters for modifications of objects that are semantically relevant, *i.e.*, if it is identifiable and carries semantic meaning. To improve the inter-subjectivity of this decision, we provide the labelers with examples of semantically important (*e.g.*, "a cross disappeared from a church tower") and unimportant (*e.g.*, "the color of a tree in a forest appears

Table C.1: Inter-labeler agreement measured on two batches

Units	Agreement (in %)				Krippendorff	
per image	Total		Positive only		α	
1	58.0	44.32 – 71.68	24.0	12.16 – 35.84	0.43	0.23 – 0.62
4	70.5	64.18 – 76.82	11.0	6.66 – 15.34	0.48	0.37 – 0.59
16	83.3	80.66 – 85.84	2.9	1.72 – 4.03	0.43	0.37 – 0.50
32	92.9	92.02 – 93.80	0.8	0.45 – 1.05	0.38	0.32 – 0.44
256	96.8	96.49 – 97.10	0.2	0.11 – 0.26	0.29	0.25 – 0.33

Three labelers on binary decisions. $\alpha > 0$ is the advantage over chance.
Ranges are 95% confidence intervals (using bootstrapping in the case of α).

darker") modifications. DN2 applies (Y) for the bag as the color has a semantic meaning and can be used to describe and identify *this specific* bag. DN3 checks for further indicators that could inform viewers about a modification and potentially allow them to imagine how the original version looked like before compression. DN3 does not apply (N) for the blue bag because there are *no* indications that would inform viewers that the bag was a different color before compression. This means that the bag is labelled as miscompression right away. Only if DN3 applies, the decision is passed over to DN4, which checks whether the modifications can be expected given the visible compression artifacts in the image (region). If the modified object is surrounded by well reconstructed areas, it is annotated. Also, DN4 does not apply (N) for the bag because there are *no* obvious changes in color or compression artifacts in the rest of the image. One would not expect the color of the bag to be different before compression.

C.3.2.3 Special cases

Some images contained *multi-miscompressions*. We use this term to describe miscompressions of multiple identical or similar objects within the same image. The labelers were instructed to annotate the first three instances and flag the image.

C.3.2.4 Inter-labeler agreement

To measure the agreement among the labelers, we assigned two batches to each of them. We asked them to annotate all instances of multi-miscompressions to avoid bias from disagreements over the selected "first three" instances. Since measuring agreement for drawing bounding boxes instead of labeling images is not standard, we need to define the unit on which we measure it. Table C.1 reports agreement scores for the labelers' binary decisions on different units. The top row shows the agreement when each image is taken as one unit. The remaining rows split each image into equally sized tiles, measuring agreement on *where* in the image the miscompression was annotated at increasing resolution. We consider units as annotated if the tile overlaps with more than 50% of at least one bounding box, ensuring that every annotation is assigned to exactly one unit.
Observe that the overall agreement ranges between 58 and 97%. However, these numbers are biased by the agreement on areas that *do not* contain miscompressions. The "positive only" column removes this effect by only counting agreement on positive units. These values are lower, but still beat random guessing by a margin. This is also evident from the Krippendorff's alpha estimates, which have confidence intervals that are strictly above zero (chance) in all cases. Values below 0.5 are common in natural language processing, particularly for semantic labels [25, 26]. We cannot expect a high α

in our context as the metric penalizes even mild disagreement in small coder groups, especially with binary decisions [19]. With only three labelers, any deviation from consensus has a strong impact.

C.3.3 Validation with a user study

Whether or not a semantic change of an image detail is considered critical is often subjective. In order to generalize beyond our (and the labelers') opinions, we have conducted a user study in a controlled lab setup. 115 participants have been shown 18 miscompressions from our dataset randomly mixed with a number of control images (uncompressed, neurally compressed but not miscompressed, or JPEG compressed). The participants had no prior exposure to neural compression and were asked to compare an "image taken by them" to one that was "received via social media" (see Fig. C.4). If they noticed a difference, they were asked to assess whether the difference can "lead to misunderstandings." On average, and after controlling for subject and image fixed effects, miscompressed images were rated 0.98 units more likely to cause misunderstandings than control images. The units refer to the 6-point rating scale depicted in Fig. C.4. The difference is statistically significant at the $p < 0.01$ level and the effect size was "large" by Cohen's $d = 0.86$. We are preparing a separate publication focusing on the theory, design, and analysis of this user study.

C.3.4 Access and licensing

The dataset includes two CSV files and the uncompressed, compressed, and reconstructed versions of all 1563 images. The first CSV file contains rows for all 18 019 annotations. It records the filename, compression model, and the annotated bounding box coordinates and dimensions. The second CSV file contains rows for all 18 756 compressed images. It records the images' filename, width and height, source dataset, compression model, bpp, PSNR, SSIM, and MS-SSIM. It also logs the number of annotations and potential multi-miscompressions that were found in the image. The tables can be merged by filename and compression model.

C.3.4.1 Download and reproducibility

The dataset is accessible on *Zenodo* via `https://zenodo.org/records/16780952`. A notebook to download and view example images is provided together with a script to sample and crop images according to a set of fixed parameter specifications. It also allows downloading the same area in images of different models, recording the number of annotations in the respective crop. The script is intended to facilitate reproducible research on miscompressions.

C.3.4.2 Licensing

All images are derived from the original sources and the licensing terms of the sources apply. We do not add any restrictions. The annotations are released under the Creative Commons–Attribution 4.0 International (CC BY 4.0) license.

C.3.4.3 Ethics and data protection

We are prepared to handle requests of data subjects depicted in our dataset who want to exercise their rights under the GDPR. In case of objection, we remove or anonymize the image data but retain the annotations with its coordinates. The users study has been approved by the University of Innsbruck's ethical review board. The annotation was carried out by three trained student assistants

Table C.2: Descriptive statistics of annotated miscompressions

Codec	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
% of viewed images that have at least one miscompression						
low	47.4	53.6	69.2	62.8	49.4	64.5
high	3.2	15.2	23.9	25.7	20.1	16.5
Mean number of annotations in images that have at least one miscompression						
low	3.5	4.4	5.4	4.4	3.6	4.9
high	1.6	2.6	2.6	2.5	2.6	3.9
Special case multi-miscompressions: % of miscompressed images that have at least one multi-miscompression						
low	13.4	44.3	36.5	51.8	10.2	47.7
high	15.8	20.9	17.0	30.3	5.0	33.9
Average size of annotations (length of one side in pixels)						
low	49.1	41.3	42.9	54.2	50.1	42.8
high	34.6	19.1	24.5	40.0	34.4	28.4
Total number of annotations						
low	997	1774	4990	2072	1066	4308
high	30	292	827	493	304	866
Total number of viewed images						
low	597	758	1352	753	597	1354
high	597	757	1356	759	592	1356

working an average of seven hours per week over a period of ten months. They were employed and compensated above minimum wage with social security coverage.

C.4 Dataset description

Table C.2 reports key statistics broken down by codecs (in columns) and compression settings (low/high). Overall, higher quality settings result in fewer miscompressed images, fewer miscompressions per image, and smaller miscompressions. Before starting this project, we did not know about the frequency with which miscompressions occur. We can see that approximately one in every two images has at least one miscompression at the low quality setting (≈0.25 bpp). For the high quality setting (≈0.75 bpp), this ratio drops to about one in five images. Between 5 and 50% of the images with at least one miscompression contain multi-miscompressions. The share varies significantly between codecs. The fact that multi-miscompressions are not rare means that they need special attention when training models with this dataset. Note that crops taken from multi-miscompressed images are not guaranteed to be free of miscompressions even outside of all annotated bounding boxes.

It is tempting to interpret the differences between codecs as benchmarks or as indications of the performance of the underlying architectures. Figure C.5 prevents us from making premature conclusion. It shows that the median annotated area per image (*i.e.*, the sum of pixels in all annotations per image divided by the total number of pixels) can be explained to a large extent with differences in the bit rate. Not all codecs closely match the target bit rate for each image; therefore, inter-image heterogeneity substantially influences the outcome. Additionally, note that the interquartile ranges

overlap significantly between the low and high parameter settings. This suggests that high bit rates do not guarantee an absence of miscompressions. Codec C1 stands out with significantly fewer miscompressions in the high setting. Future research should investigate whether this is primarily due to the high bit rate or caused by its architecture. The decoder of C1 does not include generative elements that are prone to hallucinations.

C.5 Conclusion

Miscompressions remain an under-researched challenge in the emerging field of neural image compression. With codec standardization underway and deployment in mobile phones on the horizon,[2] understanding and mitigating such semantic changes becomes increasingly important, also outside research labs.[3] To pave the way for future work on miscompressions, we present the first curated dataset designed to support mitigation efforts. This dataset is unique in that it focuses on the semantics of image details, a task that currently requires human judgment.
This focus has some limitations. Due to the significant manual effort required and our goal of collecting enough samples to train neural networks, annotations were collected by individual trained labelers without overlap (except for measuring inter-labeler agreement, *cf.* Sect. C.3.2.4). Future versions of the dataset could include verifications and attributes, *e.g.*, based on existing taxonomies [14]. Moreover, we may have overlooked semantically relevant changes that require domain-specific expertise, which can be critical for tasks such as plant or animal classification. Future studies could address this by incorporating such expertise and using domain-specific image sources. We refer the reader to the supplemental material for more details and invite them to explore the dataset.

Acknowledgements

We thank our labelers Leny Barry, Valerie Huter, and Max Ninow for many hours of concentrated work and for sharing useful insights from inspecting thousands of images; the anonymous participants of the user study; and Martin Beneš and Kristina Magnussen for their comments on earlier versions of this manuscript. We gratefully acknowledge funding by the state of Tyrol (F.50541/6-2024).

References

[1] E. Agustsson and R. Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[2] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, pages 90–93, 1974.

[3] E. Alshina, J. Ascenso, and T. Ebrahimi. JPEG AI: the first international standard for image coding based on an end-to-end learning-based approach. *IEEE MultiMedia*, 31(4):60–69, 2024.

[4] J. Anger. vpv: Image viewer designed for image processing experts. (v0.8.2), 2023.

[5] J. Ascenso, E. Alshina, and T. Ebrahimi. The JPEG AI standard: providing efficient human and machine visual data consumption. *IEEE MultiMedia*, pages 100–111, 2023.

[2] "[T]he first ever implementation [...] of JPEG AI encoder and decoder on their mobile phone" `https://www.linkedin.com/posts/touradjebrahimi_wearejpeg-activity-7346065622880976896-Izoh` (posted: July 2025; accessed: August 2025)

[3] Readers may recognize miscompressions as reminiscent of the flaws in optical character recognition found in copy machines that randomly alter digits in documents [18]. Miscompressions may affect a broad set of image details, not just digits.

[6] J. Ballé, V. Laparra, and E. Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

[7] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

[8] J. Ballé, S. J. Hwang, and E. Agustsson. TensorFlow Compression: Learned data compression, 2024.

[9] S. Bergmann, D. Moussa, and C. Riess. Trustworthy compression? impact of AI-based codecs on biometrics for law enforcement. *arXiv preprint arXiv:2408.10823*, 2024.

[10] S. Bhardwaj, I. Fischer, J. Ballé, and T. Chinen. An unsupervised information-theoretic perceptual quality metric. In *Advances in Neural Information Processing Systems*, pages 13–24, 2020.

[11] C. Dang-Nguyen, D.and Pasquini, V. Conotter, and G. Boato. RAISE: A raw images dataset for digital image forensics. In *Multimedia Systems Conference*, pages 219–224. ACM, 2015.

[12] K. Ding, K. Ma, S. Wang, and E. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *Transactions on Pattern Analysis and Machine Intelligence*, pages 2567–2581, 2020.

[13] Z. Duan, M. Lu, Z. Ma, and F. Zhu. Opening the black box of learned image coders. In *Picture Coding Symposium*, pages 73–77. IEEE, 2022.

[14] N. Hofer and R. Böhme. A taxonomy of miscompressions: Preparing image forensics for neural compression. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2024.

[15] D. Huffman. A method for the construction of minimum-redundancy codes. *IRE*, pages 1098–1101, 1952.

[16] E. Jalilian, H. Hofbauer, and A. Uhl. Iris image compression using deep convolutional neural networks. *Sensors*, 22(7):2698, 2022.

[17] C. Jiang, H. Jia, M. Dong, W. Ye, H. Xu, M. Yan, J. Zhang, and S. Zhang. Hal-eval: A universal and fine-grained hallucination evaluation framework for large vision language models. In *ACM Multimedia*, pages 525–534, 2024.

[18] D. Kriesel. Xerox scanners/photocopiers randomly alter numbers in scanned documents, 2013. (accessed: August 2025).

[19] K. Krippendorff. Computing krippendorff's alpha-reliability, 2011. (accessed: August 2025).

[20] E. Lei, Y. Berkay Uslu, H Hassani, and S. Bidokhti. Text+ sketch: Image compression at ultra low rates. In *International Conference on Machine Learning 2023 Workshop Neural Compression: From Information Theory to Applications*, 2023.

[21] K. Lieberman, J. Diffenderfer, C. Godfrey, and B. Kailkhura. Neural image compression: Generalization, robustness, and spectral biases. In *International Conference on Machine Learning 2023 Workshop Neural Compression: From Information Theory to Applications*, 2023.

[22] J. Madden, L. Dorje, and X. Li. Bitstream collisions in neural image compression via adversarial perturbations. *arXiv preprint arXiv:2503.19817*, 2025.

[23] D. Mari, S. Cavasin, S. Milani, and M. Conti. Effectiveness of learning-based image codecs on fingerprint storage. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2024.

[24] F. Mentzer, G. Toderici, M. Tschannen, and E. Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 2020.

[25] L. Mertens, E. Yargholi, H. Op de Beeck, J. Van den Stock, and J. Vennekens. FindingEmo: An image dataset for emotion recognition in the wild. *Advances in Neural Information Processing Systems*, 37:4956–4996, 2024.

[26] H. Pardawala, S. Sukhani, A. Shah, V. Kejriwal, A. Pillai, R. Bhasin, A. DiBiasio, T. Mandapati, D. Adha, and S. Chava. SubjECTive-QA: Measuring subjectivity in earnings call transcripts' QA through six-dimensional feature analysis. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[27] T. Qiu, A. Nichani, R. Tadayontahmasebi, and H. Jeong. Gone with the bits: Revealing racial bias in low-rate neural compression for facial images. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pages 1862–1889, 2025.

[28] A. Schlögl, N. Hofer, and R. Böhme. Causes and effects of unanticipated numerical deviations in neural network inference frameworks. *Advances in Neural Information Processing Systems*, 2024.

[29] L. Theis, W. Shi, Q. Cunningham, and F. Huszár. Lossy image compression with compressive autoencoders. *International Conference on Learning Representations*, 2017.

[30] G. Toderici, W. Shi, R. Timofte, L. Theis, J. Balle, E. Agustsson, N. Johnston, and F. Mentzer. Workshop and challenge on learned image compression (clic2020). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[31] D. Tsereh, M. Mirgaleev, I. Molodetskikh, R. Kazantsev, and D. Vatolin. JPEG AI image compression visual artifacts: Detection methods and dataset. *arXiv preprint arXiv:2411.06810*, 2024.

[32] G. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, pages 30–44, 1991.

[33] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018.

[34] T. Xue, B. Chen, J. Wu, D. Wei, and W. Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127:1106–1125, 2019.

[35] R. Yang and S. Mandt. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 2024.

[36] F. Zhou, X. Huang, P. Zhang, M. Wang, Z. Wang, Y. Zhou, and H. Yin. Enhanced screen content image compression: A synergistic approach for structural fidelity and text integrity preservation. In *ACM Multimedia*, pages 7900–7908, 2024.

[37] R. Zou. Googolxx/stf: Pytorch implementation of the paper "the devil is in the details: Window-based attention for image compression, 2023.

[38] R. Zou, C. Song, and Z. Zhang. The devil is in the details: window-based attention for image compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

C.A Dataset preparation

We selected three benchmark datasets. To account for CUDA memory limits and have comparable image sizes between datasets, all images were resized if needed to a maximum of 2304 pixels on each side with ImageMagick's `resize`. [4] Table C.3 summarizes the source datasets. Column 2, *# available* refers to the size of the original dataset. The columns 3 and 4 refer to the number of images before (*# considered*) and after (*# selected*) our data pre-processing, described in Section 3.1 of the main paper. Column 5, *# viewed* refers to the number of images that were viewed by the labelers.

[4] `https://usage.imagemagick.org/resize/`

Table C.3: Summary of source datasets.

Dataset	# available	# considered	# selected	# viewed	Source
CLIC2020 v1.0 Testset*	428	428	379	336	TF datasets
CLIC2020 v1.0 Trainset*	1 633	0	0	0	TF datasets
CLIC2020 v1.0 Validset*	102	102	81	72	TF datasets
CLIC2024 Testset	32	32	29	25	compression.cc
CLIC2024 Validset	29	29	27	22	compression.cc
DIV2K v2.0 Trainset HR	800	800	603	523	TF datasets
DIV2K v2.0 Validset HR	100	100	62	53	TF datasets
RAISE**	7 441	500	382	332	loki.disi.unitn.it/RAISE
Total	10 565	2 491	1 563	1 363	

∗ mobile and professional ∗∗ all camera models; categories: People, Indoor, Objects, Buildings

Table C.4: Summary of codecs used to compress images included in the dataset.

	Hyper I [7]	**Hyper II [7]**	**STF [38]**	**HiFiC [24]**	**CDC [35]**	**JPEG AI [5]**
Preprocessing	–	–	–	–	–	YUV
Transform	VAE/CNN		VAE/Att.	VAE/CNN	VAE/CNN	VAE/Att.
Entropy distrib.	Hyperprior					
Reconstruction	VAE		VAE/Att.	GAN	Diffusion	VAE/Att.
Optim. metric	MSE	MS-SSIM	MSE	MSE, LPIPS	LM, LPIPS	MSE, MS-SSIM

C.1.1 Source datasets

C.1.1.1 CLIC [30]

We downloaded images of the *clic2020 v1.0* test- and validation sets, including both mobile and professional.[5] We omitted the v1.0 trainset. The *clic2024* test, and validation set (32 and 29 respectively) was downloaded from the CLIC competition page.[6]

C.1.1.2 DIV2K [1]

We downloaded images of the *div2k v2.0 HR* train and validation set. [7]

C.1.1.3 RAISE [11]

We randomly sampled images (from all camera models) of the categories Indoor, People, Objects, and Buildings.[8] RAISE images are very large (approximately 7 times the size of DIV2K images) and caused CUDA memory errors for some models, so we resized them to 40% or a maximum dimension of 2304 pixels.

C.1.2 Codecs

We selected six compression codecs from the literature. Table C.4 gives an overview of their architectures.

[5] `https://www.tensorflow.org/datasets/catalog/clic`
[6] `https://storage.googleapis.com/clic2023_public/`
[7] `http://data.vision.ee.ethz.ch/cvl/DIV2K/`
[8] `http://loki.disi.unitn.it/RAISE/`

C.1.2.1 C1/C2

The **Hyperprior construction [7]** forms the basis of most neural compression schemes in the literature. We use the two pretrained models optimized for MSE at α 3 (C1-lo) and 7 (C1-hi), and two models optimized for MS-SSIM with target bitrates of 0.25 (C2-lo) and 0.75 (C2-hi) bpp. [9] The hyperprior models were trained on approximately 1 million 256×256 pixel crops of web scraped color JPEG photographs[10], downsampled by a randomized factor to minimum heights/widths between 640 and 1200 pixels from original heights/widths between 3000 and 5000 pixels.

C.1.2.2 C3

The **High-Fidelity generative image Compression (HiFiC) codec [24]** implements a decoder as a generative adversarial network (GAN) conditioned on the hyperprior described above. HiFiC's loss function incorporates rate in bits, distortion in MSE and perception in LPIPS [33]. The HiFiC models were trained on 256×256 pixel crops of "a large set of high-resolution images" [24, p.5] that were scraped from the web and downsampled to a random size between 500 and 1000 pixels. We include two pretrained HiFiC models optimized for the target bitrates 0.14 (C3-lo), called HiFiC^{lo} and 0.45 (C3-hi), called HiFiC^{hi}, available in `compression`.

C.1.2.3 C4

The **Symmetrical TransFormer (STF) framework** [38] uses transformers with window-attention modules in the transform network. The models were trained on 300k 256×256 pixel crops of JPEG images from the OpenImages dataset of heights/widths between 1200 and 1600 pixels and optimized for MSE and MS-SSIM. We include two pretrained models optimized for MSE and choose $\lambda = 0.0067$ (C4-lo), and $\lambda = 0.025$ (C4-hi) to get visibly pleasing reconstructions and match the bit rates of approximately 0.25 and 0.75 bpp. [11]

C.1.2.4 C5

The **Conditional Diffusion Compression (CDC) [35]** models are trained on 90k 256×256 pixel crops taken of frames taken from clips of the Vimeo-90k dataset [34], optimizing a rate-distortion-perception trade-off between using size in bits, a Lagrange multiplier (LM) and the LPIPS perception metric. We include two pretrained CDC models optimized for the Lagrange multiplier values $\lambda = 2048$ (C5-lo) and $\lambda = 0512$ (C5-hi). Note that a larger value means lower image quality and smaller file size.

C.1.2.5 C6

The **JPEG AI reference implementation [5]** is part 3 of the Rec. ITU-T T.840.1 | ISO/IEC 6048-1 JPEG AI Standard, which is under review by the ISO, at the time of writing[12]. While the standard specifies decoder requirements, the repository includes implementations for both encoder and decoder. Also, JPEG AI is trained end-to-end and uses the hyperprior prediction to model the distribution of the latent space for entropy coding, but differs from previous codecs in several aspects. Inspired by conventional compression methods, JPEG AI separates luminance from

[9] Available in TensorFlow's `compression` library $v2.17.0$ [8]

[10] Non-photograpic images were detected by saturation levels.

[11] Pretrained models were provided by the authors [37]. Unfortunately, no weights for models optimized for MS-SSIM were available at the time of writing.

[12] `https://www.iso.org/standard/88911.html?browse=tc`

chrominance information by converting the input image form RGB to the YUV444 color space, and processes both channels separately, which allows for grayscale-only reconstructions. Luminance is transformed to a latent of 160 dimensions and the chrominance channels are processed as a stacked latent tensor of 96 dimensions. The JPEG AI reference implementation provides two transform networks. The High Operation Point (HOP) transform are two transformer attention modules with channel attention blocks for adaptive channel-wise weighting. The Baseline Optimized Prediction (BOP) is a CNN based variational autoencoder transform intended to be used on devices with limited compute. The implementation provides three decoder networks of different complexities (8, 23, and 216 kMAC/pixel), and multiple pre-/post-processing filters. The quantization is done by rounding latent values to the nearest integer. Their latent's entropy is predicted with the hyperprior model and encoded by an arithmetic encoder. The implementation supports progressive decoding, realized by reconstructing the latent space of the entropy prediction model, and partial reconstruction by tiled processing of the input image. The JPEG AI models were trained, optimizing for rate in bits, MSE, and MS-SSIM on a specifically constructed dataset of approximately 5k PNG images of different resolutions from 256×256 to 8K pixels. We select two versions of the transformer based HOP model with all filters off, optimized for the target bit rates 0.25 (C6-lo) and 0.75 (C6-hi). We instruct JPEG AI at version 0.7, commit 50ec147866a51da33c90065aefdbd770cc1723a6 with `"config": ["tools_off.json", "oper_pointhop.json"]` configuration. A new version was released in January 2025. As we had already started labelling, we verified that the new version did not introduce different distortions for the same configuration and decided to stick to v 0.7 for consistency. We opted for all tools off, because this way, we were able to get deterministic images (at least on the same machine).

C.1.3 Data management

For annotation, we distributed the compressed dataset (two halves, each compressed with four codecs, two codecs were used in both halves) into eight *bulks*. Each bulk contained reconstructions of 200 distinct source images, distributed randomly into 64 batches, each containing 25 images. Before annotation, we split the dataset into two halves such that each half contains all images compressed with 4 models. Bulks 1–4 were compressed with C1, C3, C5, and C6. Bulks 5–8 were compressed with C2, C3, C4, and C6. The compressed dataset available on Zenodo (`https://zenodo.org/uploads/16780952`) contains the whole dataset compressed with all six codecs.

To avoid bias of the labelers based on the model or on the dataset, we renamed the images. Using Python's `hashlib` module, we created a 160-bit hash value of the SHA-1 hash (checksum) of the file's content rendered as a 40-char hex string. The files were read in chunks of 8 KB, updating the hash. All files were stored separated into 256 subdirectories of the filenames' first two characters.

Due to time constraints, bulk 4 is not fully annotated on submission date. Missing annotations will be added shortly in v 1.1.0. of the dataset. The batching strategy allows us to statistically describe our dataset without these images.

C.1.4 Compression

Figure C.5 shows the achieved compression rates over the whole dataset. Codecs C1, 3, and 4 did not allow setting a target bit rate so we chose a quality parameter to best match the target rates of 0.25 and 0.75 bpp.

Table C.5: Aggregate bit rates over 1563 images.

Configuration	target	mean	STD	25%	50%	75%
C1-lo Hyper mse	-	0.27	0.17	0.15	0.23	0.35
C2-lo Hyper msssim	0.25	0.25	0.04	0.22	0.25	0.27
C3-lo HiFiC	0.14	0.15	0.06	0.11	0.15	0.19
C4-lo STF	-	0.26	0.17	0.13	0.22	0.35
C5-lo CDC	-	0.25	0.06	0.21	0.25	0.30
C6-lo JPEG-AI	0.25	0.26	0.02	0.26	0.26	0.26
C1-hi Hyper mse	-	1.08	0.58	0.65	0.99	1.42
C2-hi Hyper msssim	0.75	0.74	0.10	0.66	0.74	0.82
C3-hi HiFiC	0.45	0.44	0.18	0.31	0.43	0.55
C4-hi STF	-	0.52	0.34	0.27	0.45	0.70
C5-hi CDC	-	0.63	0.10	0.57	0.64	0.71
C6-hi JPEG-AI	0.75	0.76	0.05	0.68	0.79	0.79

C.1.4.1 Determinism

Most models produced indeterministic outputs, *i.e.*, reconstructed images differed for the same input image. To avoid this we fixed all random seeds. Note, that depending on the hardware architecture, rerunning even with a fixed seed might still result in different images [28].

C.1.5 Hardware

The compression was executed on a shared GPU cluster using a 64-core Nvidia A100 GPU.

C.B Instrument

C.2.1 Setup

Each labeler annotated independently with no overlap in batch assignments. A progress file listed all batches and their annotation status. To balance the workload and reduce human bias, batches were assigned to labelers roughly evenly. Labelers could annotate batches at their own speed and were encouraged to take breaks or switch tasks to reduce fatigue-related bias. Images were viewed and annotated on two SI iMacs with 24" Retina $4-5^K$ screens (4480×2520 resolution at 218 ppi). An image pair including the compressed and uncompressed version was displayed as a stack in the `VPV` image viewer [4]. Labelers could toggle between two versions of the same image using the space bar and navigate through all images of the batch, using arrow keys. They could zoom in and out as wanted and display a color map to see pixel differences as shown in Figure C.6. We extended `VPV` for labelling with the feature to draw a selection window over a miscompressed region. The coordinates of the selected window were recorded automatically.

C.2.2 Instructions

C.2.2.1 Miscompressions

We instructed the labelers to annotate each miscompression separately, and draw tight bounding boxes around the miscompressed objects. They used the decision tree, described in Figure 3 of

the main paper as annotation guideline. In cases where they were unsure, they were instructed to annotate. We provided further hints for each decision node:

Changes Labelers were instructed to consider a change for annotation in the following scenarios:

- an identifiable object appears
- an identifiable object changes to a different identifiable object
- an identifiable object disappears completely

Semantic relevance We provided example images and descriptions of semantically (ir)relevant changes:

- Semantically relevant: a cross disappeared from a church tower, a ring disappeared from a finger, brake lights went off, a birthmark disappeared, the color of a window changed, a person in the background disappeared.
- Semantically irrelevant: a single star in a picture of the Milky Way is missing, the color of a tree in a forest appears darker.

Indications Indicators are any artifacts that allow a viewer to be able to tell that something was there from the compressed image alone.

Expectation Labelers were instructed to annotate in the following scenarios:

- A disappearing object is a miscompression when you would not expect it to disappear given the compression of the surrounding.
- If the whole area is well reconstructed, but something disappears, this would be unexpected. (One would not assume that something was there when looking at the reconstruction alone.)

C.2.2.2 Multimiscompressions

Frequently, images contained *multimiscompressions*, *i.e.*, multiple instances of the same miscompression in the same image, as shown in Figure C.7. Labelers were instructed to annotate the first three occurrences and take a note regarding the type of multimiscompression in the batch CSV file. The following types were defined:

- `color` – if colors of multiple objects are missing or change.
- `merging` – multiple instances of objects blurred into the background (often houses, faces, *etc.*).
- `irregular_blurring` – inconsistency whether background or objects are blurring or not.
- `texture` – texture or material of multiple objects seem different (*e.g.*, windows to patterns).
- `reshaping` – shape, geometry (bending of multiple lines or corners, changes in geometry, *e.g.*, straight fence to curved fence).
- `noteworthy` – when a notable phenomenon occurred, e.g., unexpected or anomalous colors appeared
- `to_discuss` – whenever you don't know what to do and want to discuss.

These reoccurring types were determined during the process of labeling and were therefore not recorded for images in earlier batches. To avoid missing values, these notes are not included in the dataset v 1.0.0., but might be added in a later version.

C.2.2.3 Overall appearance

Changes in color tones, structure in the background, blurring or focus can give a different overall appearance to the image. Labelers were instructed to annotate and note this in the CSV file.

- Different colors in the sky, sea, grass, *etc.* can change the impression of weather, season, climate, or time.
- Significant blurring on certain objects with comparatively less blurring on others can indicate a focus on certain objects, suggest different depth of the image, thereby suggesting a change in camera angle or focus.
- Irregular blurring of walls, buildings, fabric *etc.* can give a different impression of condition or quality.
- The annotation protocol excludes smoothing or blurring of landscapes in the background, but indicates to be precise with identifiable objects.

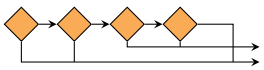
C.2.3 Labeler training

Training included joint labeling sessions and in-depth discussions of individual potential annotations in two training images.
First, the three labelers and the main researcher annotated both images independently. Then, all annotations were discussed in detail. Based on the decision tree, we evaluated whether it constitutes a miscompression, and should be annotated for the dataset. Training images are shown in Figure C.8 and Figure C.9. The compressed and uncompressed versions are available at `https://fileshare.uibk.ac.at/d/cd6d5c7f9d954f18a19a/`.

C.2.3.1 Image I: Detected changes that <u>should</u> be annotated

1. Birthmark on right person's face missing - right cheek

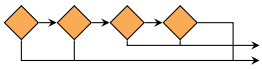
- What: An identifiable object disappeared.
- Semantic relevance: Birthmarks are biometric identifiers.
- Indication: There is no remaining indication of a birthmark left - one could not guess that there was a birthmark.
- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change; additionally, one would not expect this to happen when using conventional compression algorithms.



2. Hair on arm of right person appears like a scar - right arm

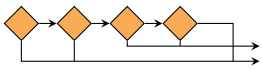
- What: An identifiable object appears.
- Semantic relevance: A scar has semantic meaning.
- Indication: One would not guess that hair turns into what looks like a scar.

- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change; additionally, one would not expect this to happen when using conventional compression algorithms.



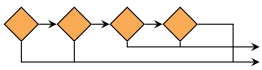
3. Watch face unreadable on right person - right arm

- What: An identifiable object changed.
- Semantic relevance: Whether the watch face is readable/on or not, does have an impact on the semantic meaning/description of the image.
- Indication: One would not guess that the watch was readable before.
- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change; additionally, one would not expect this to happen when using conventional compression algorithms.



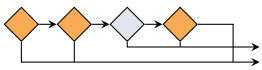
4. Thumb looks scarred on right person - right hand

- What: An identifiable object changed;
- Semantic relevance: A scar has a semantic meaning;
- Indication: One would not guess that the thumb was healthy on the original image, or the nail would turn into a scar and be locally shifted;
- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change; additionally, one would not expect this to happen when using conventional compression algorithms.



5. Nail missing on left person's thumb - left hand

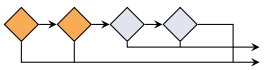
- What: An identifiable object changed.
- Semantic relevance: Nails might have a semantic meaning.
- Indication: One would consider a missing thumbnail.
- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change. One would not expect this to happen when using conventional compression algorithms.



C.2.3.2 Image I: Detected changes that <u>should not</u> be annotated

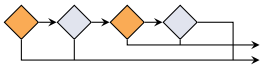
1. Light birth marks or freckles on left person's face missing

- What: An identifiable object disappeared.
- Semantic relevance: Small marks might also have a semantic meaning / be considered as identifiers;
- Indication: One would guess that there might be small/light marks on the skin, not visible in the image.
- Expectation: Given the (local) quality and visible compression artifacts, one could expect this change; one might also expect this to happen when using conventional compression algorithms.



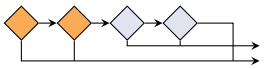
2. Right frame temple on left person's glasses missing

- What: An identifiable object disappeared.
- Semantic relevance: If the temple of glasses is visible, does have little semantic meaning.
- Indication: One would not recognize the absence of the temple.
- Expectation: Given the (local) quality and visible compression artifacts, one could expect this change; one might also expect this to happen when using conventional compression algorithms.



3. Right person's lips change color

- What: An identifiable object changed.
- Semantic relevance: Might be an indication for illness;
- Indication: One would consider the existence of a reflection.
- Expectation: Given the (local) quality and visible compression artifacts, one might also expect this change.

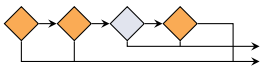


C.2.3.3 Image II: Detected changes that <u>should</u> be annotated

1. Sign "Halteverbot" - left of central house

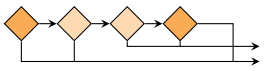
- What: An identifiable object disappeared.
- Semantic relevance: A sign has a semantic meaning.
- Indication: One could guess that there is something at the wall of the house (maybe but not a traffic sign).

- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change; additionally, one would not expect this to happen when using conventional compression algorithms.



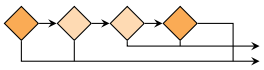
2. Lantern at left side of central house

- What: An identifiable object disappeared.
- Semantic relevance: The existence of a lantern / a streetlight might be relevant.
- Indication: The mount on the wall of the house could be a hint to some missing object.
- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change; additionally, one would not expect this to happen when using conventional compression algorithms.



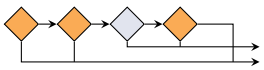
3. Roof shadow - central house

- What: An identifiable object changed.
- Semantic relevance: Shadows are relevant in image forensics to check for manipulations on photos.
- Indication: One would not guess that the profile of the shadow changes.
- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change; additionally, one would not expect this to happen when using conventional compression algorithms.



4. Top characters at clock of tower

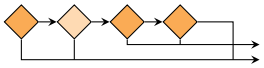
- What: An identifiable object changed.
- Semantic relevance: The character appears to be a Roman numeral, which could be historically relevant.
- Indication: One would guess that on a clock with Roman lettering a 'XII' is at the top.
- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change; additionally, one would not expect this to happen when using conventional compression algorithms.



5. Bottom sign at right house

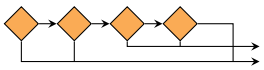
- What: An identifiable object changed.
- Semantic relevance: The sign could be some kind of certificate which could be identified by its layout.

- Indication: One would not guess that the color changes and the top left corner gets cut off.
- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change; additionally, one would not expect this to happen when using conventional compression algorithms.



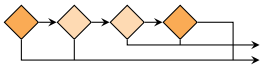
6. Chain in front of red car

- What: An identifiable object disappeared.
- Semantic relevance: Whether the chain is missing or not is relevant.
- Indication: There is no indication for the existence of the chain.
- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change; additionally, one would not expect this to happen when using conventional compression algorithms.



7. Antenna at roof of second house from left side

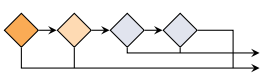
- What: An identifiable object disappeared.
- Semantic relevance: The existence of an antenna can have a semantic meaning.
- Indication: One could guess that there has been an object, but it could also be just dirt on the roof.
- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change; additionally, one would not expect this to happen when using conventional compression algorithms.



C.2.3.4 Image II: Detected changes that <u>should not</u> be annotated

1. Camouflaged poles on the ground, right square

- What: An identifiable object changed.
- Semantic relevance: Color of the poles might be relevant.
- Indication: One can still identify the poles as such and recognize that they matched the background colors.
- Expectation: Given the (local) quality and visible compression artifacts, one could expect this change.

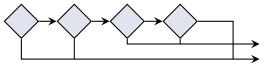


2. Object with red line at bottom left side of statue

Table C.6: Inter-labeler agreement breakdown with 95% confidence intervals

Unit	c-neg	c-pos	p-pos	p-neg	c-total
1	34.00 (20.87 – 47.13)	24.00 (12.16 – 35.84)	24.00 (12.16 – 35.84)	18.00 (7.35 – 28.65)	58.00 (44.32 – 71.68)
2	59.50 (52.70 – 66.30)	11.00 (6.66 – 15.34)	14.50 (9.62 – 19.38)	15.00 (10.05 – 19.95)	70.50 (64.18 – 76.82)
4	80.38 (77.62 – 83.13)	2.88 (1.72 – 4.03)	7.62 (5.79 – 9.46)	9.12 (7.13 – 11.12)	83.25 (80.66 – 85.84)
8	92.16 (91.22 – 93.09)	0.75 (0.45 – 1.05)	2.56 (2.02 – 3.11)	4.53 (3.81 – 5.25)	92.91 (92.02 – 93.80)
16	96.61 (96.30 – 96.92)	0.19 (0.11 – 0.26)	0.85 (0.69 – 1.01)	2.35 (2.09 – 2.61)	96.80 (96.49 – 97.10)

- What: A non-identifiable object changed.
- Semantic relevance: Non-identifiable, therefore no semantic meaning.



C.2.4 Inter-labeler agreement

C.2.4.1 Agreement score

Figure C.10 visualizes our approach to measure labeler agreement using colors for the four agreement scenarios (c–neg, c–pos, p–neg, p–pos). No labeler annotated in the gray-scale top–left and bottom-right units (c–neg), two annotated in the bottom-left (p–pos), and all labelers annotated at least one miscompression in the top-right unit (c–pos). This would result in agreement scores of c–neg: 0.5, c–pos: 0.25, p–neg: 0, p–pos:0.25.

C.2.4.2 Krippendorff's alpha

Krippendorff's Alpha [19] is a common approach to measure agreement across coder for a set of labeled items. It is defined as $\alpha = 1 - \frac{D_o}{D_e}$. The observed disagreement D_o is computed as the sum of disagreements among coder pairs across all items, normalized by the total number of coder pairs. The expected disagreement D_e is derived from the overall label distribution to estimate how often each pair of labels would co-occur by chance. Krippendorff's alpha is 0.686 for the single example image in Figure C.10.

C.2.5 User study

We have conducted the validation study with 115 (31 female, 80 male, 4 non-binary or other) German-speaking undergraduates in the age range from 19 to 40 (median 21). The data collection took place in January 2025 during the first 15 minutes of a weekly first-year computer science lab. The students were divided into eight lab sessions which ran partly in parallel in three consecutive time slots. The median response time for the entire study was $12'15''$ (quartiles $10'29''$ and $14'16''$). In each session, we started the experiment by handing out a printed briefing sheet that informed the students that their participation is voluntary, that all data will be fully anonymized, and the

Table C.7: Meta information for the sample miscompressions illustrated in the paper.

Reference	ID	model	description
Fig. C.1	328f0a7f-01L-b95007e0	CDC xparam 0.9 $\lambda = 2048$	Changed bag color
Fig. C.1	c0efe885-06L-48d97723	JPEG-AI 0.25bpp	Hallucinated antenna
Fig. C.1	e03dec99-06L-36e1eace	JPEG-AI 0.25bpp	Different skin tone
Fig. C.7	4154727c-06L-4ffe5f30	JPEG-AI 0.25bpp	Red flags disappear
Fig. C.13	01f0f60f-03L-38b9cdda	HiFiC lo	House appears demolished
Fig. C.14	2a7aac4b-03L-41776961	HiFiC lo	Changed eye color
Fig. C.15	2b73d284-03L-56c37ba9	HiFiC lo	Head disappeared
Fig. C.16	328f0a7f-05L-6e11a50c	CDC 2048	Statue disappeared
Fig. C.17	3141bfc1-03L-32cdf7fd	HiFiC lo	License plate is illegible
Fig. C.18	4154727c-06L-4ffe5f30	JPEG-AI 0.25bpp	Red flags disappear
Fig. C.19	ab5e3676-02L-6c9e20d7	Hyperpr. MS-SSIM 0.25bpp	Changed direction of currogate
Fig. C.20	ad865bfc-02H-eaaaac43	Hyperpr. MSE 0.75bpp	Brake lights turn off
Fig. C.21	b4a7895e-06L-2fe29252	JPEG-AI 0.25bpp	Bench and baby disappear
Fig. C.22	c0efe885-06L-48d97723	JPEG-AI 0.25bpp	Hallucinated antenna
Fig. C.23	e03dec99-06L-36e1eace	JPEG-AI 0.25bpp	Different skin tone

expected time for completing the study. The sheet also contained the declaration of consent for them to sign.

The lab was equipped with one desktop computer per student, all with identical hardware. To keep the stimulus presentation constant, all participants accessed the instrument with the same web browser (Firefox). The images were displayed at 512^2 pixels on 23" flat screens with resolution 1920×1080, resulting in a square image with side length 13.5 cm. Some miscompressions were so small that we had to present crops of dimension 128^2 or 256^2, which we scaled up to 512^2 pixels with nearest neighbor upsampling. This kept the size constant and avoided uncontrolled upsampling by the browser.

C.C Dataset usage

The dataset is available on Zenodo via `https://zenodo.org/uploads/16780952`. A Jupyter notebook with download instructions and sample images is available too. All annotations have unique identifiers, starting with the eight leading characters of the image's file name, followed by the model identifier and a sha-1 hash of the annotation's coordinates (x,y,w,h) in the reconstruction image. The images are structured by the compression codecs and contain the compressed and reconstructed files. Note that CDC and STF did not output compressed files.

C.D Statistical analysis

C.E Samples

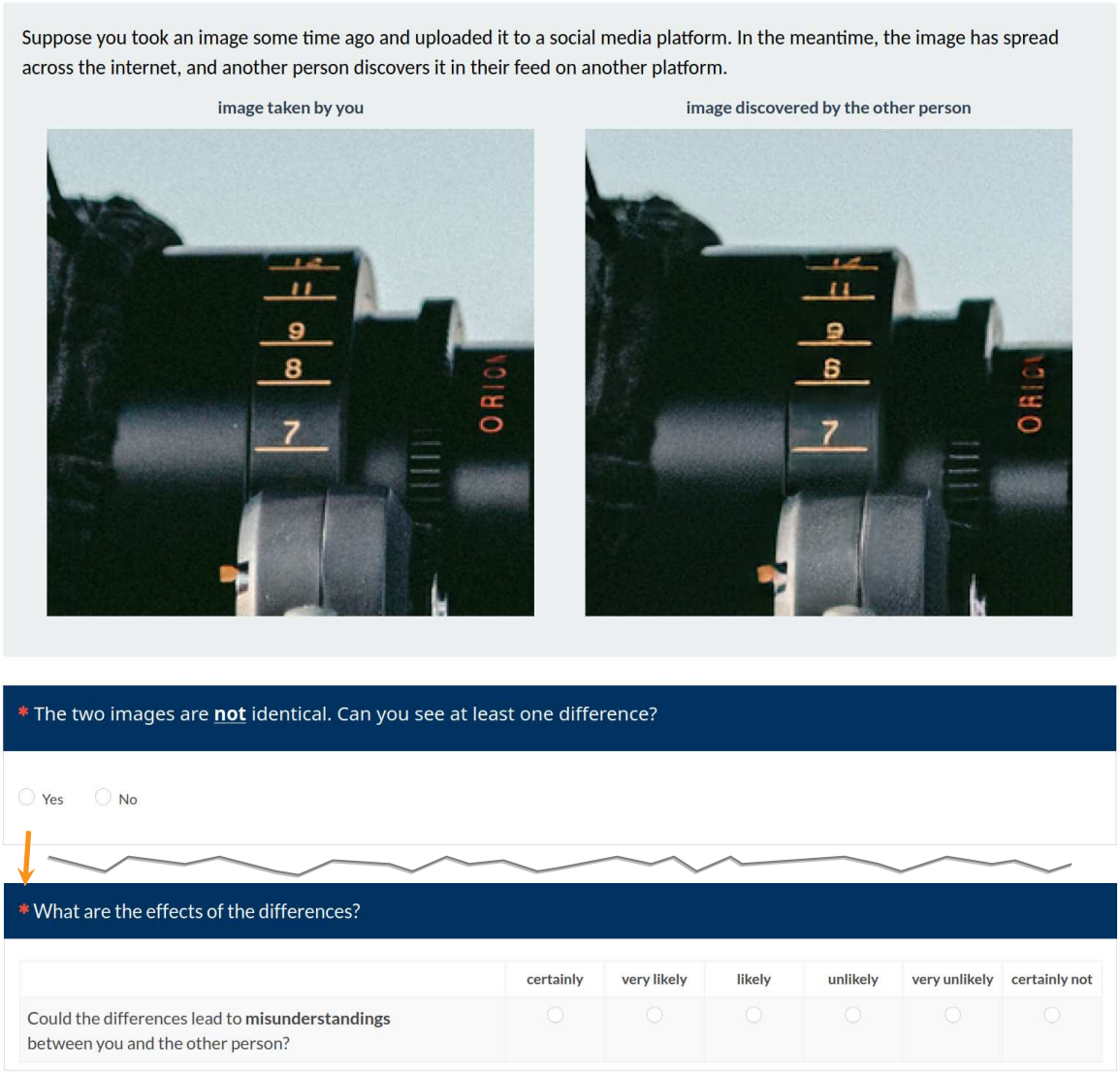


Figure C.4: Screenshot of our validation study. The stimulus images stayed on top of both question blocks. In this example, the digit "8" turns into a "6." The second block was skipped if the first question was answered with "no."

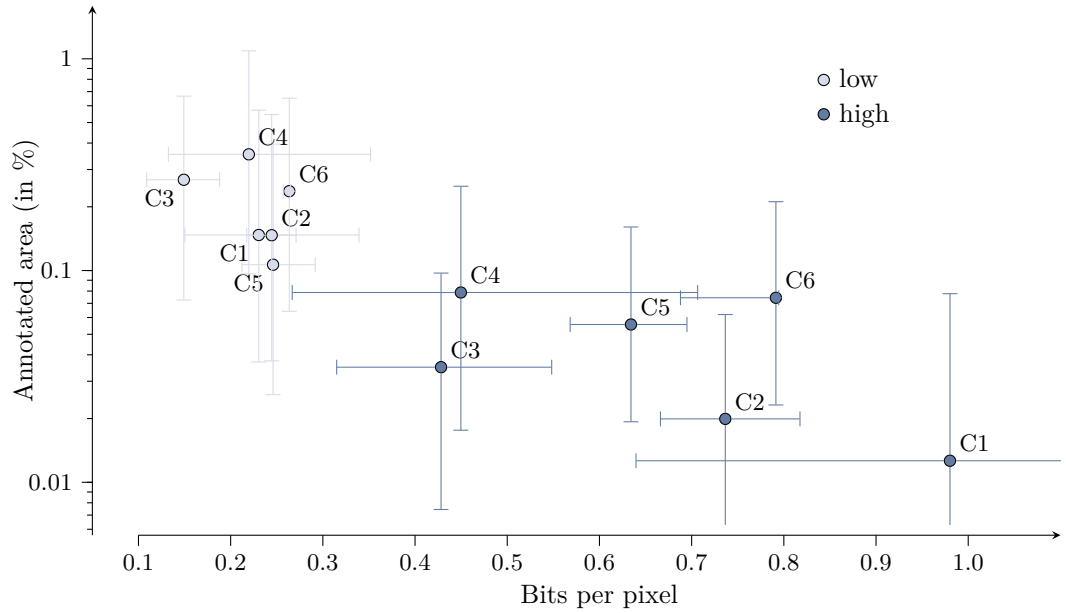


Figure C.5: Median and interquartile ranges of the proportion of pixels per image annotated as miscompressed by codec and compression rate. Lower is better. Note the log scale.

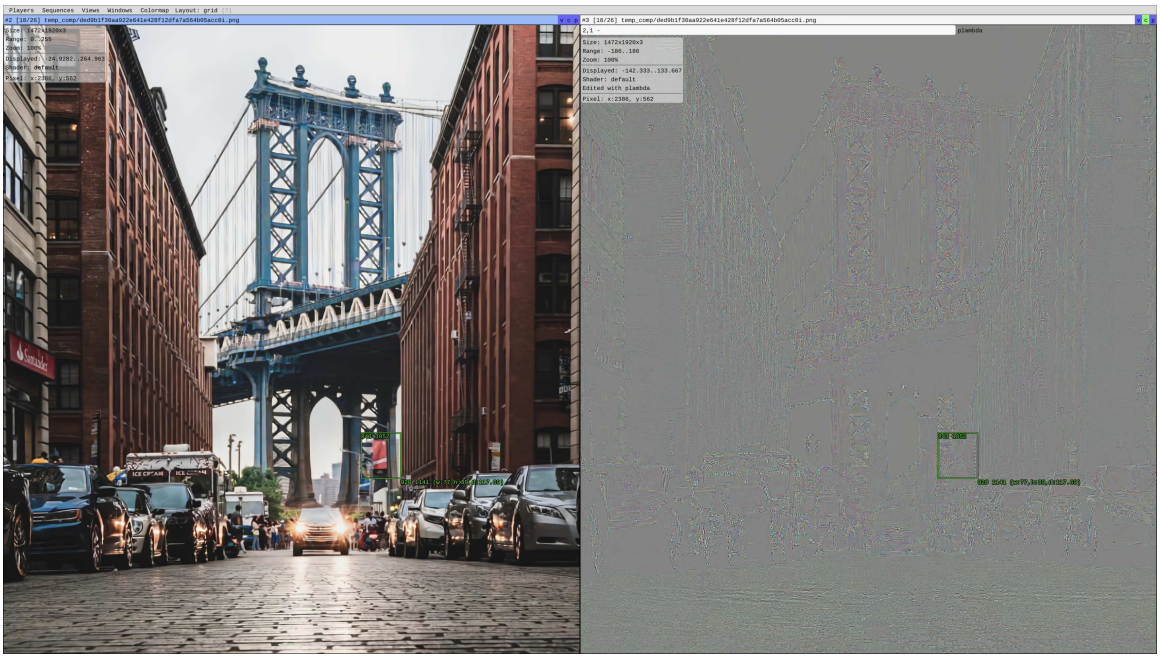


Figure C.6: Screenshot of the labeling setup with VPV [4] modified to allow tagging of miscompressed areas. Raters could toggle between the compressed and uncompressed version of the image on the left and view the difference image on the right. Zooming in was possible. The difference image could be hidden.

Figure C.7: `4154727c-6L-4ffe5f30` Instance of a multimiscompression: all red flags disappear. Labelers were instructed to annotate three instances and take a note in the respective batch file.

Figure C.8: In–depth training image I

Figure C.9: In–depth training image II

Figure C.10: Visualization of our labeler agreement measurement with 4 units per image. Each labeler is assigned one of the RGB colors. Grayscale units contain no annotations, the RGB unit contains annotations of all three labelers, and the yellow unit contains annotations from the two labeler assigned to R and B.

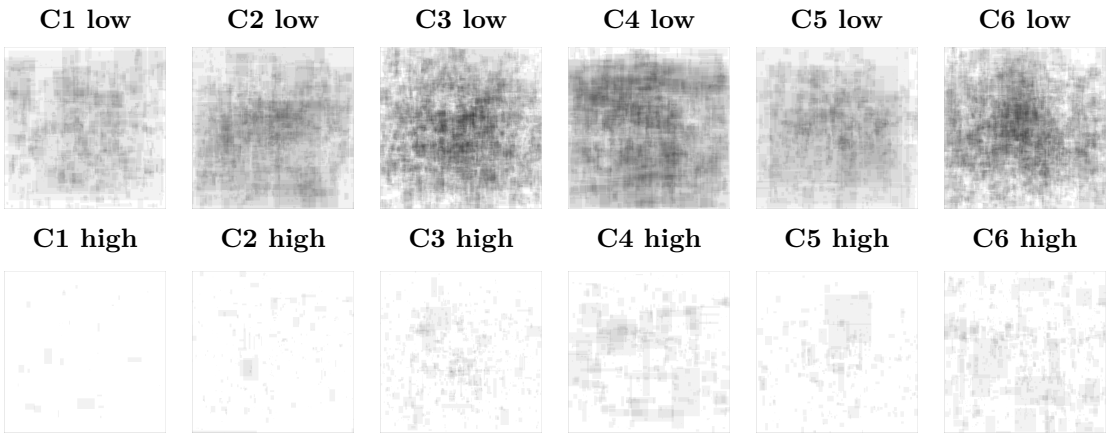


Figure C.11: Visualization of the annotated areas superimposed from all images for each codec. The coordinates for images with varying aspect ratios were scaled to unit squares for this visualization.

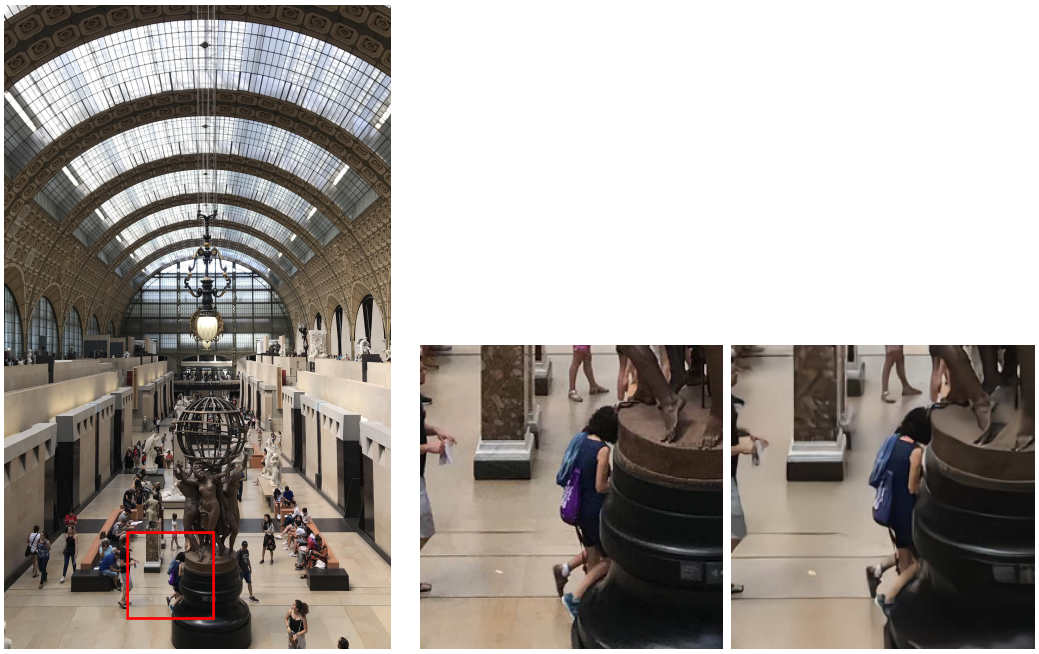


Figure C.12: Example of a miscompression: a purple bag (middle) turns blue (right).

Figure C.13: `01f0f60f-03L-38b9cdda`.

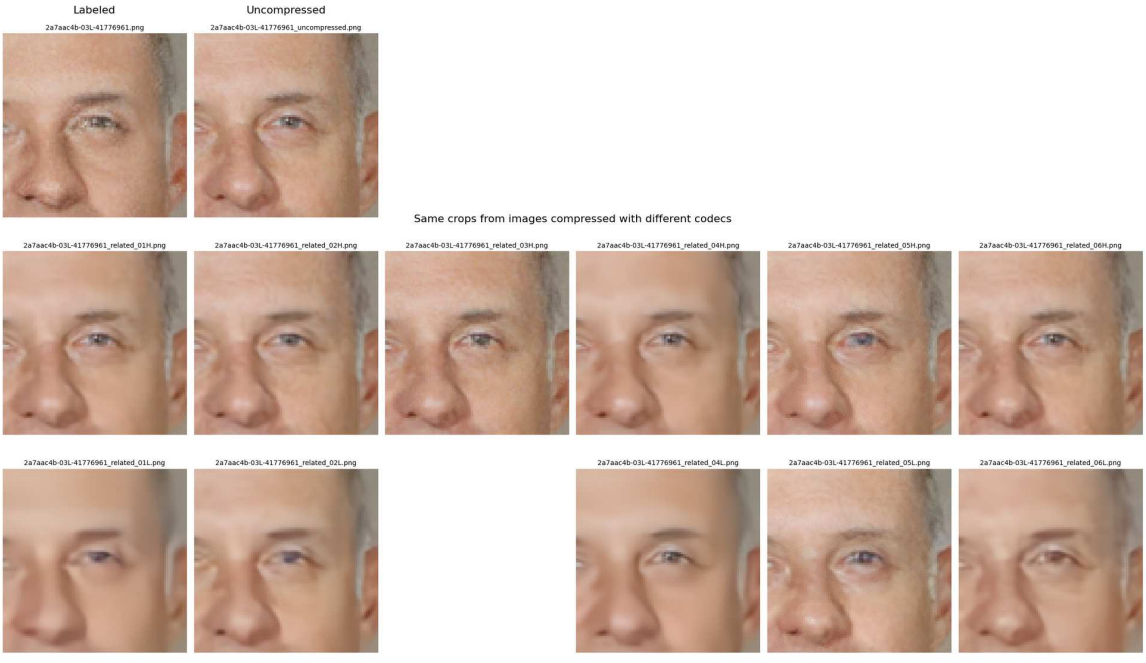


Figure C.14: `2a7aac4b-03L-41776961`.

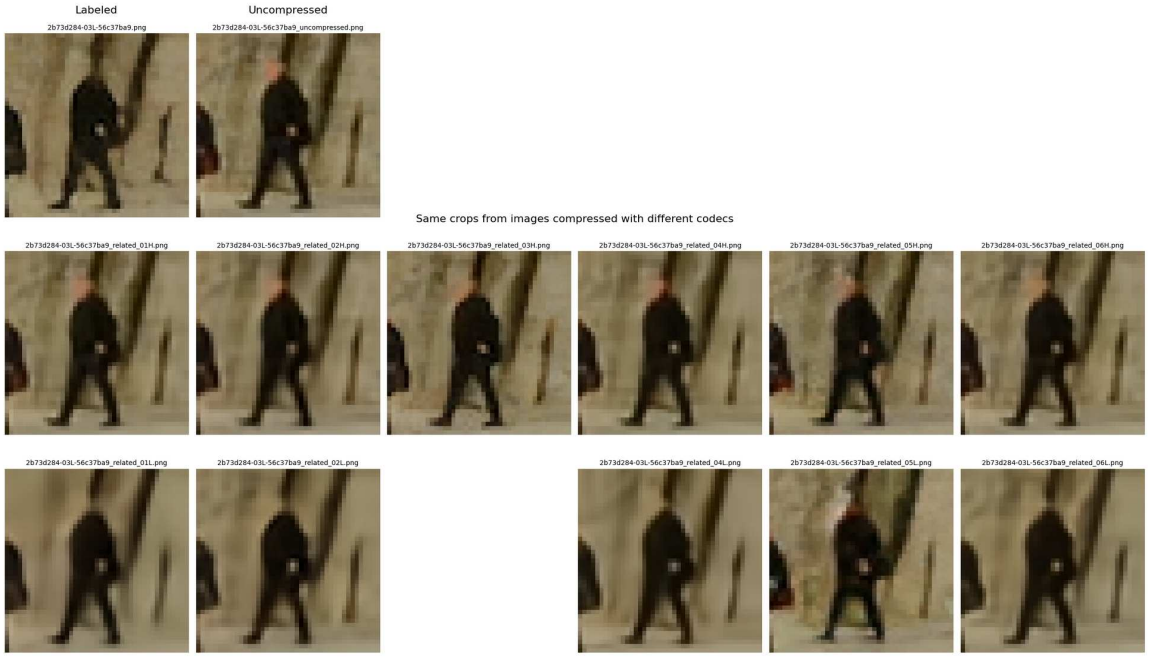


Figure C.15: `2b73d284-03L-56c37ba9`.

Figure C.16: `328f0a7f-05L-6e11a50c`.

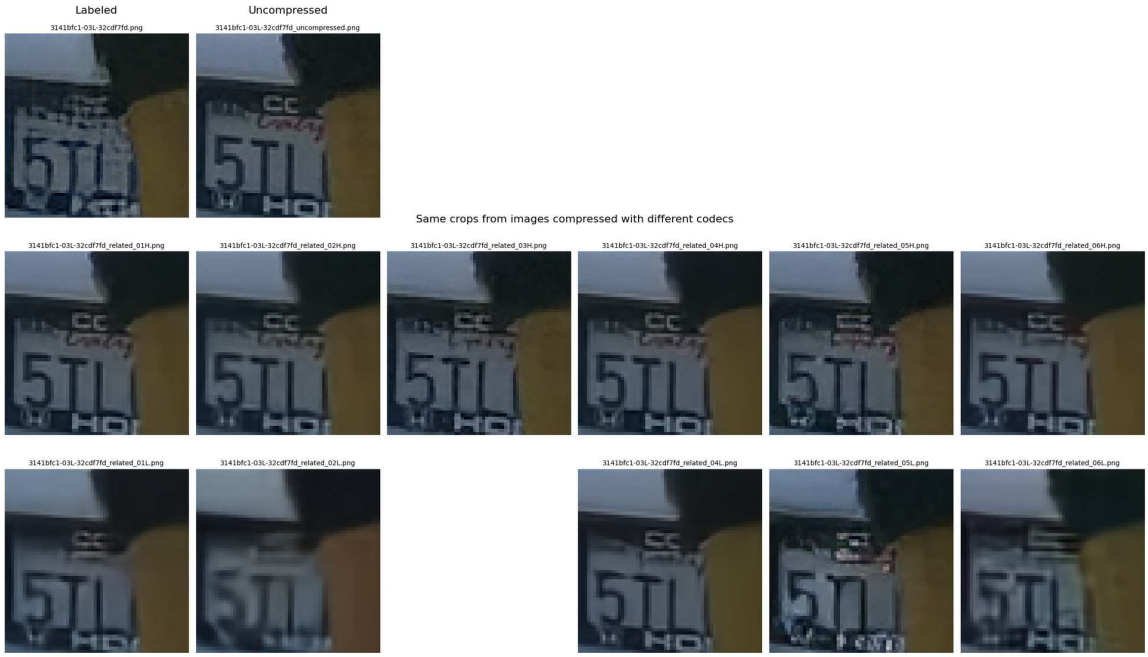


Figure C.17: `3141bfc1-03L-32cdf7fd`.

Figure C.18: `4154727c-06L-4ffe5f30`.

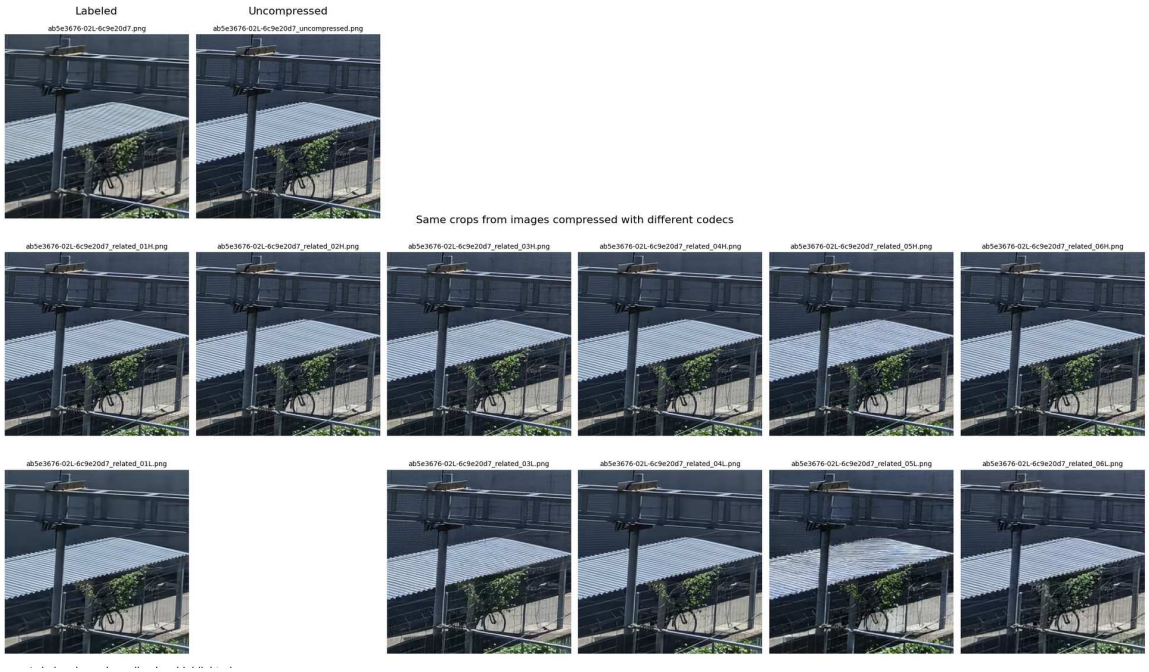


Figure C.19: `ab5e3676-02L-6c9e20d7`.

Figure C.20: `ad865bfc-02H-eaaaac43`.

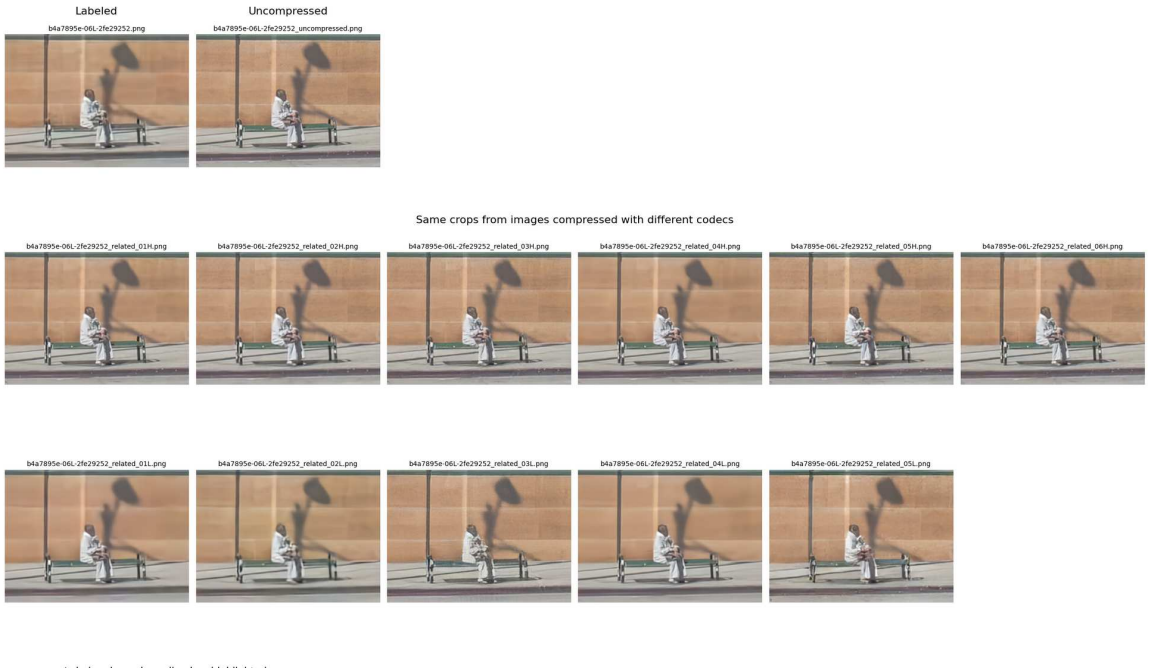


Figure C.21: `b4a7895e-06L-2fe29252`.

Figure C.22: `c0efe885-06L-48d97723`.

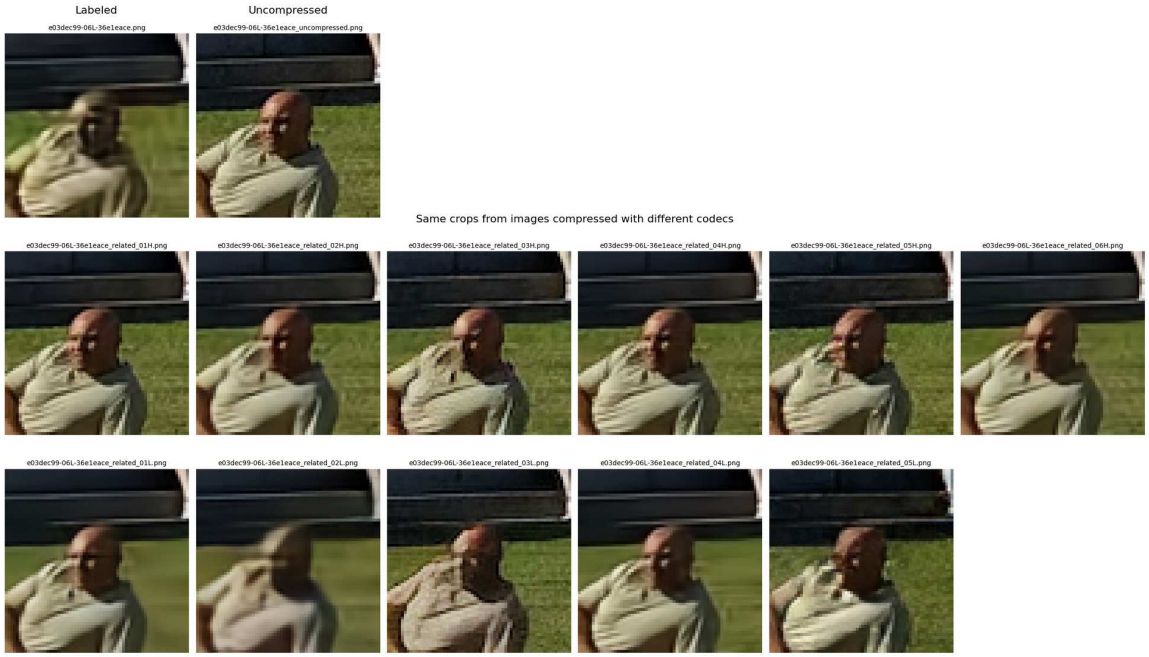


Figure C.23: `e03dec99-06L-36e1eace`. Prior work has analyzed racial bias of neural image compression [27].

D. Understanding *Mozjpeg*

Authors

Nora Hofer, University of Innsbruck
Rainer Böhme, University of Innsbruck

Title

Progressive JPEGs in the Wild: Implications for Information Hiding and Forensics

Conference

ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '23)
Chicago IL, USA · June, 28–30, 2023

Abstract

JPEG images stored in progressive mode have become more prevalent recently. An estimated 30% of all JPEG images on the most popular websites use progressive mode. Presumably, this surge is caused by the adoption of *MozJPEG*, an open-source library designed for web publishers. So far, the optimizations used by *MozJPEG* have not been considered by the multimedia security community, although they are highly relevant. The goal of this paper is to document these optimizations and make them accessible to the research community. Most notably, we find that Trellis optimization in *MozJPEG* modifies quantized DCT coefficients in order to improve the rate–distortion tradeoff using a perceptual model based on PSNR-HVS. This may compromise the reliability of known methods in steganography, steganalysis, and image forensics when dealing with images compressed with *MozJPEG*. We also find that the type and order of scans in progressive mode, which *MozJPEG* adjusts to the image, offer novel cues that can aid forensic source identification.

D.1 Introduction

JPEG is a popular standard for the compression and decompression of digital images. Introduced in 1991, it is now supported by countless applications [20] and more than 75% of all websites including digital images use JPEG [37]. JPEG aims at removing imperceptible information, and hence reducing the file size, while preserving the perceptual quality of an image. The JPEG standard [38] defines different modes, including the "baseline" sequential mode and the progressive mode. While the sequential mode encodes images as a whole, the progressive mode partitions image data into several scans, allowing decoders to display a low-quality version of an image even before all image data is received, *e.g.*, via a slow communication link. The image quality is then gradually improved as more scans are received and decoded.

Although the progressive mode has been part of the standard from the very start, many applications do not use it by default, or not at all. Reports of bugs in browsers when displaying progressive JPEGs have led to recommendations against their use [10]. As a result, the multimedia security research community has barely studied the specifics of progressive JPEGs.

While ignoring the progressive mode was perhaps justifiable in the 1990s and 2000s, when the community was formed, the reality has changed in recent years. Figure D.1 reports results of an ad-hoc crawl of roughly 200.000 images from two sources of historical JPEGs on the web: the image sharing platform Flickr.com, used by amateur and professional photographers; and a sample of images from the 2022 Tranco top-5000 websites [26] archived in the Internet Archive's Wayback Machine [32]. While less than 2% of the images on Flickr were progressive in the 2000s, this share more than tripled after 2020. The Internet Archive sample exhibits a similar growth, however from a higher baseline. Today, about one in three JPEG images on the web is progressive.

The increase of progressive JPEGs found after 2014 can likely be explained with the release of *MozJPEG*, an open-source library, which outputs progressive mode by default. Its declared target group are web publishers. Therefore, the library is tuned specifically for compression with the aim to improve end user experience. Shorter loading times of websites with many images and allowing modern browsers to progressively decode both contribute to this objective. Demand for these features go hand in hand with the evolution of web protocols, like SPDY/HTTP2 [17] and QUIC/HTTP3 [5, 22]. Both successors of HTTP enable connection multiplexing, which allows servers to interleave scans of different progressive images and thus better control the rendering of complex websites. As a result, large social networking platforms quickly adopted progressive JPEG,

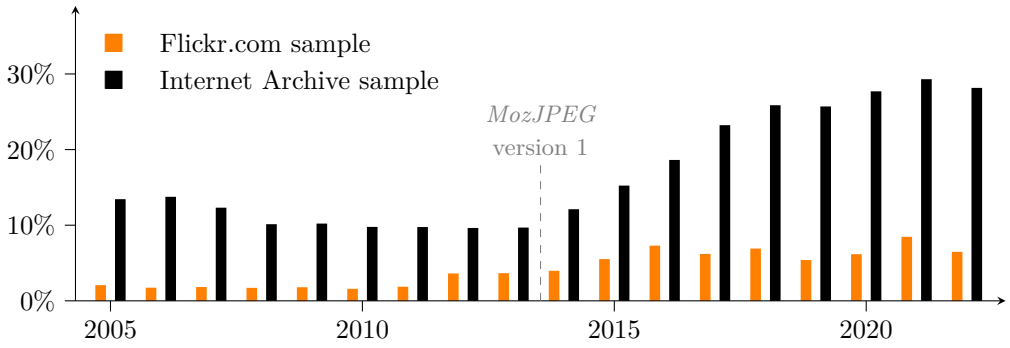


Figure D.1: Prevalence of progressive JPEG images on the web. The bars show the share of progressive mode among all sampled JPEGs per year and data source. Note the increase after *MozJPEG* was released.

as documented in blog posts by Instagram [14] in 2015, Yelp [1] in 2017, Facebook [2] in 2018, and Twitter [36] in 2023, among others. Moreover, popular messenger services such as WhatsApp and Telegram use progressive JPEG. It is also the default mode for libraries like Fresco [15] that help developers to deal with images on Android.
While exploring the reasons behind the adoption of *MozJPEG* might be interesting, in this work we focus on its consequences. Since many methods in multimedia security rely on subtle traces in the signal originating from compression and decompression operations, it is important to understand optimizations implemented in popular implementations. Researchers tended to assume that their methods generalize to progressive mode, arguing that it merely changes the order of encoding in the JPEG file, but does not modify the signal itself. However, *MozJPEG* invalidates this assumption. In a nutshell: our community should not continue to ignore progressive mode images because they are relevant in practice.
In this paper we address this matter and

- analyze the internals of *MozJPEG*, specifically its default Trellis optimization, which changes DCT coefficients in order to find a rate–distortion tradeoff;
- document characteristic traces in the DCT domain that Trellis optimization leaves behind in output images;
- describe how progressive-mode scan scripts interact with the compression pipeline, and
- discuss the implications for steganalysis, steganography, image forensics, and watermarking.

The remainder of this paper is organized as follows. Section D.2 recalls key concepts of JPEG compression, with emphasis on the bit stream encoding and the progressive mode. Section D.3 explains how *MozJPEG* differs from commonly known image compression libraries. It details Trellis optimization, the perceptual model, and scan optimization. Section D.4 presents the results of our experiments on the effects of *MozJPEG* on image data. Section D.5 discusses implications for our research community, before Section D.6 concludes.

D.2 Background

We recall JPEG compression with special emphasis on the bit stream encoding as this is relevant for the rate–distortion optimization. We then expand on the progressive mode before reviewing popular implementations.

D.2.1 JPEG in a Nutshell

An input image in spatial domain representation is converted from the RGB to the $YCbCr$ color space. This process separates the luminance channel Y from the chrominance channels Cb and Cr. As the human eye is less sensitive to changes in brightness than to changes in color, the chrominance channels can be sub-sampled to increase the compression ratio. Typical 4:2:0 subsampling halves each dimension of the chroma channels, *i.e.*, keeping one quarter of the information. All channels are divided into blocks of 8×8 pixels. Each block is transformed to the frequency domain using the Discrete Cosine Transform (DCT). The resulting coefficients are divided by subband-specific quantization factors before rounding to the nearest integer. The quantization factors are derived from an adjustable quality factor (QF), which is commonly chosen between 75 and 100. Lower QFs imply larger quantization factors, which in turn result in smaller quantized coefficient values and more zeros.

Table D.1: Variable-length encoding of coefficient values

Size	Coefficient values		Bit sequence	
0	0		–	
1	**−1**	**1**	0	1
2	**−3**, −2	2, **3**	00, 01	10, 11
3	**−7**,..., −4	4,..., **7**	000,..., 011	100,..., 111
4	**−15**,..., −8	8,..., **15**	0000,..., 0111	1000,..., 1111
5	**−31**,..., −16	16,..., **31**	00000,..., 01111	10000,..., 11111
⋮			...	

Table adapted from [38]; candidates in boldface (see Sec. D.3.3).

JPEG defines a special source coder which combines Huffman encoding with run-length encoding of zeros. High-frequency coefficients are often quantized to zero, therefore a zigzag arrangement yields longer sequences of consecutive zeros, also called *zero runs*. The DC coefficient is treated separately and not detailed here for brevity. Interestingly, the actual values of non-zero coefficients are not subject to Huffman encoding. Instead, the standard defines a variable-length encoding scheme, shown in Table D.1. The actual stream is composed of alternating *control bytes* and *variable-length coefficient values*. The control bytes combine the size tag with an optional number of zeros (NZ) preceding the coefficient. Table D.2 illustrates the control bytes, also indicating special symbols, such as end-of-block (EOB) and zero run length (ZRL). Only the control bytes are Huffman-encoded using tables stored in the file. Table D.2 is annotated with the number of bits required to store each displayed control byte using an example Huffman table from the Y channel of an image compressed with QF 75. For example, the AC coefficient sequence $(3, 0, 0, 8, 0, 4)$ is encoded as follows:

$$\overbrace{100}^{(0,2)}\underbrace{11}_{3},\overbrace{111\,1110\,0100}^{(2,4)},\underbrace{1000}_{8},\overbrace{11\,0111}^{(1,3)}\underbrace{100}_{4}\quad,$$

where Huffman-encoded control bytes are annotated above and variable-length coefficient values are annotated below the bit stream.

D.2.2 Progressive Mode

While baseline JPEG uses the sequential mode, the standard also defines a progressive mode [38]. It partitions the information before lossless encoding into several scans. This enables to store all necessary data for a low-quality version of the complete image in the first scan, *i.e.*, at the beginning of the file. Subsequent scans refine the transmitted version of the image until the full image quality is reached. In situations where a JPEG file is transmitted over a slow communication link, a decoder can quickly produce a low-quality image and then gradually improve the displayed quality as more scans are received [21]. After all scans are complete, the final image is identical to that of a sequential JPEG file compressed with the same settings.

The partitioning of image data is specified in the *scan script*. This script can combine *spectral selection*, where lower-frequency subbands are transmitted before higher-frequency subbands, with *successive approximation*, where bits of lower significance are omitted initially and supplied in later scans [25]. Figure D.2 illustrates a typical scan script of a grayscale image. The DCT-transformed data is arranged as a cube where the axes represent the subbands (in zig-zag order), the coefficient

Table D.2: Control bytes encoding the size of the coefficient value in bits and the preceding number of zeros (NZ). The smaller numbers in parentheses show the bit length of the control bytes for a specific Huffman table example. (Table adapted from [38].)

	Bits to store coefficient value								
NZ	**0**	**1**	**2**	**3**	**4**	**5**	**...**	**14**	**15**
0	EOB (3)	01 (2)	02 (3)	03 (4)	04 (5)	05 (6)	...	14 (0)	15 (0)
1	-	17 (3)	18 (5)	19 (6)	20 (8)	21 (9)	...	30 (0)	31 (0)
2	-	33 (5)	34 (7)	35 (10)	36 (11)	37 (12)	...	46 (0)	47 (0)
3	-	49 (5)	50 (8)	51 (11)	52 (13)	53 (14)	...	62 (0)	63 (0)
4	-	65 (5)	66 (8)	67 (10)	68 (11)	69 (14)	...	78 (0)	79 (0)
5	-	81 (6)	82 (9)	83 (12)	84 (13)	85 (14)	...	94 (0)	95 (0)
⋮									
14	-	225 (11)	226 (14)	227 (14)	228 (14)	229 (14)	...	238 (0)	239 (0)
15	ZRL (11)	241 (13)	242 (14)	243 (14)	244 (14)	245 (14)	...	254 (0)	255 (0)

bits (from MSB to LSB), and the block index. Spectral selection slices the cube in rows, whereas successive approximations cut the cube into columns.

Figure D.3 shows the visual effect of progressive decoding of a 768×128 color image compressed with QF 99, default chroma subsampling 4:2:0 and the scan script shown in Figure D.4. The image data is partitioned into a total of nine scans using both spectral selection and successive approximation. From left to right, each part of the figure shows an increasing number of scans. We combine corresponding scans for both chroma channels, although they must be stored in separate scans in the file. The initial Scan 1 contains the DC coefficient band without LSB, resulting in an image of 256 blocks representing the block average color. Scan 2 contains the first five low-frequency AC coefficient bands of the luminance channel, excluding the LSB. Scans 3 and 4 contain all AC coefficient bands of both chrominance channels, again excluding the LSB. Scan 5 contains all remaining AC coefficient bands of the luminance channel, again excluding the LSB. At this point, all but the least significant bits of all coefficients are sent. Scan 6 transmits the LSBs of all channels of the DC coefficient bands, and scans 7 to 9 transmit the LSBs of the AC coefficient bands of all channels. Observe from the cumulative shares of DCT coefficients and compressed file size that the information in the first scans is compressed at a lower rate than the refinements in the later scans. The reason for this might be that later scans contain mainly zeros and ones, which can be compressed very efficiently with tailored Huffman tables.

So far, the progressive mode has not been in the center of attention in the research area of multimedia security. We are aware of [35], which proposes selective image encryption specifically for scans in progressive mode. Another innovative use is described in [29]. The authors observe that the set of custom Huffman tables of progressive JPEG images increase the entropy of the file header, allowing to uniquely identify images from header information only.

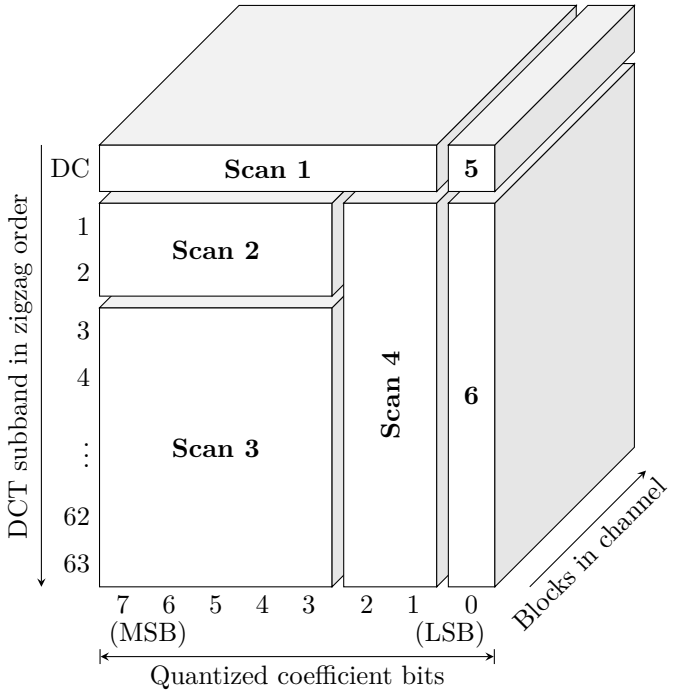


Figure D.2: Partitioning of a grayscale image into six scans according to a scan script. Illustration adapted from [38].

D.2.3 Popular Implementations

Many software packages build on the open-source C library *libjpeg*. *libjpeg* has been developed by the Independent JPEG Group [25] since 1991. In 2010, *libjpeg–turbo* was created as a fork of *libjpeg* with the aim of improving the decompression and compression performance by using optimized platform-specific SIMD instructions [28]. Both libraries support the progressive mode, although not as their default.

In 2014, Mozilla forked *libjpeg–turbo* into *MozJPEG* [30] to optimize it for a different objective. *MozJPEG* aims to achieve higher compression rates at the same perceived quality, effectively reducing the loading times of images on the web. To do so, it implements a rate–distortion optimization inspired from trellis quantization [11, 39], it uses progressive mode by default, and selects scan scripts based on the image content.

1.37%	5.93%	34.64%	87.50%	87.70%	100%
4.17%	**13.34%**	**25.05%**	**73.14%**	**73.68%**	**100%**

Figure D.3: The visual effect of progressive decoding. Percentages show the cumulative share of DCT data (upper row) and compressed file size (lower row) at different steps of decoding.

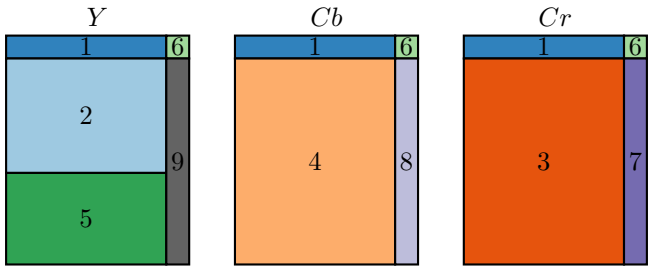


Figure D.4: The scan script used to generate Figure D.3.

All three libraries have a common interface and can be used widely interchangeably. The authors of [4] compare different JPEG implementations, including different versions of *libjpeg*, and point out implications for multimedia security. However, due to the different default compression mode, they do not compare to *MozJPEG*. The present work seeks to close this gap.

D.3 Understanding *MozJPEG*

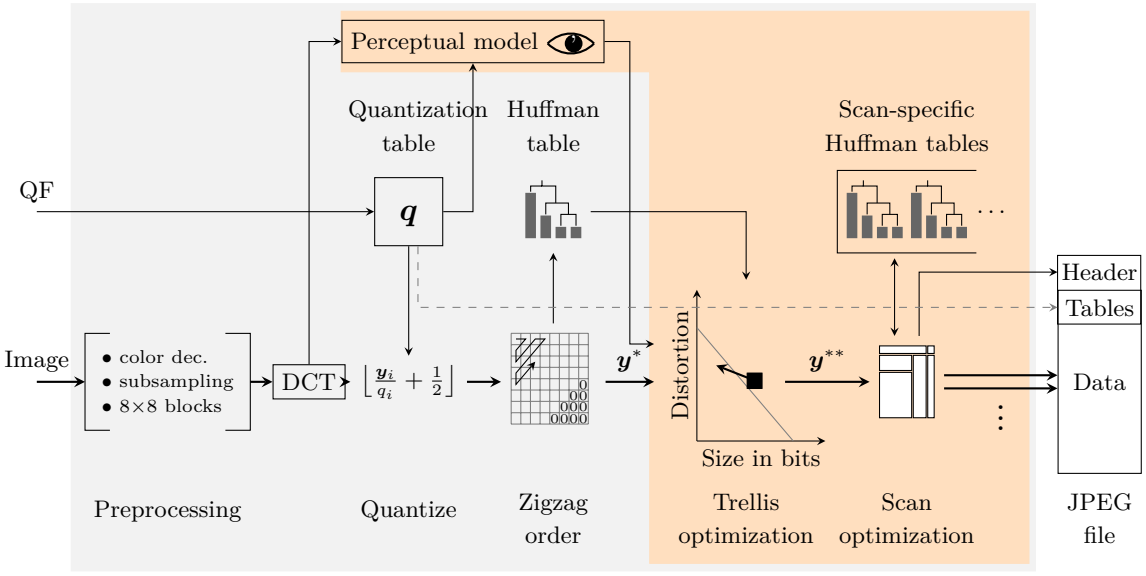


Figure D.5: Compression pipeline for progressive JPEG. The orange parts are specific to default *MozJPEG*.

To the best of our knowledge, the multimedia security community has not devoted much attention to progressive JPEG.[1]
This can be justified by the observation that progressive and sequential modes transmit the very same information on the level of quantized DCT coefficients. Hence, research on steganography, steganalysis, watermarking, and forensics dealing with the signal itself should be unaffected. However, the adoption of *MozJPEG* thwarts this rationale: its Trellis optimization *does* modify the DCT coefficients.

[1] An exception is [7] who take into account the changes in DCT coefficients introduced by the rate-distortion optimization of *MozJPEG* while proposing a method for robust steganography.

Table D.3: List of symbols

y_i	*un*quantized DCT coefficient value of the i-th subband
y_i^*	quantized DCT coefficient value *before* Trellis
y_i^{**}	quantized DCT coefficient value *after* Trellis
$\boldsymbol{q}$	quantization matrix
q_i	quantization factor of the i-th subband
$\mathcal{C}$	set of suitable candidate values (cf. Table D.1)
$\mathcal{C}_i$	set of candidates for a given y_i^*
$c_{i,k}$	elements of $\mathcal{C}_i$
r	run (number of zeros)
n	coefficients in a block, i. e., length of the trellis
$S(y^*, r)$	size (in bits) of a sequence of r zeros followed by the non-zero coefficient y^* after Huffman encoding
$D_i(y^*, y)$	(additive) distortion when the quantized coefficient y^* represents the unquantized value y in subband i
λ	rate–distortion parameter (depends on QF)
κ	cost (sum of size in bits and distortion)

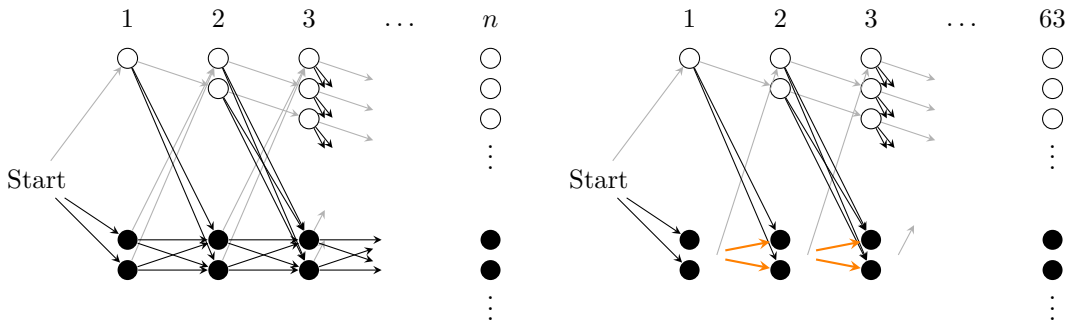
In this section, we explain *MozJPEG*'s modifications to the typical JPEG compression pipeline, thereby commenting on the realized savings for images of varying size and QF. We then explain the perceptual model, the Trellis optimization, and the scan optimization in separate subsections. There are four major versions of *MozJPEG*. The analysis in this paper refers to the latest version 4.1.1 released in August 2022.

D.3.1 *MozJPEG*'s Compression Pipeline

Figure D.5 shows a block diagram of *MozJPEG*'s image compression pipeline. The signal path is located in the bottom. Parts where *MozJPEG* innovates compared to other implementations are highlighted in orange. For the gray parts, we refer the reader to the description in Section D.2.1. Our convention on the formal notation is summarized in Table D.3.

The heart of *MozJPEG* is the rate–distortion optimization. It uses a perceptual model to calculate the distortion implied by reducing non-zero DCT coefficients to values with shorter bit size or even zeroing them out in order to increase the length of zero runs. The distortion is scaled to be comparable to storage bits. The algorithm tries to move each block in each channel independently leftwards in the size-distortion space, illustrated in Figure D.5. Small upwards movements are tolerated, as indicated by the indifference line. The estimated size in bits is calculated using a Huffman table specific to the distribution of quantized DCT coefficients in the given image.

Note that the Trellis optimization is not performed on a scan level but under the assumption of sequential mode. Only the resulting quantized DCT coefficients are passed to the scan optimization. This step jointly optimizes a scan script and a set of corresponding Huffman tables, which go into the output file. If both optimizations are enabled (as by default), the estimated sizes used for Trellis optimization do not necessarily correspond to the actual sizes. This is because the subsequent scan optimization almost certainly generates Huffman tables that encode control bytes with fewer bits. This suggests that repeated optimization passes may exhibit similar convergence behavior as reported for repeated quantization [8, 24].



(a) Generic, adapted from [39]. **(b)** Simplified, implemented in *MozJPEG*.

Figure D.6: Comparing *MozJPEG*'s Trellis optimization (right) to the general case (left). In *MozJPEG*, the cost function is additive for non-zero AC DCT coefficients. This curbs the path explosion by limiting the path dependencies to zero runs.

D.3.2 Perceptual Model

The purpose of the perceptual model is to estimate the perceived distortion for alternative (i. e., smaller) coefficient values. *MozJPEG* implements the variant of PSNR-HVS described in [13]. This metric, denoted $D_i(y^*, y)$, is based on the square error between the unquantized coefficient y_i and the dequantized coefficient $q_i \cdot y_i^*$, where q_i is the quantization factor in subband i (cf. Table D.3). The implied assumption is that the quantization table is a good approximation of the human sensitivity to changes in subbands [13].
An advantage of this model is that the distortion is additive in the DCT domain, unlike many earlier models of the human visual system [31]. This property considerably simplifies the search for a rate–distortion tradeoff as calculating the distortion does not involve transformations between domains and the coefficients can be selected independently with regards to distortion.

D.3.3 Trellis Optimization

Trellis optimization in general solves the rate–distortion problem by finding the path through a trellis structure that minimizes a cost function [39]. The Viterbi algorithm can find a solution with modest complexity even if the size in bits is variable and non-monotonic in the coefficient value, and the distortion is non-additive across subbands. This way, a coefficient could be rounded upwards with small impact on the size in order to compensate a larger downward rounding of another coefficient, leading to a net reduction in size and a better rate–distortion tradeoff. This general case has been proposed for video codecs [39]. *MozJPEG* takes a much simpler approach. Specifically, *MozJPEG* exploits two properties:

1. the additive distortion model (cf. Section D.3.2), and
2. the fact that the variable-size encoding of coefficient values is fixed in the JPEG standard, independent of other coefficients, and monotonic in the absolute value of the coefficient (cf. Table D.1).

Since distortion and size are independent for non-zero coefficients, the cost κ is independent, too. The only remaining dependency between AC coefficients in a block arises in the case of zero runs. Visualizing this observation in a trellis diagram, Figure D.6 shows how the number of paths to consider is limited in *MozJPEG*. Observe that each candidate node in Figure D.6b has exactly one

incoming edge (in orange) from the previous non-zero coefficient, regardless of its value; plus one for each potential run of preceding zeros, implying a quadratic upper bound of the search space. By contrast, the general case has one incoming edge for each possible value of the preceding non-zero coefficient, implying an exponential search space in n.

We now describe *MozJPEG*'s Trellis optimization of zigzag-ordered AC coefficients in a block. The sequence of the difference-encoded DC coefficients of all blocks is optimized similarly, but we omit them here for brevity. Trellis optimization consists of two steps: first it evaluates potential alternative coefficient values with smaller bit sizes and then tries to increase the number and length of zero runs. We denote y_i^* and y_i^{**} as the quantized coefficients *before* and *after* Trellis optimization, respectively. *MozJPEG* constructs a set of possible alternative candidates

$$c_{i,j} \in \mathcal{C}_i = \{\forall c \in \mathcal{C} : |c| < |y_i^*|\} \cup \{y_i^*\}, \tag{D.1}$$

where set $\mathcal{C} = \{\pm(2^i - 1); i = 1, \ldots, 15\}$ contains all numbers with maximum absolute value per size in bits according to the variable-length encoding of coefficient values (cf. Table D.1). These numbers are suitable candidates as they allow to encode a larger original coefficient value with fewer bits while minimizing the distortion.

The size of non-zero coefficients also depends on the number of preceding zeros. *MozJPEG* iterates over all paths with zero runs of varying length r preceding the non-zero coefficient. This causes the quadratic complexity as visible in Fig. D.6b. The cost κ is given by

$$\kappa_{i,k,r} = \underbrace{S(c_{i,k}, r)}_{\text{total size in bits}} + \underbrace{D_i(c_{i,k}, y_i)}_{\text{distortion in subband } i} + \underbrace{\sum_{j=i-r}^{i-1} D_j(0, y_j)}_{\text{distortion of } r \text{ zeros}} . \tag{D.2}$$

Function S returns the size (in bits) of the Huffman-encoded control byte for a sequence of r zeros plus the size of the variable-length coefficient value. Note that more than one control byte may be necessary to encode zero runs larger than 15 (cf. Table D.2). Function D is the distortion model as defined in Section D.3.2.

The final quantized coefficient after Trellis optimization y_i^{**} is set to $c_{i,k}$ with the lowest $\kappa_{i,k,r}$. If $r > 0$, some non-zero coefficients $y_j^{**}, j < i$ may be set to zero to realize the run.

Example Table D.4 shows a numerical example. The first four unquantized AC DCT coefficients y_i are quantized to y_i^* with quantization factors q_i from *MozJPEG*'s quantization table for QF 75. The table shows the path exploration for $i = 4$ (in bold). The set of candidates $\mathcal{C}_i$ includes the original value 8, which has a variable bit size of 4, and all positive numbers with the highest absolute value for each smaller variable bit size, i. e., 7, 3, and 1. The table has one section for the exploration of the paths associated with each candidate. For the original value $c_{4,1} = 8$, the distortion is determined by the quantization error. In our example, the distortion of candidate $c_{4,2} = 7$ is the same, because the quantization error $\|8 \cdot 72 - 540\| = 36$ is exactly half of the quantization factor q_4. Hence, the direction of rounding does not matter in this case. Since 7 can be encoded in two fewer bits — one from the variable-size encoding and the other one from the Huffman table of the control byte — the resulting cost of 35.89 is exactly two "bits" smaller than for the original value. In this example, this costs is the minimum of all rows, hence $y_4^{**} = 7$ will be the result of this optimization step. Note that in order to get the minimum costs, the algorithm needs to calculate them of all other rows shown, which involves the exploration of three additional paths per candidate for potential zero runs. For this purpose, the size column shows the accumulated size of all coefficients up to i. The costs of the example coefficients are prohibitively high. This is different for higher frequency AC coefficients, which tend to be of smaller absolute value. We have chosen $i = 4$ to fit the table in a column.

Table D.4: Example of a Trellis iteration for a coefficient with three candidates and three preceding non-zero AC coefficients. The smaller numbers in parentheses show the bit lengths of the Huffman-encoded control byte plus the size of the variable-length coefficient value.

	AC coefficients				Size	Distortion	Cost
	zigzag order →						
i	1	2	3	4			
y_i	−574	−635	−107	540			
q_i	64	64	64	72			
y_i^*	−9	−10	−2	**8**			
Candidate $c_{4,1} = 8$:							
	−9 (5+4)	−10 (5+4)	−2 (3+2)	**8** (5+4)	32	5.89	37.89
	−9 (5+4)	−10 (5+4)	0	**8** (8+4)	30	49.20	79.20
	−9 (5+4)	0	0	**8** (11+4)	24	1639.87	1663.87
	0	0	0	**8** (13+4)	17	2939.60	2956.60
Candidate $c_{4,2} = 7$:							
	−9 (5+4)	−10 (5+4)	−2 (3+2)	**7** (4+3)	30	5.89	35.89
	−9 (5+4)	−10 (5+4)	0	**7** (6+3)	27	49.33	76.33
	−9 (5+4)	0	0	**7** (10+3)	22	1639.88	1661.88
	0	0	0	**7** (11+3)	14	2939.60	2953.60
Candidate $c_{4,3} = 3$:							
	−9 (5+4)	−10 (5+4)	−2 (3+2)	**3** (3+2)	28	329.06	357.06
	−9 (5+4)	−10 (5+4)	0	**3** (5+2)	25	372.49	397.49
	−9 (5+4)	0	0	**3** (7+2)	18	1963.04	1981.04
	0	0	0	**3** (8+2)	10	3262.77	3272.77
Candidate $c_{4,4} = 1$:							
	−9 (5+4)	−10 (5+4)	−2 (3+2)	**1** (2+1)	26	684.53	710.53
	−9 (5+4)	−10 (5+4)	0	**1** (3+1)	22	727.97	749.97
	−9 (5+4)	0	0	**1** (5+1)	15	2318.52	2333.52
	0	0	0	**1** (5+1)	6	3618.24	3624.24

D.3.4 Scan Optimization

MozJPEG enables scan optimization by default. It reimplements an idea proposed in the form of a code snippet[2] published by Loren Merritt in 2009. Unlike Trellis optimization, scan optimization does not alter the signal. The algorithm searches 23 variations of scans for the luminance channel and 41 for the chrominance channels to compose the scan script with the shortest output size. This involves deriving the optimal Huffman table for each scan. Our experiments with 100 test images from the Alaska dataset [9] resulted in 86 distinct scan scripts for QF 100, 69 for QF 99, and 35 for QF 75. Observe that the user-experience during progressive decoding over slow connections is not an optimization criterion. We conjecture that large websites therefore supply their own scan scripts.

D.4 Effects of *MozJPEG*

This section analyzes the impact of *MozJPEG* on file sizes and DCT coefficients experimentally. We use a random sample of 100 never-compressed color images from the Alaska dataset [9]. The images were acquired with different cameras and cropped to 512×512 pixels by the publishers of the dataset.

D.4.1 Effect on File Size

Figure D.7 shows how different parts of the *MozJPEG* compression pipeline influence the output size. Solid lines are averages over 100 images with 512×512 pixels; dashed lines represent images downscaled to 48×48, the typical size of icons on social media websites. Both sets of images are compressed with the typical 4:2:0 chroma subsampling using three different QFs, 100, 99, and 75. *MozJPEG*'s default setting with progressive mode, Trellis optimization, and scan optimization is shown in the rightmost column. All numbers are scaled to *MozJPEG*'s sequential mode without any optimization as 100%. For comparison, the sequential default in baseline *libjpeg* version 6b is given on the left.

The difference between *libjpeg* and *MozJPEG* sequential is due to *MozJPEG*'s more aggressive quantization matrices and the use of custom Huffman tables.[3] Our measurements for 512×512 are broadly in line with numbers reported by [27]. This source compares compression ratio and performance of default *MozJPEG* to *libjpeg–turbo*. It finds that *MozJPEG* outputs 20% smaller files while taking on average 25% more time to compress. Note that the study compares *MozJPEG* at version 2.1, which does not apply Trellis optimization to DC coefficients. Icon-sized images seem to benefit significantly more from *MozJPEG*, speculatively a reason for its adoption by large websites.

Turning to the compression options of *MozJPEG*, Trellis optimization offers the largest space savings, followed by scan optimization. The use of the progressive mode alone is advantageous for larger images; only in combination with Trellis and scan optimization it does not blow up the file size of icon-sized images. This suggests that *MozJPEG*'s defaults are chosen very reasonably for the variety of images served on the web.

[2] https://github.com/bsandrow/utils/blob/master/jpegrescan

[3] *libjpeg* and *libjpeg–turbo* use the general-purpose Huffman tables defined in the JPEG standard, which are clearly suboptimal for certain scans in progressive mode.

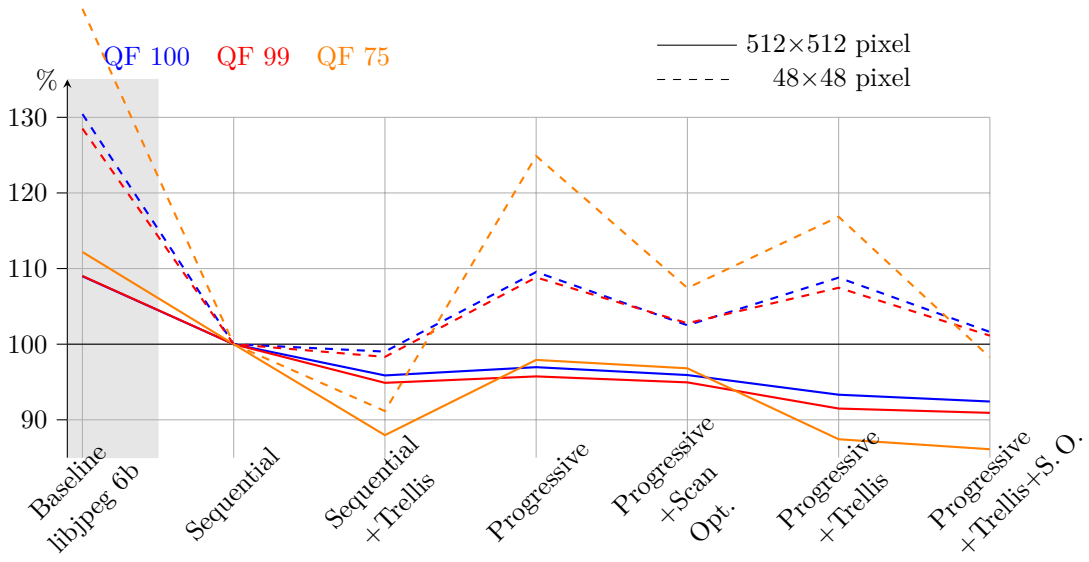


Figure D.7: Effect of compression options on file size. Average over 100 color images. Sequential *MozJPEG* with all optimizations disabled = 100%.

D.4.2 Effect on DCT Coefficients

We measure the effect of Trellis optimization in the frequency domain by comparing quantized AC coefficients of the luminance channel before and after Trellis optimization.
The distribution of DCT coefficients from natural images usually follows a Laplace distribution [34]. Trellis optimization shifts probability mass towards zero, and towards suitable candidate values of shorter variable-length encoding (cf. Table D.1). Therefore, we expect characteristic deviations from this discretized distribution for images compressed with *MozJPEG*.
Figure E.1 illustrates these deviations for coefficients of the first AC subband of 100 sample images. It is visible that candidate coefficients occur more frequently, while upper neighboring coefficients appear less often. Also, the frequency of zeros increases. Figure D.9 shows the same effect for all AC coefficient subbands for different QFs. The same effect is present, albeit less visible for non-

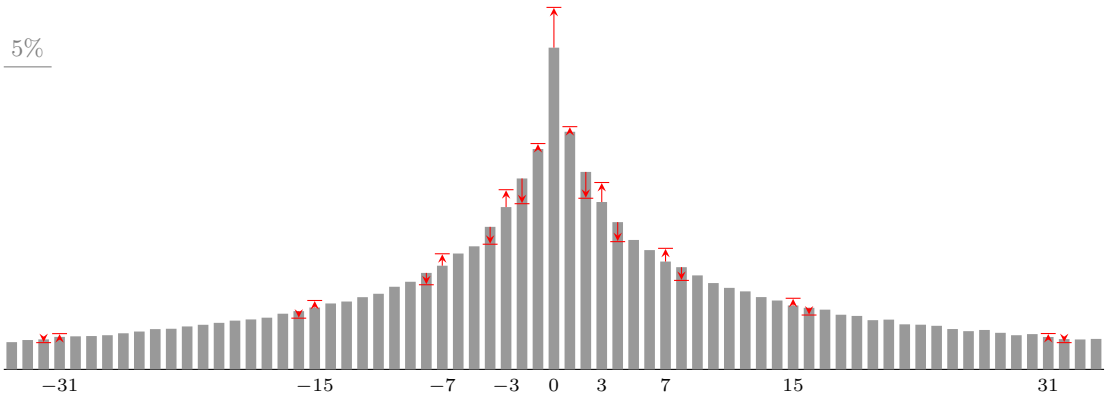
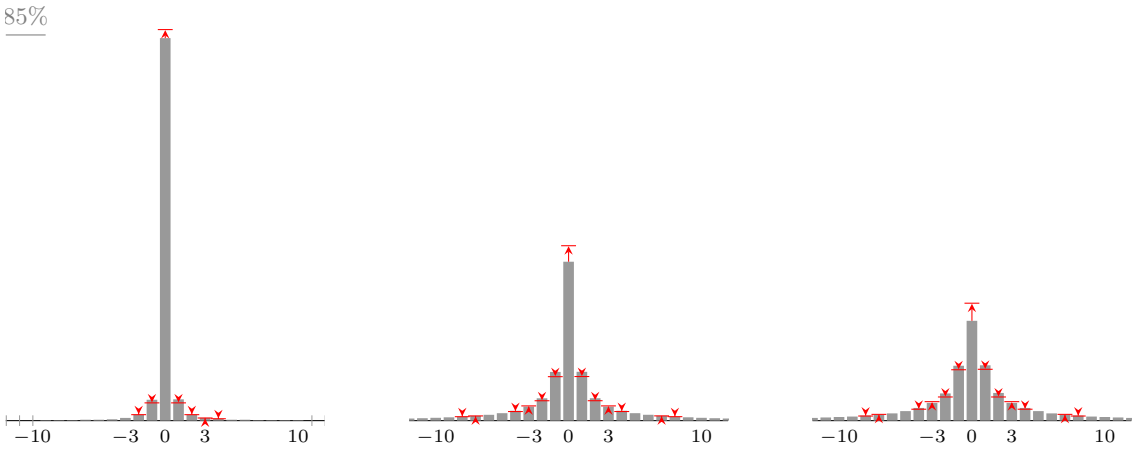
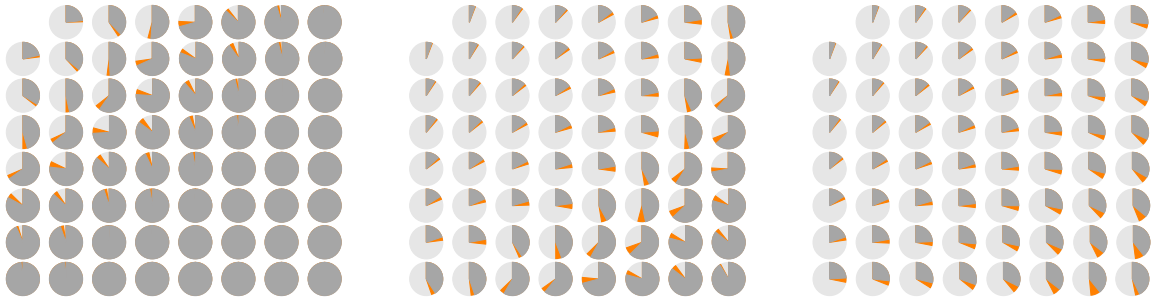


Figure D.8: Histograms of the first DCT AC coefficient before (bar) and after (arrow) Trellis optimization for QF 99.



(a) QF 75 **(b)** QF 99 **(c)** QF 100

Figure D.9: Histograms of all quantized AC DCT coefficients before (bar) and after (arrow) Trellis optimization.



(a) QF 75 **(b)** QF 99 **(c)** QF 100

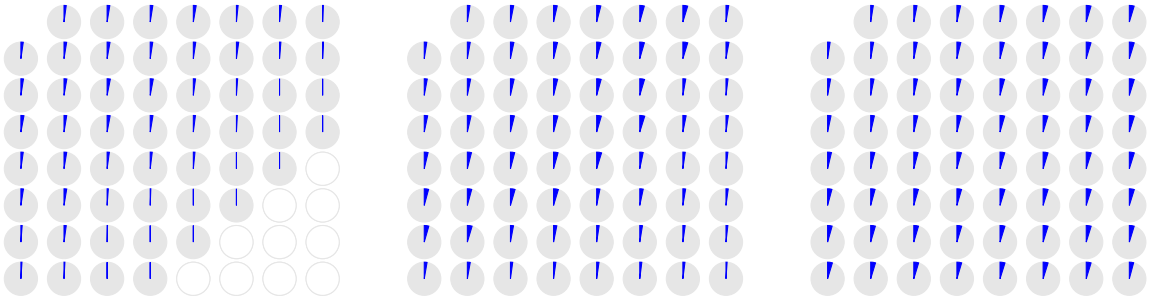
Figure D.10: Share of zeros before (dark) and after (dark + orange) Trellis optimization by DCT subband and JPEG quality factor.

zero values due to the increasing number of coefficients equal and close to zero in high-frequency subbands.

Figure D.10 shows the share of coefficients being changed to zero broken down by DCT subband and QF. The highest increases in zeros are observed in subbands that already have a high share of zeros. This is plausible as preexisting zero runs can be combined or enlarged with modest additional distortion. Future detectors of *MozJPEG* should weigh the subbands used for the decision by this statistic.

Non-zero coefficients are changed at a small, but relatively consistent rate in all subbands except the ones with a very high share of zeros, as shown in Figure D.11. For QF 75, some high-frequency subbands are all zeros; here the ratio is not defined. Figure D.12 extends the analysis and shows that the findings hold true for all QFs from 50 to 100. It shows the average share of introduced changes of all AC subbands as well as the first (1,2), and the two next-to-last AC subbands in the purely horizontal (1,7), and vertical (7,1) frequency dimension.

Finally, Figure D.13 populates the rate–distortion tradeoff sketched in Figure D.5 with empirical data collected from an instrumented version of *MozJPEG*. Each arrow shows how one of 128 randomly



(a) QF 75 **(b)** QF 99 **(c)** QF 100

Figure D.11: Share of non-zero DCT coefficients modified by Trellis optimization.

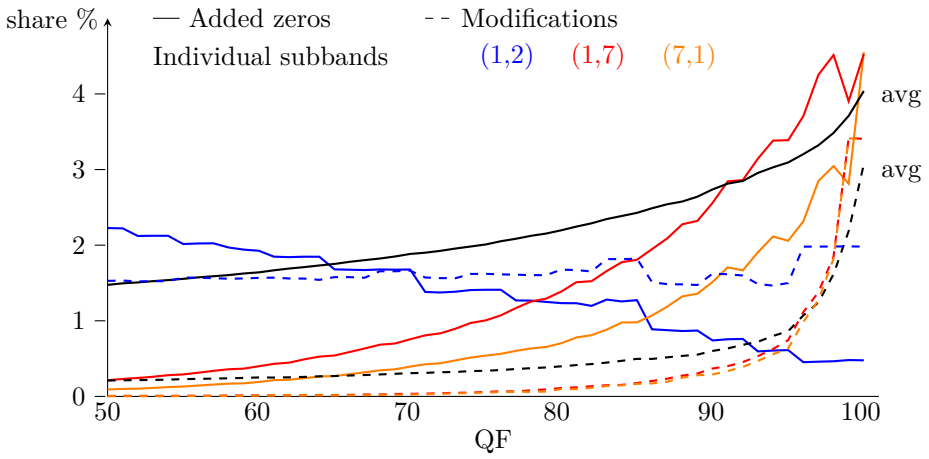


Figure D.12: The average share of *additional zeros* and *non-zero modifications* caused by Trellis optimization in the luminance channel of 100 512×512 images from the Alaska dataset for all QFs from 50 to 100.

selected blocks from a sample image moves in the size–distortion space. The same blocks are selected for each of the three QFs. As expected, the algorithm moves blocks to the upper left, i.e., reducing the size in bits and slightly increasing the distortion, crossing imaginary indifference lines at 45°. Horizontal left arrows indicate blocks that can be encoded in fewer bits without increasing distortion. This can only happen if the initially quantized coefficient is rounded upwards and the quantization error is exactly half of the quantization factor. Our experiments show that the special case in the numerical example of Section D.3.3 actually happens in practice. The concentration of blocks at size 3 for QF 75 can be explained by the bits required to encode the EOB symbol according to the image's Huffman table (cf. Tab. D.2).

D.5 Discussion

The adoption of *MozJPEG* has implications for multimedia forensics in research and practice. In this section we map out avenues for future research.

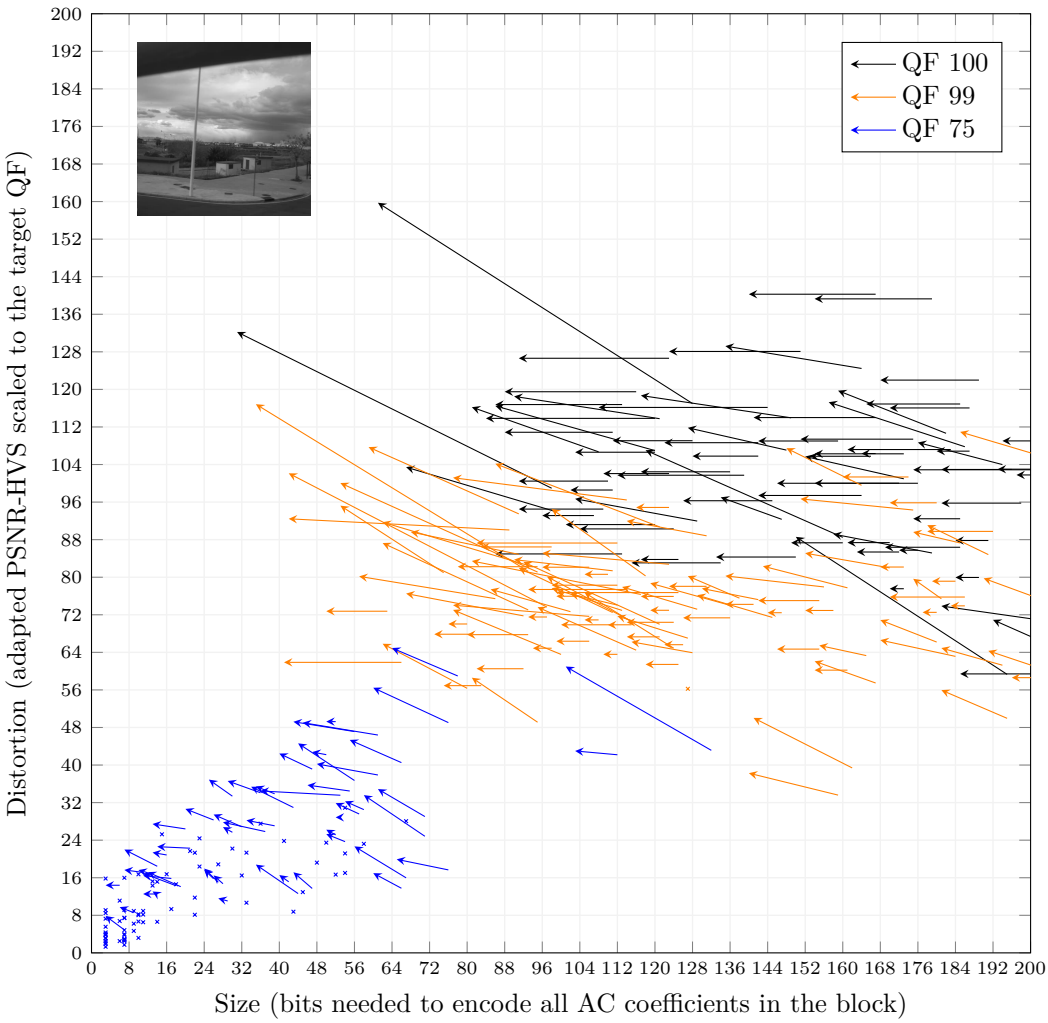


Figure D.13: The effect of *MozJPEG*'s Trellis optimization on selected blocks in the luminance channel for an example image from the Alaska dataset [9].

D.5.1 Implications for Steganography

Besides one exception [7], all research in JPEG steganography and steganalysis assumes baseline JPEG. All practical steganographic tools we are aware of support sequential mode only. Consequently, a progressive JPEG most likely does not contain steganography embedded with the known tools.

However, steganalysts may not only benefit. Modern learning-based steganalysis is sensitive to cover–source mismatch: detectors degrade significantly if they are run on material that deviates from the training data [18]. Early evidence suggests that the choice of the JPEG compression library may contribute to this mismatch [3]. Since *MozJPEG* is being widely adopted, but barely considered by researchers, the true extent of this error remains unknown. Specifically, the changes to the DCT coefficients introduced by Trellis optimization resemble in part the changes of popular embedding functions. For example, F5 [40] decrements the absolute value of DCT coefficients and inflates the number of zeros, quite akin *MozJPEG*. Consequently, pristine images compressed with *MozJPEG*'s

default may appear as false positives in steganalysis. More research is required to evaluate and quantify this effect. Also, steganalysis based on JPEG compatibility [12, 16] is sensitive to the very details of the implementation and known methods should be revisited in the light of *MozJPEG*.

Finaly, steganographers might try to mimic these compression artifacts in order to hide secret messages. While the capacity is probably very low given the low number of changes made during Trellis optimization (cf. Figures D.10 and D.11), such an embedding function could be very secure. Another insight for steganography concerns the generalization of known embedding functions from grayscale (single-channel) to color. An open question in the field is whether independent embedding in all color channels is secure. Since *MozJPEG* optimizes DCT coefficients in each channel independently, there exists at least one benign process which does exactly this. Hence, steganalysis exploiting dependencies in color channels may be less promising than researchers suggest [23].

D.5.2 Implications for Forensics

Similarly to steganalysis, most JPEG forensic techniques were designed for sequential images. Therefore, these techniques may be prone to decision errors if presented with images from *MozJPEG*. For example, techniques for detecting JPEG double compression usually rely on assumptions about the distribution of DCT coefficients. Future research should re-evaluate and, if necessary, adapt existing methods and tools. First and foremost, techniques that rely on assumptions about the distribution of DCT coefficients or repeated quantization [33] seem most affected.

However, *MozJPEG* also provides a number of opportunities for forensic analysis. A common task in forensics is to identify the compression history [6]. A next logical step would be to develop and evaluate a detector for *MozJPEG*.

The very fact that progressive mode has been used, the scan script, and the use of custom Huffman tables may reveal information about the origin and authenticity of an image. Since baseline *libjpeg* and *libjpeg–turbo* use fixed Huffman tables by default, and a fixed scan script if instructed to produce progressive output, there is little identifying information in these entries. As demonstrated in [29], the adoption of *MozJPEG* has changed this.

Another interesting trace is the specific scan script used in progressive images. In a preliminary experiment, we analyzed the scan scripts in social media images from the Forchheim image database [19] and find that 97% use progressive mode. While images assigned to Facebook, Telegram, and Twitter use the default scan script of *libjpeg* and *libjpeg–turbo*, images assigned to WhatsApp and Instagram contain different, distinct scan scripts. These findings suggest that different platforms fine-tune the scan script to their needs. Figures D.14–D.16 show the scan scripts found. More research is needed to validate these findings independently and over time, while controlling the device type and software version of sender and recipient.

D.5.3 Implications for Watermarking

Custom scan scripts may also have applications in digital watermarking. Compressing a JPEG with a unique custom scan script can serve as a fragile watermark to recognize marked images or to detect recompression if a supposedly marked image does not have the specific script anymore. To reduce the risk that other images accidentally share the marking scan script, it can be designed in a “useless” way, e. g., transmitting less relevant information first.

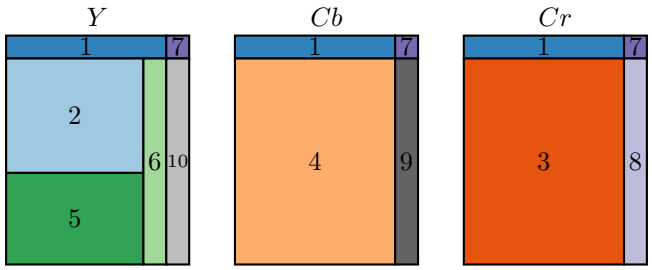


Figure D.14: The scan script found in images in the Forchheim database assigned to *Telegram*, *Twitter*, and *Facebook*. The script is identical to the standard scan script used by *libjpeg* and *libjpeg–turbo*.

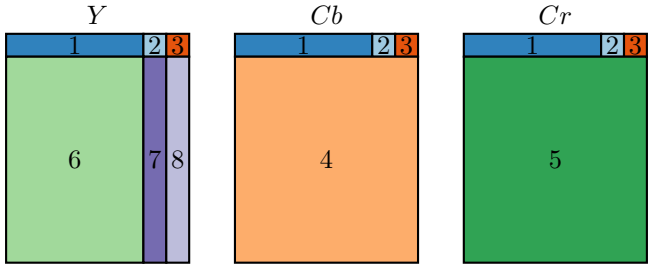


Figure D.15: The scan script found in all images in the Forchheim database assigned to *WhatsApp*.

D.6 Conclusion

Progressive JPEG has become more prevalent than commonly assumed, presumably due to the adoption of *MozJPEG*, an open-source library optimized for web publishers. To the best of our knowledge, we are the first to document the optimizations implemented in this library to make them accessible to the multimedia security community. Our experiments reveal characteristic traces in images compressed with *MozJPEG*. We discuss how these may affect established methods in steganography, steganalysis, image forensics, and watermarking. In particular researchers proposing learning-based methods should in the future include images compressed with *MozJPEG* in their evaluation protocol.

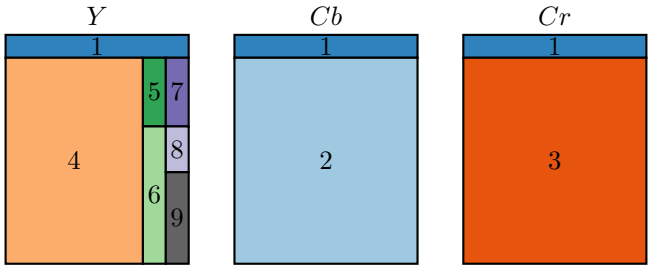


Figure D.16: The scan script found in all images in the Forchheim database assigned to *Instagram*.

Acknowledgements

We thank Maximilian Hils for his support on the web crawling measurement and Benedikt Lorch and Martin Beneš for valuable comments on the draft. This work is funded by the EU's Horizon 2020 program under grant agreement No. 101021687 (UNCOVER).

References

[1] Stephen Arthur. Making photos smaller without quality loss, 2017. https://engineeringblog.yelp.com/2017/06/making-photos-smaller.html (accessed: Jan 9, 2023).

[2] Tomer Bar. Faster photos in facebook for ios, 2018. https://engineering.fb.com/2015/01/28/ios/faster-photos-in-facebook-for-ios/ (accessed: Jan 9, 2023).

[3] Martin Beneš, Nora Hofer, and Rainer Böhme. The effect of the jpeg implementation on the cover-source mismatch error in image steganalysis. In *European Signal Processing Conference*, pages 1057–1061. IEEE, 2022.

[4] Martin Beneš, Nora Hofer, and Rainer Böhme. Know your library: How the libjpeg version influences compression and decompression results. In *Workshop on Information Hiding and Multimedia Security*, pages 19–25. ACM, 2022.

[5] Mike Bishop et al. Hypertext transfer protocol version 3 (http/3). *Internet Engineering Task Force, Internet-Draft draft-ietf-quic-http-34*, 2021.

[6] Jan Butora and Patrick Bas. High quality jpeg compressor detection via decompression error. In *GRETSI*, 2022.

[7] Jan Butora, Pauline Puteaux, and Patrick Bas. Errorless robust jpeg steganography using outputs of jpeg coders. *arXiv preprint arXiv:2211.04750*, 2022.

[8] Matthias Carnein, Pascal Schöttle, and Rainer Böhme. Forensics of high-quality jpeg images with color subsampling. In *WIFS*, pages 1–6. IEEE, 2015.

[9] Rémi Cogranne, Quentin Giboulot, and Patrick Bas. The alaska steganalysis challenge: A first step towards steganalysis. In *IH&MMSec*, pages 125–137. ACM, 2019.

[10] Wikimedia commons. Help:jpeg, 2017. https://commons.wikimedia.org/wiki/Help:JPEG, (accessed: Feb 13, 2023).

[11] Matt Crouse and Kannan Ramchandran. Joint thresholding and quantizer selection for transform image coding: entropy-constrained analysis and applications to baseline jpeg. *Transactions on Image Processing*, 6(2):285–297, 1997.

[12] Eli Dworetzky, Edgar Kaziakhmedov, and Jessica Fridrich. Advancing the jpeg compatibility attack: Theory, performance, robustness, and practice. In *Workshop on Information Hiding and Multimedia Security*. ACM, 2023.

[13] Karen Egiazarian, Jaakko Astola, Nikolay Ponomarenko, Vladimir Lukin, Federica Battisti, and Marco Carli. New full-reference quality metrics based on hvs. In *International Workshop on Video Processing and Quality Metrics*, volume 4, 2006.

[14] Instagram Engineering. Under the hood: Instagram in 2015, 2015. https://instagram-engineering.com/under-the-hood-instagram-in-2015-8e8aff5ab7c2, (accessed: Feb 13, 2023).

[15] Fresco. An image management library, 2023. https://frescolib.org/, (accessed: Feb 13, 2023).

[16] Jessica Fridrich, Miroslav Goljan, and Rui Du. Steganalysis based on jpeg compatibility. In *Multimedia Systems and Applications IV*, volume 4518, pages 275–280, 2001.

[17] Andrew Galloni and Kornel Lesiński. Progressive image streaming, 2020. https://blog.cloudflare.com/parallel-streaming-of-progressive-images/, (accessed: Mar 05, 2023).

[18] Quentin Giboulot, Rémi Cogranne, Dirk Borghys, and Patrick Bas. Effects and solutions of cover-source mismatch in image steganalysis. *Signal Processing: Image Communication*, 86:115888, 2020.

[19] Benjamin Hadwiger and Christian Riess. The forchheim image database for camera identification in the wild. In *Pattern Recognition, Computer Vision, and Image Processing*, pages 500–515. Springer, 2021.

[20] Graham Hudson, Alain Léger, Birger Niss, István Sebestyén, and Jørgen Vaaben. Jpeg-1 standard 25 years: past, present, and future reasons for a success. *Journal of Electronic Imaging*, 27(4):040901–040901, 2018.

[21] Jaehan In, Shahram Shirani, and Faouzi Kossentini. Jpeg compliant efficient progressive image coding. 5:2633–2636, 1998.

[22] Jana Iyengar, Martin Thomson, et al. Quic: A udp-based multiplexed and secure transport. In *RFC 9000*. 2021.

[23] Matthias Kirchner and Rainer Böhme. "steganalysis in technicolor" boosting ws detection of stego images from cfa-interpolated covers. In *International Conference on Acoustics, Speech and Signal Processing*, pages 3982–3986. IEEE, 2014.

[24] ShiYue Lai and Rainer Böhme. Block convergence in repeated transform coding. In *ICASSP*, pages 3028–3032. IEEE, 2013.

[25] Thomas Lane. Using the ijg jpeg library, 1994. `https://www.freedesktop.org/wiki/Software/libjpeg/`, (accessed: Jan 9, 2023).

[26] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Annual Network and Distributed System Security Symposium*, 2019. `//tranco-list.eu/list/5Y5GN`, (accessed: Mar 05, 2023).

[27] libjpeg turbo. What about mozjpeg?, 2017. `https://www.libjpeg-turbo.org/About/Mozjpeg`, (accessed: Feb 13, 2023).

[28] The libjpeg-turbo Project. libjpeg–turbo, 2022. `https://libjpeg-turbo.org/` (accessed: Jan 28, 2023).

[29] Sean McKeown, Gordon Russell, and Petra Leimich. Fingerprinting jpegs with optimised huffman tables. *The Journal of Digital Forensics, Security and Law*, 13(2), 2018.

[30] Mozilla Foundation. Mozjpeg: Improved jpeg encoder, 2014. `https://github.com/mozilla/mozjpeg` (accessed: Jan 28, 2023).

[31] Norman Nill. A visual model weighted cosine transform for image compression and quality assessment. *Transactions on Communications*, 33(6):551–557, 1985.

[32] Greg Notess. The wayback machine: The web's archive. 26(2):59–61, 2002.

[33] Cecilia Pasquini and Rainer Böhme. Towards a theory of jpeg block convergence. In *International Conference on Image Processing*, pages 550–554. IEEE, 2018.

[34] Randall Reininger and Jerry Gibson. Distributions of the two-dimensional dct coefficients for images. *Transactions on Communications*, 31(6):835–839, 1983.

[35] Thomas Stiitz and Andreas Uhl. Image confidentiality using progressive jpeg. In *International Conference on Information Communications & Signal Processing*, pages 1107–1111. IEEE, 2005.

[36] Twitter. Twitter image pipeline (a.k.a. tip), 2023. `https://github.com/twitter/ios-twitter-image-pipeline`, (accessed: Feb 13, 2023).

[37] W3Techs. Usage statistics of jpeg for websites, 2022. `https://w3techs.com/technologies/details/im-jpeg` (accessed: Jan 9, 2023).

[38] G.K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.

[39] Jiangtao Wen, M. Luttrell, and J. Villasenor. Trellis-based r-d optimal quantization in h.263+. *Transactions on Image Processing*, 9(8):1431–1434, 2000.

[40] Andreas Westfeld. F5—a steganographic algorithm: High capacity despite better steganalysis. In *Information Hiding: 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings*, pages 289–302. Springer, 2001.

E. Detecting Trellis Artifacts

Author

Nora Hofer, University of Innsbruck

Title

Increasing Trust in Image Analysis by Detecting Trellis Quantization in JPEG Images

Conference

IEEE International Conference on Image Processing (ICIP '24)
Abu Dhabi, UAE · October, 27–30, 2024

Abstract

JPEG image forensics investigates the authenticity and origin of compressed images. Many established methods rely on assumptions about the statistical distribution of quantized discrete cosine transform coefficients. However, JPEG implementations that use trellis quantization, such as *mozjpeg*, produce images that challenge these assumptions. In this study, we demonstrate that artifacts resulting from trellis quantization can compromise the reliability of established forensic methods and cause false alarms for innocuous images. We address this issue by presenting methods to detect trellis artifacts and validating their robustness in scenarios commonly encountered in forensic analyses.

Table E.1: State-of-the-art steganalysis models misclassify innocent cover images if they are unaware of trellis artifacts.

Detection performance			
Embedding	*libjpeg–turbo*		*mozjpeg*
	Baseline acc.	FPR	FPR
nsF5 [14]	99%	1%	**99%**
UERD [16]	93%	4%	**43%**
J-UNIWARD [18]	91%	8%	**94%**

ImageNet-pretrained, 32 batch size, 0.25 dropout rate, 0.0001 learning rate, Adam, QF 75, 0.4 bits per non-zero AC coefficients (bpnzAC), ALASKA2.

E.1 Introduction

Trellis quantization [24] addresses the rate-distortion problem in data compression by finding the path through a trellis structure that minimizes a cost function. This cost function balances the size in bits needed to encode a coefficient value against the distortion introduced by quantization. By evaluating the cumulative cost of different paths, trellis quantization identifies the sequence of quantization steps that results in the most efficient compression with minimal loss in quality. Trellis quantization is particularly effective in the compression of transform coefficients, such as those obtained from the Discrete Cosine Transform (DCT) in video [34] and image compression [23].

Mozjpeg [26] is a popular JPEG compression library that implements a variant of trellis quantization by default to achieve reduced file sizes. Specifically, it employs a perceptual model that accounts for the additional distortion introduced by changing coefficients to values with shorter variable-length encodings. These modifications have been found to cause characteristic artifacts in the DCT coefficient distribution of compressed images [17]. While such artifacts can be exploited in image forensics to fingerprint the JPEG implementation [28], they might pose challenges to other forensic applications if they rely on assumptions about the distribution of quantized DCT coefficients.

In recent years, statistical learning has gained popularity in multimedia forensics [19]. However, while machine learning detectors achieve high performances, they are known to be sensitive to training-test mismatches.

Table E.1 demonstrates the detrimental effects of unaddressed trellis artifacts on the detection of image steganography. Similar to related work [37], we train three EfficientNet-B0 detectors [32] on cover and stego images compressed with *libjpeg–turbo*, a widely used JPEG compression library that does not implement trellis quantization. The left and center column in Table E.1 show the baseline accuracy and the flase positive rate (FPR). Next, we evaluate the detectors' sensitivity to images compressed with *mozjpeg*. Up to 99% of all innocuous cover images are now falsely classified as stego images, as shown in the rightmost column. Both test sets use the same images, DCT method, subsampling, and quantization table (QT). The differences in the FPR can, therefore, be attributed to the characteristics of *mozjpeg*'s trellis quantization.

In this paper, we analyze and quantify trellis artifacts and determine characteristics in the frequency distribution of quantized coefficient values. Leveraging these characteristics, we build detectors for trellis artifacts based on analytic modelling and statistical learning. The detectors are intended to serve as forensic preprocessors and can help practitioners that apply forensic tools, to make informed interpretations of their results.

The remainder of the paper is organized as follows: Section E.2 describes processing steps specific to *mozjpeg* and quantifies their effect on the image signal. Section E.3 describes the proposed detectors,

and Section E.4 evaluates their performance for in- and out-of-distribution scenarios. Section E.5 discusses our findings and their implications for the research community before Section E.6 concludes our paper.

E.2 Mozjpeg

Mozjpeg has recently attracted attention within the multimedia security community due to its changes in output images (*e.g.*, [6, 25]). The library implements several compression optimizations to reduce file size and improve the perceptual image quality, namely overshoot deringing, adapted QTs, trellis quantization, and default progressive encoding with optimized scan scripts and Huffman tables. The first three alter the DCT coefficients, whereas the latter optimize the stream encoding without altering coefficients. This section reviews the background of the signal-based optimizations and investigates their effects on coefficients in an isolated manner. We refer the reader to [17] for a detailed description of mozjpeg's optimized stream encoding.
To quantify these effects, we use the *image change rate*, *i.e.*, the share of images with at least one changed DCT coefficient, and the average *coefficient change rate*, *i.e.*, the number of changed DCT coefficients normalized by the number of non-zero DCT coefficients. We use 10 000 never-compressed images of size 512×512 randomly sampled from ALASKA2 [9], the benchmark dataset in steganography. As our reference, we compress these images using *mozjpeg* v4.0.3 with all optimizations disabled. We then selectively enable individual optimizations and measure the image and coefficient change rates compared to our reference. We do this for the quality factors (QFs) 50, 75, 80, 85, 90, 95, and 100.

Overshoot deringing During JPEG compression, the DCT converts blocks of 8×8 pixels into a frequency domain representation. When blocks contain combinations of pixels that cannot be exactly represented by the discretized cosine functions, the DCT causes ringing at the upper (overshoot) and lower (undershoot) value range, also known as the *Gibbs phenomenon* [15]. During decompression, the JPEG decoder clips the positive overshooting values to 255 and negative undershooting values to 0. This results in visual artifacts known as ringing artifacts. They typically appear around sharp edges or text and are visible as alternating light and dark signals.
Starting in version 3, *mozjpeg* implements an overshoot deringing algorithm that tackles the ringing of positively overshooting pixels during compression [21, 30]. It enlarges the upper bound of pixel values and deliberately moves overshoots outside the 8-bit range, hiding ringing waves from the decoder. The allowed overshoot is based on the sharpness of edges. The algorithm extrapolates the pixel in a block with the highest value using *Catmull-Rom* splines.

Effect: 18% of all images from our dataset are changed by the overshoot deringing algorithm. Less than 1% of the DC and AC coefficients in those images are changed. The change rates are largely independent of the QF.
The low change rates can be attributed to the dataset, which contains photographs of natural scenes with few instances of ringing. To highlight the effect of the image content, we repeat this evaluation on images that mainly consist of text. Specifically, we collect 1 000 PDF documents [1] and convert them to the TIFF format using the Python package *pdf2image* with 300 dpi resolution. We center crop them to 512×512 and compress with *mozjpeg*.

Effect: The deringing optimization now changes now changes more than 92% of the images. The coefficient change rate increases to 40%. Again, the change rates are largely independent of the QF.

[1] `https://github.com/tpn/pdfs`

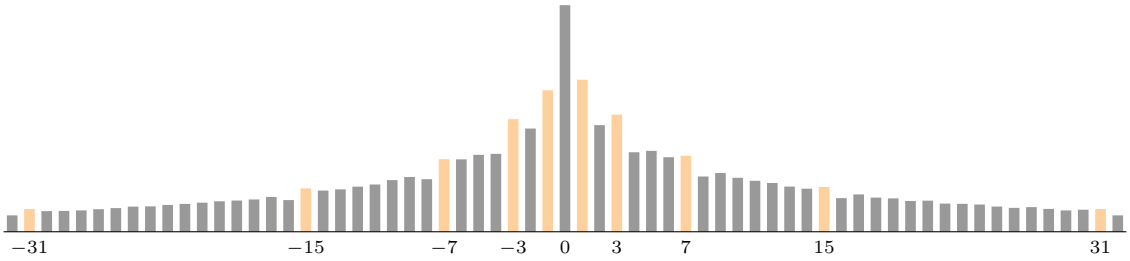


Figure E.1: Candidate values (orange) are amplified in the coefficient histogram of images compressed with trellis quantization. (Data for AC subband 1, compressed with QF 90.)

Quantization tables *Libjpeg* and *libjpeg–turbo* (with one exception [3]) use the QTs for luminance and chrominance components as defined in Annex K of the JPEG standard [33]. While supporting several QTs, including the standard tables, *mozjpeg* implements stronger quantization by default and uses specific QTs [31]. To measure the change rate in images using *mozjpeg*'s specific QTs, we compare them to images compressed using the tables defined in the standard.

Effect: We observe changes in all images except for QF 100, where all entries of the QT are 1 and no images are changed. The AC change rate is between 49% and 60% for the tested QFs. We observe no changes in DC coefficients.

Trellis quantization *Mozjpeg*'s trellis quantization aims to improve the rate–distortion tradeoff after an initial quantization step for each 8×8 coefficient block. It uses a perceptual model to calculate the distortion implied by reducing non-zero DCT coefficients to values of shorter bit sizes. Following [17], we denote y_j^* and y_j^{**} as the coefficient value at subband j before and after trellis quantization, respectively. We define the set

$$\mathcal{C} = \{\pm(2^k - 1) : k = 1, \ldots, 15\}, \tag{E.1}$$

which leads us to the **candidate values** for a given y_j^*,

$$\mathcal{C}_j = \{c \in \mathcal{C} : |c| < |y_j^*|\} \cup \{y_j^*\}. \tag{E.2}$$

Moreover, we define the set of **outer neighbors**

$$\mathcal{C}^{++} = \{\pm 2^k : k = 1, \ldots, 15\}. \tag{E.3}$$

For a given y_j^*, the algorithms evaluates the cost implied by replacing y_j^* with any $c \in \mathcal{C}_j$, weighs the additional distortion against the bits saved by shorter encoding, and sets y_j^{**} to the c with the lowest cost.

Effect: For all QFs, more than 99% of all images contain changes. The average change rate over all AC coefficients is between 10% and 18% for the measured QFs, with a decreasing trend for higher QFs. The opposite trend is observable for the change rate of DC coefficients, which is constant below 5% for QFs up to 90 and exceeds 10% at QF 100.
Figure E.1 shows the distribution of quantized AC coefficients after trellis quantization. For natural images, the coefficient distribution can be approximated by Laplacian distributions [29]. Observe, that this is not the case for images compressed with trellis quantization. Here, the probability mass increases for bins of candidate values and decreases for their outer neighbors, the pair of which we call **candidate pairs**.
Note that the effect of trellis quantization is limited when recompressing previously compressed images. We demonstrate this in a simplified example: Let $y_j = 540$ be an unquantized coefficient

value and $q_j = 72$ the quantization factor. Quantization divides y_j by q_j to 7.5 and rounds to the nearest integer $y_j^* = 8$. During decompression y_j^* gets dequantized by $y_j^* \times q_j$, resulting in $y_j' = 576$, which is now evenly divisible by q_j. This prevents trellis quantization from modifying the rounding in a direction of fewer bits. In reality, multiple rounding operations during de- and recompression influence the effectiveness.
In [10], the authors uncover the quantization factor by searching for two local minima in the quantization error of recompressed images. It seems intuitive to follow their approach for the detection of trellis quantization and recompress an image with and without trellis quantization before comparing the magnitude of artifacts in the recompressed images. However, as the effectiveness of trellis quantization is limited in previously compressed images, this approach is unsuitable for our means.

E.3 Detectors

In this section, we propose methods based on analytic modelling and statistical learning for detecting trellis artifacts in the distribution of quantized DCT coefficients. We use coefficients of the first eight AC DCT subbands (in zigzag order) with values $i \in \mathcal{I} = \{-32, \dots, 32\}$. This ensures that our methods generalize to low QFs, where bins with higher absolute values are often unpopulated. Without loss of generality, we consider i as an absolute value and denote the outer neighboring coefficient value as $i + 1$.
To construct the dataset for the detection of trellis artifacts, we use the same sample of 10 000 never-compressed images from the ALASKA2 dataset and compress with *mozjpeg* v4.0.3 with default settings (4:2:0 subsampling, DCT *ISLOW*, *mozjpeg*'s QTs, progressive encoding). We generate two datasets: The negative class are images where all optimizations are disabled during compression. The positive class are images compressed with trellis quantization. We use a $50 : 50$ train–test split.

E.3.1 Analytic modelling

Our modelling based detection aims to analytically describe the distribution of DCT coefficients of images compressed with trellis quantization. In our measurements in Section E.2 we find that for images from the ALASKA2 dataset and coefficient values greater than 2, there are no changes further than from $c + 1$ to c. As changes for the absolute candidate values $2, 1$, and 0, are more complicated, we exclude them from the analytical models. We propose two approaches.

Calibration Trellis artifacts resemble in part those of popular steganographic embedding functions. For example, F5 [35] decrements the absolute value of DCT coefficients and inflates the number of zeros. Previous work on the detection of F5 uses *calibration*, which exploits the regularity of the JPEG 8×8 grid. Calibration estimates the histogram of the cover image by cropping a decompressed stego image by 4 pixels on each side and recompresses it using the QT of the stego image [12]. The authors calculate the embedding rate by comparing the histogram of the stego image and the estimated cover histogram. We build on their approach and use calibration to estimate the histogram of an image before trellis quantization. Let H_i be the histogram bin of an image compressed with trellis quantization holding the number of AC coefficients with value equal to i, and $\hat{H}_i$ the respective bin before trellis quantization, as estimated by calibration. We define $H_i \coloneqq \hat{H}_i + \hat{H}_{i+1} \times \alpha_i$, where α_i is the relative frequency of coefficients with value $i+1$ being changed to i. This results in

$$\alpha_i = \frac{H_i - \hat{H}_i}{\hat{H}_{i+1}}. \tag{E.4}$$

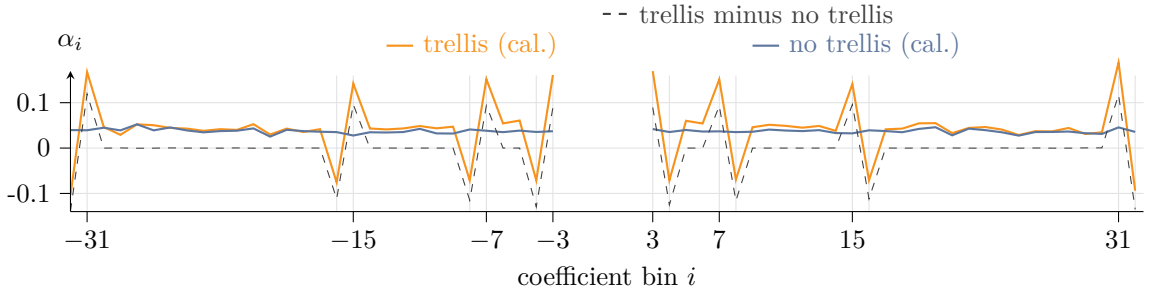


Figure E.2: The relative frequency of coefficients being changed to their inner neighbor. α_i increases at candidate values for images compressed with trellis quantization.

α_i of 0 refers to the same number of coefficients at bin i before and after calibration, suggesting no changes caused by trellis quantization. A positive α_i and a negative α_{i+1}, at bins where $i \in \mathcal{C}$, indicates the presence of trellis artifacts. As shorthands we write α_c for $\alpha_{i\in\mathcal{C}}$ and α_{c+1} for $\alpha_{i\in\mathcal{C}^{++}}$. (See Eqs. E.1 and E.3 for the set definitions.)
Figure E.2 visualizes α_i as the average over the training set. For images compressed with trellis quantization, we observe high values for α_c and low values for α_{c+1}. For images compressed without trellis quantization, we observe a nearly constant α_i with no deviations at c or $c+1$. The dashed line in Figure E.2 plots α_i measured by comparing the same images, compressed with and without trellis quantization, *i.e.*, without applying calibration. The difference between the dashed line and α_i measured for images compressed with trellis quantization is the calibration estimation error. Note that we do not have access to this value when detecting trellis artifacts.
For our purposes we aggregate α_i to α_c and define

$$\alpha_{\mathcal{C}} = \sum_{i\in\mathcal{C}} (\alpha_i - \alpha_{i+1}) \tag{E.5}$$

as score to detect the presence of trellis artifacts. We use Youden's $\tilde{\mathcal{J}}$ statistic [36] to select the optimal threshold based on the classification performance on the training set.

Vampire neighborhoods In a second approach, we assume a monotonous histogram and measure deviations at candidate pairs with regard to their inner and outer neighbors. We call this set **candidate neighborhoods** $(c_{i-1}, \ldots, c_{i+2})$. We measure the deviation using a *vampire score* β,[2] and define

$$\beta_i = H_i - \frac{H_{i-1} + H_{i+2}}{2} + H_{i+1} - \frac{H_{i-1} + H_{i+2}}{2}. \tag{E.6}$$

Figure E.3 visualizes β_i as the average over the training set. For images compressed without trellis quantization we observe a smooth β_i. For images compressed with trellis quantization we can see spikes at bins where $i \in \mathcal{C}$. This indicates an increased frequency of candidate values and a decreased frequency of their outer neighbors with regard to the candidate neighborhood. We aggregate β_i to β_c, simplify Eq. E.6, and calculate

$$\beta_{\mathcal{C}} = \sum_{i\in\mathcal{C}} (H_i - H_{i-1} + H_{i+1} - H_{i+2}) \tag{E.7}$$

as score to detect the presence of trellis artifacts. Again, we find the empirically optimal threshold using Youden's $\tilde{\mathcal{J}}$.

[2] The name originates from trellis artifacts in the plotted histogram, which reminded the authors of inverted vampire teeth.

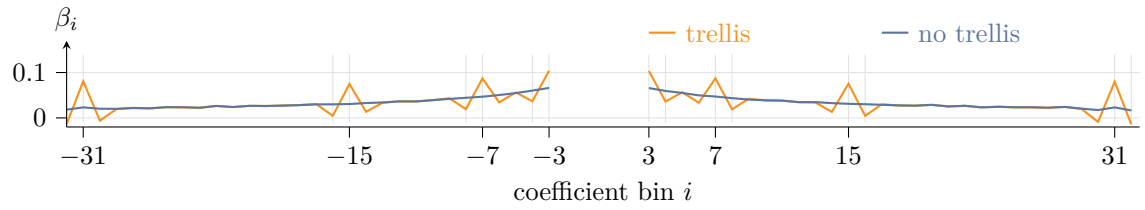


Figure E.3: The average deviation of H_i from a monotonous pattern within neighborhoods. β_i increases at candidate values for images compressed with trellis quantization.

E.3.2 Statistical learning

Statistical learning refers to the use of machine learning to infer patterns from training images. We evaluate three different types of features extracted from the DCT coefficients. The features are then classified using an ensemble of Fisher linear discriminant base learners [11], which is commonly used in steganalysis [20].

Cartesian calibration Like before, we use calibration to estimate the histogram before trellis quantization. Next, we extract candidate neighborhood features of calibrated images and use them together with the same set of histogram features from the original images to train the classifier on the cartesian product. There are ten candidate values in $\mathcal{I}$. The resulting features have dimensions of $4 \times 8 \times 10 \times 2 = 640$.

Vampire neighborhoods As before, we train the classifier on candidate neighborhood features, this time without cartesian calibration. The resulting features have dimensions of $4 \times 8 \times 10 = 320$.

JRM features As a third approach, we use features extracted from an ensemble of JPEG Rich Models [13] (JRM) with 11 255 dimensions to train a classifier. JRMs model different types of dependencies between adjacent coefficient subbands.

E.4 Results

Table E.2 reports the accuracies on the test set. Both analytic detectors perform well for high QFs but degrade for low QFs. The decline in accuracy for low QFs is also visible, albeit less prevalent, for the detectors based on statistical learning. They all achieve high accuracies. The detector trained on JRM features reaches near-perfect test accuracy, also for low QFs.

E.4.1 Robustness

Out-of-distribution scenarios occur when the distribution of coefficients in test images differs from that in the training images. In this section, we report the robustness in the following scenarios: detectors tested on images of a different QF than the training images, detectors tested on stego images while trained on covers, detectors tested on double-compressed images while trained on single-compressed, and images that were compressed with the deringing optimization.

Unseen QFs We find that the detectors generalize well to higher QFs. For lower QFs, the performance decreases slightly due to missed detections. The detector based on JRM features is an exception. It does not generalize well to lower QFs, especially when trained on QFs above 90. For all

Table E.2: Detection accuracies of proposed trellis detectors.

	Analytic detectors		**Learning detectors**		
QF	cal.	vamp.	cal.	vamp.	JRM
100	95%	99%	100%	100%	100%
95	84%	92%	99%	99%	100%
90	80%	88%	99%	98%	100%
85	75%	82%	99%	97%	100%
80	72%	79%	98%	96%	100%
75	71%	76%	98%	96%	99%
50	72%	75%	97%	93%	98%

following out-of-distribution experiments, we focus on our detectors based on vampire neighborhoods (without calibration). The positive class $\oplus$ contains images compressed **with** trellis quantization, and the negative class $\ominus$ contains images compressed **without** trellis quantization. We apply processing operations to either the positive or the negative class and report the effect on the performance in Table E.3. The reference is the in-distribution performance on images of QF 90.

Steganography In Section E.1, we show that a state-of-the-art steganalysis model fails when facing images containing trellis artifacts. Now we investigate the opposite scenario, namely whether our trellis artifact detectors are robust to steganography. We evaluate them for three prominent embedding methods, nsF5, UERD, and J-UNIWARD, with an embedding rate of 0.4 bpnzAC. We assume that a high embedding rate increases the difficulty of identifying trellis artifacts.

Exp. 1: $\oplus$ **trellis** $\ominus$ **no trellis, stego**
Both detectors differentiate between the positive and the negative class with the same performance as before. They are robust against stego embeddings in images without trellis artifacts.

Exp. 2: $\oplus$ **trellis, stego** $\ominus$ **no trellis**
The performance of our detectors drops slightly due to an increase in missed detections. While the embedding with nsF5 has little effect on our detectors, the embeddings with UERD and J-UNIWARD seem to wash out trellis artifacts. However, at least 70% of all images from $\oplus$ are still correctly classified by the analytic detector and 78% by the learning-based detector. Note that this is a hypothetical experiment. No practical steganographic tool we know of uses trellis quantization during compression.

Double compression artifacts Double compression causes periodic artifacts and discontinuities in the coefficient distribution. To evaluate our detectors, we use images compressed with QF_1=90 and recompress them with QF_2. We evaluate the detectors trained on single compressed images of QF_2.

Exp. 3: $\oplus$ **trellis** $\ominus$ **no trellis, double compression**
When $\mathrm{QF}_2 > \mathrm{QF}_1$, the performance of both detectors drops. As for the analytic detector, $\beta_{\mathcal{C}}$ of the negative class now roughly resembles the pattern of trellis artifacts at some candidate values, causing the performance to decrease to 55%. We observe the same for QF_2=90 with a drop to 45%. The learning-based detector fails for QF 93 (acc. = 50%) but is robust against double compression

Table E.3: Robustness of two detectors based on candidate neighborhoods to deviations in distributions. The effect is measured as the performance difference in %-pts. Reference in-distribution performance for QF 90 is given at the top.

	Analytic detector			**Learning detector**		
	Acc.	FPR	FNR	Acc.	FPR	FNR
Ref.	88.48	12.40	10.60	98.18	1.62	2.02
Exp. 1: Steganography in $\ominus$ no effect						
Exp. 2: Steganography in $\oplus$						
nsF5	− 3		+ 5	− 0		+ 4
UERD	− 7		+14	− 4		+ 9
J-UNI.	− 9		+18	− 9		+20
Exp. 3: Double compression in $\ominus$ (QF_1: 90)						
QF_2: 93	−37	+75		−49	+98	− 0
QF_2: 90	−44	+88		+ 1	− 1	− 0
QF_2: 87	+ 5	−11	−2	+ 1	− 2	
QF_2: 75	+ 2	− 5		−36	+72	+ 0
Exp. 4: Double compression in $\oplus$ (QF_1: 90)						
QF_2: 93	+ 4		− 8	+ 0	+ 2	− 1
QF_2: 90	+ 0		+ 0	+ 1	− 1	− 1
QF_2: 87	−14		+29	−48		+97
QF_2: 75	− 2		+ 4	+ 2	− 0	− 3

$\oplus$ positive class (trellis) $\ominus$ negative class (no trellis)

with $QF_2 = QF_1$. When $QF_2 < QF_1$, $\beta_{\mathcal{C}}$ follows a different pattern. This amplifies the differences between the classes and slightly increases the performance. Interestingly, the performance of the learning based detector decreases for QF_2=75.

Exp. 4: $\oplus$ **trellis, double compression** $\ominus$ **no trellis**
Again, double compression with $QF_2 > QF_1$ causes $\beta_{\mathcal{C}}$ to resemble the pattern of trellis artifacts. However, in this case, it happens in the positive class, which amplifies trellis artifacts. The performance of both detectors increases slightly. Respectively, $\beta_{\mathcal{C}}$ for $QF_2 < QF_1$ follows a different pattern than trellis artifacts; now, concealing them. This leads to missed detections of both detectors for QF_2=87. Interestingly, double compression with QF_2=75 has close to no effect.
To investigate if the results of Exp. 3 and 4 impair the reliability of double compression detection, we apply the pre-trained double compression detection model DJPEG-torch [27] on images compressed with trellis quantization. DJPEG-torch uses histogram features and extracted quantization tables as input to a convolution neural network. We find that it is robust, also to images where double compression amplifies trellis artifacts.

Overshoot deringing artifacts To ensure the reliability on images from *mozjpeg*, we measure the effect of overshoot deringing artifacts on our detector. We use the ALASKA2 dataset.

Exp. 5: $\oplus$ **trellis** $\ominus$ **no trellis, deringing** ,
Exp. 6: $\oplus$ **trellis, deringing** $\ominus$ **no trellis**

Overshoot deringing does not affect on the performance of our detectors for the tested QFs. For the sake of space, we do not include this result in Table E.3.

E.5 Discussion

In this paper, we find that state-of-the-art steganalysis models misclassify innocuous cover images when they are unaware of trellis artifacts. To address this, we propose methods based on analytic modelling and statistical learning to detect trellis artifacts in compressed JPEG images. The detectors are intended to help practitioners applying forensic tools to make informed interpretations of their results and avoid unexpected behavior of tools tailored for different libraries when analyzing images compressed with *mozjpeg*.

Our detectors are robust against steganographic embeddings of three popular embedding methods and artifacts from *mozjpeg*'s overshoot deringing algorithm. We find that double compression operations can diffuse trellis artifacts, causing our detectors to fail.

The characteristic of double compression artifacts in an image can reveal information about the history of an image and potential manipulations. We find that the effectiveness of trellis quantization is limited in previously compressed images. Future research should analyze whether this can be exploited during the detection of manipulations in images compressed with trellis quantization.

Mozjpeg's overshoot deringing algorithm introduces changes in approximately 18% of images capturing natural scenes; however, it changes only 1% of the coefficients. In a dataset of JPEG compressed computer graphics and text, where there are more instances of ringing, it changes up to 40% of the coefficients in 90% of the images. This can have implications for other fields, *e.g.*, the detection of sharpening, where the absence [7] or characteristics [8] of ringing artifacts are used as a telltale for image manipulation.

Our detectors complement previous efforts to fingerprint JPEG libraries. Existing approaches investigate implementation differences in common processing steps, such as DCT [2], chroma subsampling [22], and rounding operations during quantization [1]. Furthermore, [4] leverage the statistical features of recompressed images and [5] use rounding errors of decompressed images. Apparent traces to fingerprint *mozjpeg* are library-specific QTs, and image-specific scan scripts and Huffman tables in progressive images. We concentrated our focus on the image signal, as these parameters can be configured by the user during compression, making them unreliable for the detection of *mozjpeg*.

E.6 Conclusion

It is important to understand optimizations of popular JPEG implementations as many methods in multimedia security rely on subtle traces in the signal originating from compression and decompression operations. Researchers proposing learning-based methods for steganography, steganalysis, or image forensics should include images compressed with *mozjpeg* in their evaluation protocol, and revisit known methods in the light of trellis quantization.

Finally, practitioners should be careful when carrying out forensic tests on images of unknown sources using tools tailored to specific libraries.

Acknowledgements

We thank Benedikt Lorch and Rainer Böhme for their support and valuable comments on the draft, and Martin Beneš for incorporating *mozjpeg* into his `jpeglib` library.

References

[1] S. Agarwal and H. Farid. Photo forensics from JPEG dimples. In *WIFS*, pages 1–6. IEEE, 2017.

[2] S. Agarwal and H. Farid. Photo forensics from rounding artifacts. In *IH&MMSec*, pages 103–114, 2020.

[3] M. Beneš, N. Hofer, and R. Böhme. Know your library: How the libjpeg version influences compression and decompression results. In *IH&MMSec*, pages 19–25, 2022.

[4] N. Bonettini, L. Bondi, P. Bestagini, and S. Tubaro. JPEG implementation forensics based on eigen-algorithms. In *WIFS*, pages 1–7. IEEE, 2018.

[5] J. Butora and P. Bas. High quality JPEG compressor detection via decompression error. In *GRETSI*, 2022.

[6] J. Butora, P. Puteaux, and P. Bas. Errorless robust JPEG steganography using outputs of JPEG coders. *IEEE TDSC*, 2023.

[7] G. Cao, Y. Zhao, and R. Ni. Detection of image sharpening based on histogram aberration and ringing artifacts. In *ICME*, pages 1026–1029. IEEE, 2009.

[8] G. Cao, Y. Zhao, R. Ni, and A. Kot. Unsharp masking sharpening detection via overshoot artifacts analysis. *IEEE Signal Processing Letters*, pages 603–606, 2011.

[9] R. Cogranne, Q. Giboulot, and P. Bas. The ALASKA steganalysis challenge: A first step towards steganalysis. In *IH&MMSec*, pages 125–137. ACM, 2019.

[10] H. Farid. Exposing digital forgeries from JPEG ghosts. *IEEE TIFS*, pages 154–160, 2009.

[11] R. Fisher. The use of multiple measurements in taxonomic problems. *Ann. of Eugenics*, pages 179–188, 1936.

[12] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images: Breaking the F5 algorithm. In *IH*, pages 310–323. Springer, 2003.

[13] J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. *IEEE TIFS*, pages 868–882, 2012.

[14] J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography: Dead ends challenges, and opportunities. In *MM&Sec*, pages 3–14. ACM, 2007.

[15] D. Gottlieb and C. Shu. On the gibbs phenomenon and its resolution. *SIAM review*, pages 644–668, 1997.

[16] L. Guo, J. Ni, W. Su, C. Tang, and Y. Shi. Using statistical image model for JPEG steganography: Uniform embedding revisited. *IEEE TIFS*, pages 2669–2680, 2015.

[17] N. Hofer and R. Böhme. Progressive JPEGs in the wild: Implications for information hiding and forensics. In *IH&MMSec*, pages 47–58. ACM, 2023.

[18] V. Holub, J. Fridrich, and T. Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP*, pages 1–13, 2014.

[19] M. Kharrazi, H. Sencar, and N. Memon. Blind source camera identification. In *ICIP*, pages 709–712. IEEE, 2004.

[20] J. Kodovský, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE TIFS*, pages 432–444, 2011.

[21] L. Kornel. Deringing via overshoot clipping, 2014. https://github.com/mozilla/mozjpeg/pull/101.

[22] B. Lorch and C. Riess. Image forensics from chroma subsampling of high-quality JPEG images. In *IH&MMSec*, pages 101–106, 2019.

[23] M. Marcellin, M. Lepley, A. Bilgin, T. Flohr, T. Chinen, and J. Kasner. An overview of quantization in JPEG 2000. *Signal Processing: Image Communication*, pages 73–84, 2002.

[24] M. W Marcellin and T. Fischer. Trellis coded quantization of memoryless and gauss-markov sources. *IEEE Transactions on Communications*, pages 82–93, 1990.

[25] S. McKeown, G. Russell, and P. Leimich. Fingerprinting JPEGs with optimised huffman tables. *JDFSL*, 2018.

[26] Mozilla Foundation. MozJPEG: Improved JPEG encoder. `github.com/mozilla/mozjpeg`, 2014. Accessed on 25 Jan, 2024.

[27] J. Park, D. Cho, W. Ahn, and H. Lee. Double JPEG detection in mixed JPEG quality factors using deep convolutional neural network. In *ECCV*, pages 636–652, 2018.

[28] A. Piva. An overview on image forensics. *ISRN Signal Processing*, pages 1–22, 2013.

[29] R. Reininger and J. Gibson. Distributions of the two-dimensional DCT coefficients for images. *IEEE TCOM*, pages 835–839, 1983.

[30] T. Richter. JPEG on steroids: Common optimization techniques for JPEG image compression. In *ICIP*, pages 61–65. IEEE, 2016.

[31] N. Robidoux. Re: Better JPEG quantization tables?, 2013. Legacy ImageMagick Discussions Archive.

[32] M. Tan and Q. Le. EfficientNet: Rethinking model scaling for CNNs. In *ICML*, pages 6105–6114. PMLR, 2019.

[33] G. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38:xviii–xxxiv, 1992.

[34] J. Wen, M. Luttrell, and J. Villasenor. Trellis-based RD optimal quantization in H.263+. *IEEE Transactions on Image Processing*, pages 1431–1434, 2000.

[35] A. Westfeld. F5 – a steganographic algorithm: High capacity despite better steganalysis. In *IH*, pages 289–302. Springer, 2001.

[36] W. Youden. Index for rating diagnostic tests. *Cancer*, pages 32–35, 1950.

[37] Y. Yousfi, J. Butora, J. Fridrich, and C. Fuji Tsang. Improving EfficientNet for JPEG steganalysis. In *IH&MMSec*, pages 149–157. ACM, 2021.

Declaration of Own Work

I, Nora Hofer, confirm that this dissertation *The Role of Lossy Compression in Digital Image Forensics* is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

Ich, Nora Hofer, erkläre hiermit am Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit *The Role of Lossy Compression in Digital Image Forensics* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht.

Die vorliegende Arbeit wurde bisher in gleicher oder ähnlicher Form noch nicht als Dissertation eingereicht.

.....	.....
Date	Nora Hofer