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

Nora Hofer
University of Innsbruck
Innsbruck, Austria
nora.hofer@uibk.ac.at

Rainer Böhme
University of Innsbruck
Innsbruck, Austria
rainer.boehme@uibk.ac.at

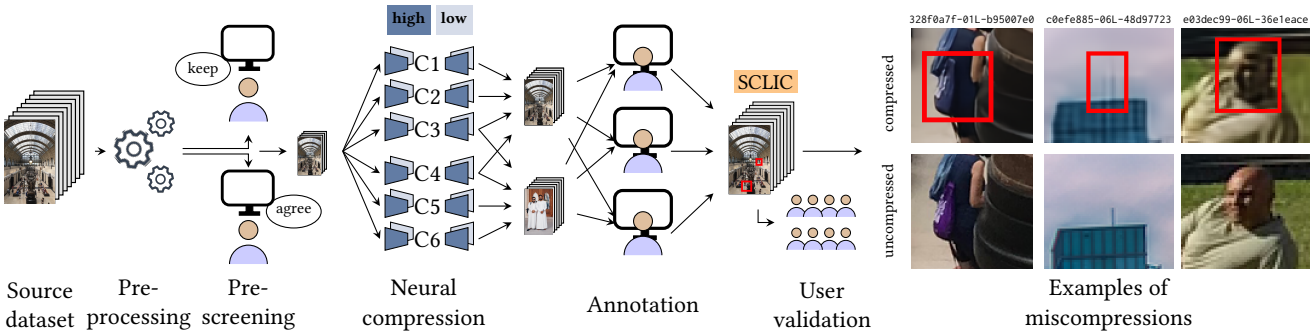


Figure 1: The SCLIC dataset: Human labelers search and annotate reconstruction errors in image details that change the semantic, creating a dataset of 18k uniquely identified *miscompressions*. Each source image was compressed with four (of six) codecs at two quality settings. Images were prescreened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

Abstract

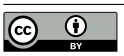
Preserving the semantic integrity of image details is difficult in neural image compression. Failure to do so can result in *miscompressions*: reconstruction errors that change the meaning between the original and reconstructed images. Undetected miscompressions can compromise the reliability of reconstructed 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.

CCS Concepts

• **Computing methodologies** → **Image compression**; **Reconstruction**; **Scene understanding**.

Keywords

Neural image compression, lossy compression, semantic changes



MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3758291

ACM Reference Format:
Nora Hofer and Rainer Böhme. 2025. Challenging Cases of Neural Image Compression: A Dataset of Visually Compelling Yet Semantically Incorrect Reconstructions. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3746027.3758291

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, 23, 31]. However, prior work has pointed out the risk of *miscompressions* [13]. Miscompressions are reconstruction errors in which the semantics of image details change after lossy compression. The examples shown to the right of Figure 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 [23, 31], proposed a taxonomy [13], and documented biases in the shift of semantics [26]. 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 [19]. In this paper we present SCLIC,[1] a dataset of 18 019 annotated miscompressions

[1] Semantic Changes in Learned Image Compression

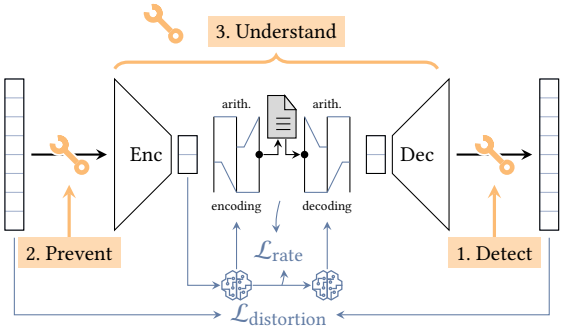


Figure 2: Diagram of a neural compression codec. Wrenches show potential applications of our dataset (*cf.* Sect. 2.3).

identified by three trained human labelers spanning 10 828 images from popular benchmark datasets.

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

The remainder of this paper is structured as follows: Section 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 3 describes our processing and annotation pipeline, access instructions, as well as legal and ethical aspects. Finally, Section 4 reports first insights using statistical analyses of the annotations before Section 5 concludes our paper.

The full dataset, all supplemental material, and download instructions are accessible via: https://zenodo.org/records/16780952.

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 [30] employ a blockwise discrete cosine transform (DCT) [2], followed by quantization using frequency-dependent tables, and entropy coding via run-length and Huffman schemes [14].

Figure 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 [12]. Quantization is commonly done with simple rounding [6, 27]. 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.

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 [33] 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 [23] replaces the decoding network with a generative adversarial network (GAN) conditioned on the hyperprior. Similarly, the CDC codec [31] uses a diffusion variational autoencoder, also conditioned on the hyperprior.

2.2 Miscompressions

Modelling human perception is not trivial and has become an active field of research [9, 11]. 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 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." [13, 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. [26] 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 [8, 15, 22]. The risk of detection errors has been reported especially for iris images [8]. 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 [21].

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 2.

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 [13]. This approach is related to the work of Tsereh et al. [29], 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. [16], 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.

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. [32] 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.

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. [20] 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. [26] 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).

3 Method

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

3.1 Preparation

To generate our dataset we took 2491 uncompressed images from the three benchmark **source datasets** CLIC [28], DIV2K [1], and RAISE [10]. **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 [23]	**low:** *HiFiC-lo*	**high:** *HiFiC-hi*
C4:	STF [33]	**low:** $\lambda = 0.067$	**high:** $\lambda = 0.025$
C5:	CDC xparam 0.9 [31]	**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 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.

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.

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.

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 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 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 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.

Table 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).

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.

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 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

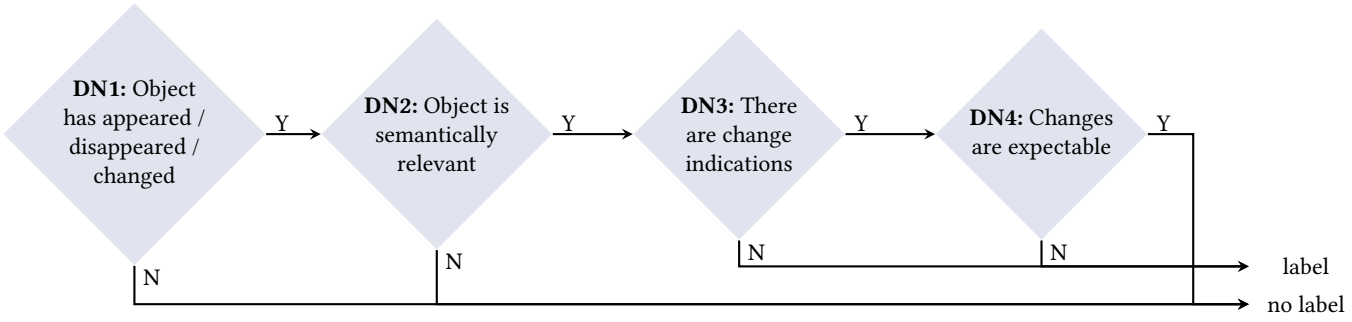


Figure 3: Decision tree to disambiguate what constitutes a miscompression. A modification (DN1) of a semantically relevant object (DN2) is labeled as miscompression if there are *no* visible indications of the modification (DN3) or one would *not* expect the modification given the visible quality of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

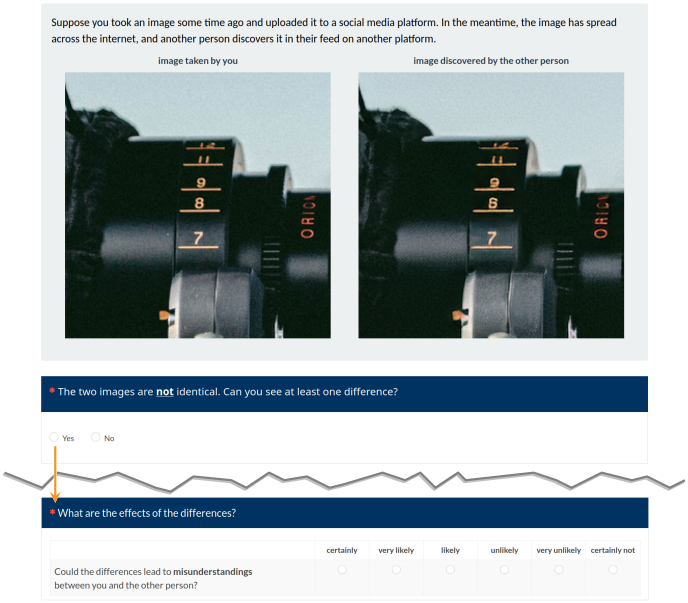


Figure 4: Screenshot of our validation study. The stimulus images stayed on top of both question blocks. In this example, the digit "8" turns into a "6." The second block was skipped if the first question was answered with "no."

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 [24, 25]. We cannot expect a high α in our context as the metric penalizes even mild disagreement in small coder groups, especially with binary decisions [18]. With only three labelers, any deviation from consensus has a strong impact.

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. 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. 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.

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.

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.

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.

Table 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

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 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.

4 Dataset description

Table 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 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

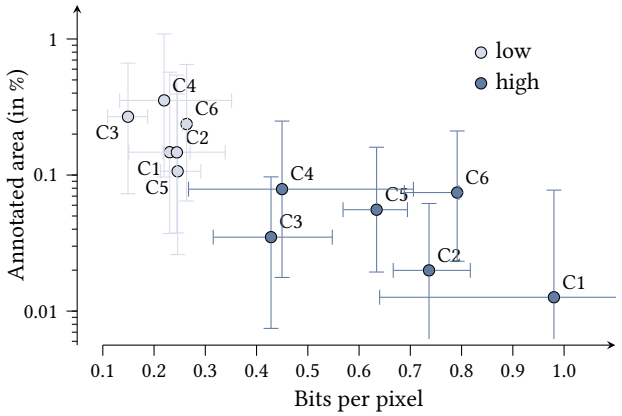


Figure 5: Median and interquartile ranges of the proportion of pixels per image annotated as miscompressed by codec and compression rate. Lower is better. Note the log scale.

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.

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. 3.2.4). Future versions of the dataset could include verifications and attributes, *e.g.*, based on existing taxonomies [13]. 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.

[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 [17]. Miscompressions may affect a broad set of image details, not just digits.

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. 2017. NTIRE 2017 Challenge on single image super-resolution: Dataset and study. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. http://www.vision.ee.ethz.ch/~timofter/publications/Agustsson-CVPRW-2017.pdf

[2] N. Ahmed, T. Natarajan, and K.R. Rao. 1974. Discrete cosine transform. *IEEE Trans. Comput.* (1974), 90–93.

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

[4] J. Anger. 2023. *vpv: Image viewer designed for image processing experts. (v0.8.2).* https://github.com/kidanger/vpv

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

[6] J. Ballé, V. Laparra, and E. Simoncelli. 2016. 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. 2018. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*.

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

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

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

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

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

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

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

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

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

[17] D. Kriesel. 2013. Xerox scanners/photocopiers randomly alter numbers in scanned documents. https://www.dkriesel.com/en/blog/2013/0802_xerox-workcentres_are_switching_written_numbers_when_scanning (accessed: August 2025).

[18] K. Krippendorff. 2011. Computing Krippendorff's alpha-reliability. https://repository.upenn.edu/server/api/core/bitstreams/0421f871-f005-4322-b06a-a66bec328e3b/content (accessed: August 2025).

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

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

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

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

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

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

[25] H. Pardawala, S. Sukhani, A. Shah, V. Kejriwal, A. Pillai, R. Bhasin, A. DiBiasio, T. Mandapati, D. Adha, and S. Chava. 2024. 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*.

[26] T. Qiu, A. Nichani, R. Tadayontahmasebi, and H. Jeong. 2025. 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*. 1862–1889.

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

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

[29] D. Tsereh, M. Mirgaleev, I. Molodetskikh, R. Kazantsev, and D. Vatolin. 2024. JPEG AI Image Compression Visual Artifacts: Detection Methods and Dataset. *arXiv preprint arXiv:2411.06810* (2024).

[30] G. Wallace. 1991. The JPEG still picture compression standard. *Commun. ACM* (1991), 30–44.

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

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

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

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

Nora Hofer
University of Innsbruck
Austria

Rainer Böhme
University of Innsbruck
Austria

ACM Reference Format:
Nora Hofer and Rainer Böhme. 2025. Challenging Cases of Neural Image Compression: A Dataset of Visually Compelling Yet Semantically Incorrect Reconstructions Appendix. In . ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 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`. [1] Table 1 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.

1.1 Source datasets

1.1.1 CLIC [11]. We downloaded images of the *clic2020 v1.0* test- and validation sets, including both mobile and professional.[2] We omitted the v1.0 trainset. The *clic2024* test, and validation set (32 and 29 respectively) was downloaded from the CLIC competition page.[3]

1.1.2 DIV2K [2]. We downloaded images of the *div2k v2.0 HR* train and validation set. [4]

1.1.3 RAISE [6]. We randomly sampled images (from all camera models) of the categories Indoor, People, Objects, and Buildings.[5] 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.

[1] https://usage.imagemagick.org/resize/
[2] https://www.tensorflow.org/datasets/catalog/clic
[3] https://storage.googleapis.com/clic2023_public/
[4] http://data.vision.ee.ethz.ch/cvl/DIV2K/
[5] http://loki.disi.unitn.it/RAISE/

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1.2 Codecs

We selected six compression codecs from the literature. Table 2 gives an overview of their architectures.

1.2.1 C1/C2. The **Hyperprior construction [4]** 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. [6] The hyperprior models were trained on approximately 1 million 256×256 pixel crops of web scraped color JPEG photographs[7], downsampled by a randomized factor to minimum heights/widths between 640 and 1200 pixels from original heights/widths between 3000 and 5000 pixels.

1.2.2 C3. The **High-Fidelity generative image Compression (HiFiC) codec [8]** 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 [12]. The HiFiC models were trained on 256×256 pixel crops of "a large set of high-resolution images" [8, 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`.

1.2.3 C4. The **Symmetrical TransFormer (STF) framework** [16] 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. [8]

1.2.4 C5. The **Conditional Diffusion Compression (CDC) [14]** models are trained on 90k 256×256 pixel crops taken of frames taken from clips of the Vimeo-90k dataset [13], 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.

[6] Available in TensorFlow's `compression` library *v*2.17.0 [5]
[7] Non-photograpic images were detected by saturation levels.
[8] Pretrained models were provided by the authors [15]. Unfortunately, no weights for models optimized for MS-SSIM were available at the time of writing.

Table 1: Summary of source datasets.

Dataset	# available	# considered	# selected	# viewed	Source
CLIC2020 v1.0 Testset (mobile and professional)	428	428	379	336	TF datasets
CLIC2020 v1.0 Trainset (mobile and professional)	1 633	0	0	0	TF datasets
CLIC2020 v1.0 Validset (mobile and professional)	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 (all camera models; categories: People, Indoor, Objects, Buildings)	7 441	500	382	332	loki.disi.unitn.it/RAISE
Total	10 565	2 491	1 563	1 363	

Table 2: Summary of codecs used to compress images included in the dataset.

	Hyper I [4]	**Hyper II [4]**	**STF [16]**	**HiFiC [8]**	**CDC [14]**	**JPEG AI [3]**
Preprocessing	–	–	–	–	–	YUV
Transform	VAE/CNN		VAE/Attention	VAE/CNN	VAE/CNN	VAE/Attention
Entropy distribution	Hyperprior					
Reconstruction	VAE		VAE/Attention	GAN	Diffusion	VAE/Attention
Optimization metric	MSE	MS-SSIM	MSE	MSE, LPIPS	LM, LPIPS	MSE, MS-SSIM

1.2.5 C6. The **JPEG AI reference implementation [3]** 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[9]. 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 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).

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

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.

1.4 Compression

Figure 3 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 3: 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

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 [10].

1.5 Hardware

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

2 Instrument

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 [1]. 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 1. 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.

2.2 Instructions

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.)

2.2.2 Multimiscompressions. Frequently, images contained *multimiscompressions*, *i.e.*, multiple instances of the same miscompression in the same image, as shown in Figure 2. 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.

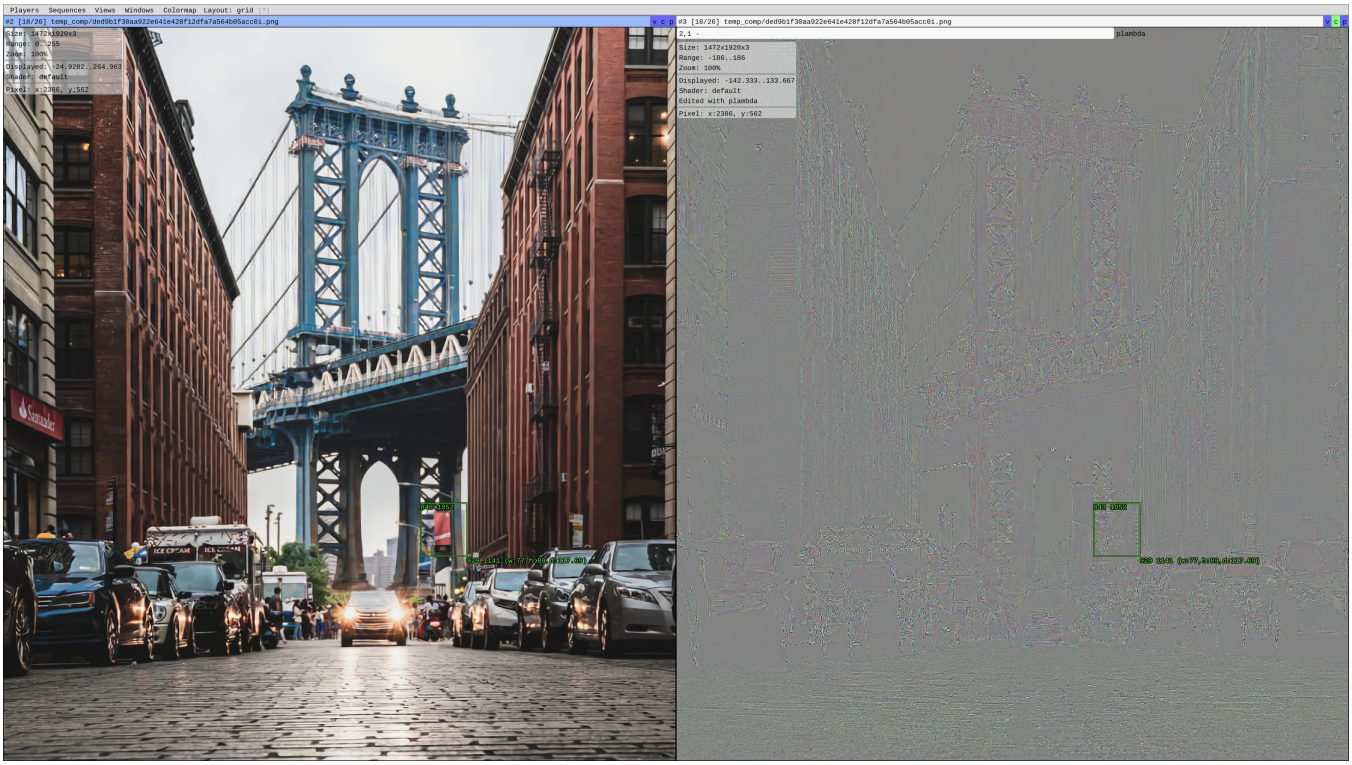


Figure 1: Screenshot of the labeling setup with VPV [1] modified to allow tagging of miscompressed areas. Raters could toggle between the compressed and uncompressed version of the image on the left and view the difference image on the right. Zooming in was possible. The difference image could be hidden.

Figure 2: `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.

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.

2.3 Labeler training

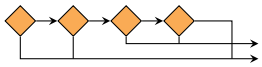
Training included joint labeling sessions and in-depth discussions of individual potential annotations in two training images.
First, the three labelers and the main researcher annotated both images independently. Then, all annotations were discussed in detail. Based on the decision tree, we evaluated whether it constitutes a miscompression, and should be annotated for the dataset. Training images are shown in Figure 3 and Figure 4. The compressed and uncompressed versions are available at https://fileshare.uibk.ac.at/d/cd6d5c7f9d954f18a19a/.

*2.3.1 Image I: Detected changes that **should** be annotated.*

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

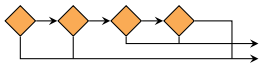
- What: An identifiable object disappeared.
- Semantic relevance: Birthmarks are biometric identifiers.
- Indication: There is no remaining indication of a birthmark left - one could not guess that there was a birthmark.

- 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.

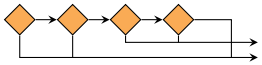


3. Watch face unreadable on right person - right arm

- What: An identifiable object changed.

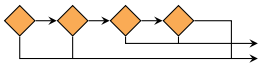
Figure 3: In–depth training image I

- 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.



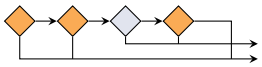
4. Thumb looks scarred on right person - right hand

- 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

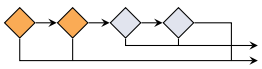
- What: An identifiable object changed.
- Semantic relevance: Nails might have a semantic meaning.
- Indication: One would consider a missing thumbnail.
- Expectation: Given the (local) quality and visible compression artifacts, one would not expect this change. One would not expect this to happen when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



*2.3.2 Image I: Detected changes that **should not** be annotated.*

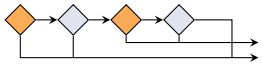
1. Light birth marks or freckles on left person's face missing

- 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.



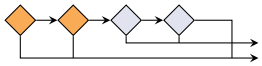
2. Right frame temple on left person's glasses missing

- 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

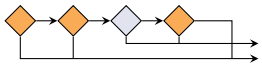
- What: An identifiable object changed.
- Semantic relevance: Might be an indication for illness;
- Indication: One would consider the existence of a reflection.
- Expectation: Given the (local) quality and visible compression artifacts, one might also expect this change.



*2.3.3 Image II: Detected changes that **should** be annotated.*

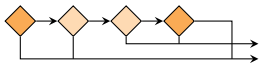
1. Sign "Halteverbot" - left of central house

- 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.

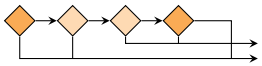


3. Roof shadow - central house

- What: An identifiable object changed.

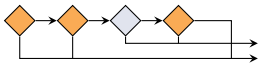
Figure 4: In–depth training image II

- 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.



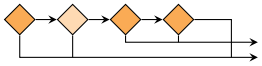
4. Top characters at clock of tower

- 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.



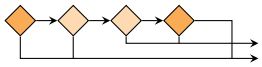
6. Chain in front of red car

- 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

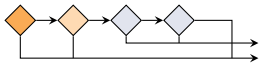
not expect this to happen when using conventional compression algorithms.



2.3.4 *Image II: Detected changes that **should not** be annotated.*

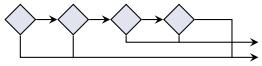
1. Camouflaged poles on the ground, right square

- 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

- What: A non-identifiable object changed.
- Semantic relevance: Non-identifiable, therefore no semantic meaning.



2.4 Inter-labeler agreement

Figure 5: 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.

2.4.1 *Agreement score.*

Figure 5 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.

2.4.2 *Krippendorff's alpha.*

Krippendorff's Alpha [7] 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 5.

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 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.

3 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.

4 Statistical analysis

5 Samples

References

[1] 2023. *kidanger/vpv: Image viewer designed for image processing experts. (v0.8.2).* https://github.com/kidanger/vpv

[2] E. Agustsson and R. Timofte. 2017. NTIRE 2017 Challenge on single image super-resolution: Dataset and study. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition.* http://www.vision.ee.ethz.ch/~timofter/publications/Agustsson-CVPRW-2017.pdf

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

Table 4: Inter-labeler agreement breakdown with 95% confidence intervals

Units	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)

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

[5] J. Ballé, S. J. Hwang, and E. Agustsson. 2024. *TensorFlow Compression: Learned Data Compression*. http://github.com/tensorflow/compression

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

[7] K. Krippendorff. 2011. Computing Krippendorff's alpha-reliability.

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

[9] T. Qiu, A. Nichani, R. Tadayontahmasebi, and H. Jeong. 2025. 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*. 1862–1889.

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

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

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

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

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

[15] R. Zou. 2023. *Googolxx/STF: Pytorch implementation of the paper "The Devil Is in the Details: Window-based Attention for Image Compression*. https://github.com/Googolxx/STF

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

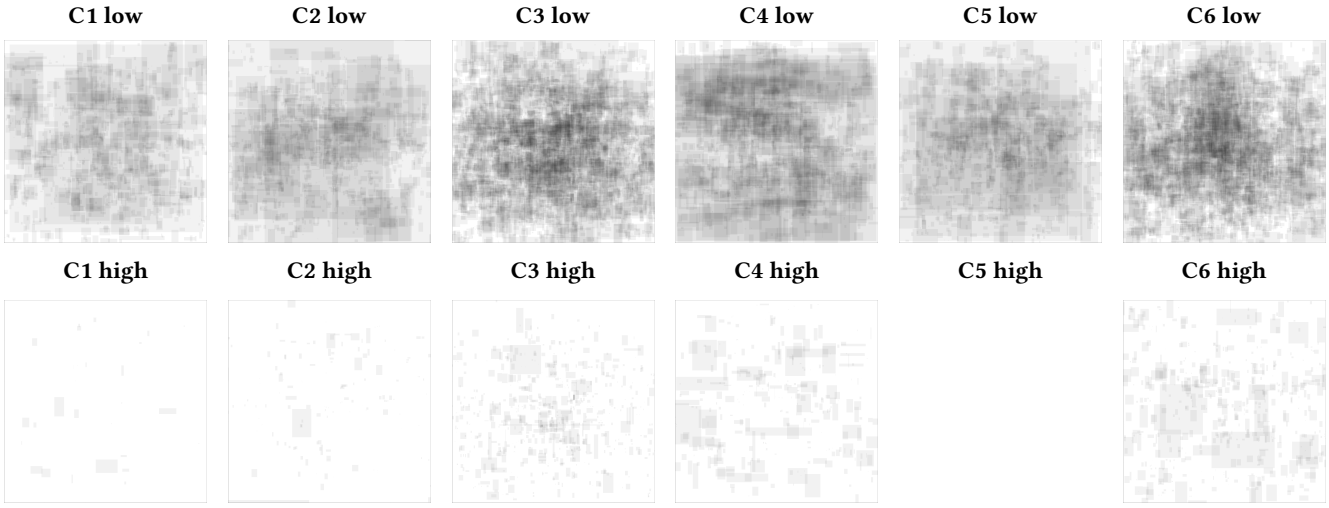


Figure 6: Visualization of the annotated areas superimposed from all images for each codec. The coordinates for images with varying aspect ratios were scaled to unit squares for this visualization.

Table 5: Meta information for the sample miscompressions illustrated in the paper.

Reference	ID	model	description
Figure 2, Figure 13	4154727c-06L-4ffe5f30	JPEG-AI 0.25bpp	Red flags disappear
Figure 8	01f0f60f-03L-38b9cdda	HiFiC lo	House appears demolished
Figure 9	2a7aac4b-03L-41776961	HiFiC lo	Changed eye color
Figure 10	2b73d284-03L-56c37ba9	HiFiC lo	Head disappeared
Figure 11	328f0a7f-05L-6e11a50c	CDC 2048	Statue disappeared
Figure 12	3141bfc1-03L-32cdf7fd	HiFiC lo	License plate is illegible
Figure 14	ab5e3676-02L-6c9e20d7	Hyperppr. MS-SSIM 0.25bpp	Changed direction of currogate
Figure 15	ad865bfc-02H-eaaaac43	Hyperppr. MSE 0.75bpp	Brake lights turn off
Figure 16	b4a7895e-06L-2fe29252	JPEG-AI 0.25bpp	Bench and baby disappear
Figure 1 main paper, Figure 17	c0efe885-06L-48d97723	JPEG-AI 0.25bpp	Hallucinated antenna
Figure 1 main paper, Figure 18	e03dec99-06L-36e1eace	JPEG-AI 0.25bpp	Different skin tone
Figure 1 main paper	328f0a7f-01L-b95007e0	CDC xparam 0.9 $\lambda = 2048$	Changed bag color

Figure 8: `01f0f60f-03L-38b9cdda`.

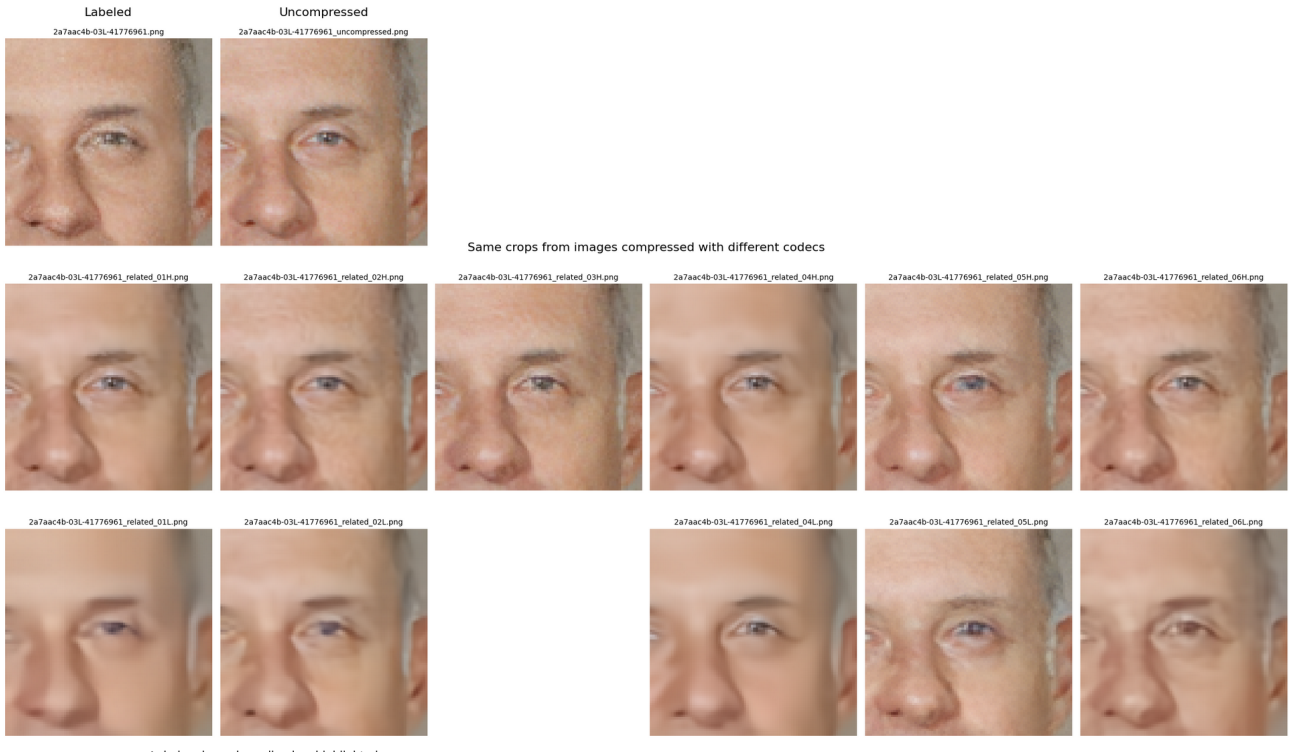


Figure 9: `2a7aac4b-03L-41776961`.

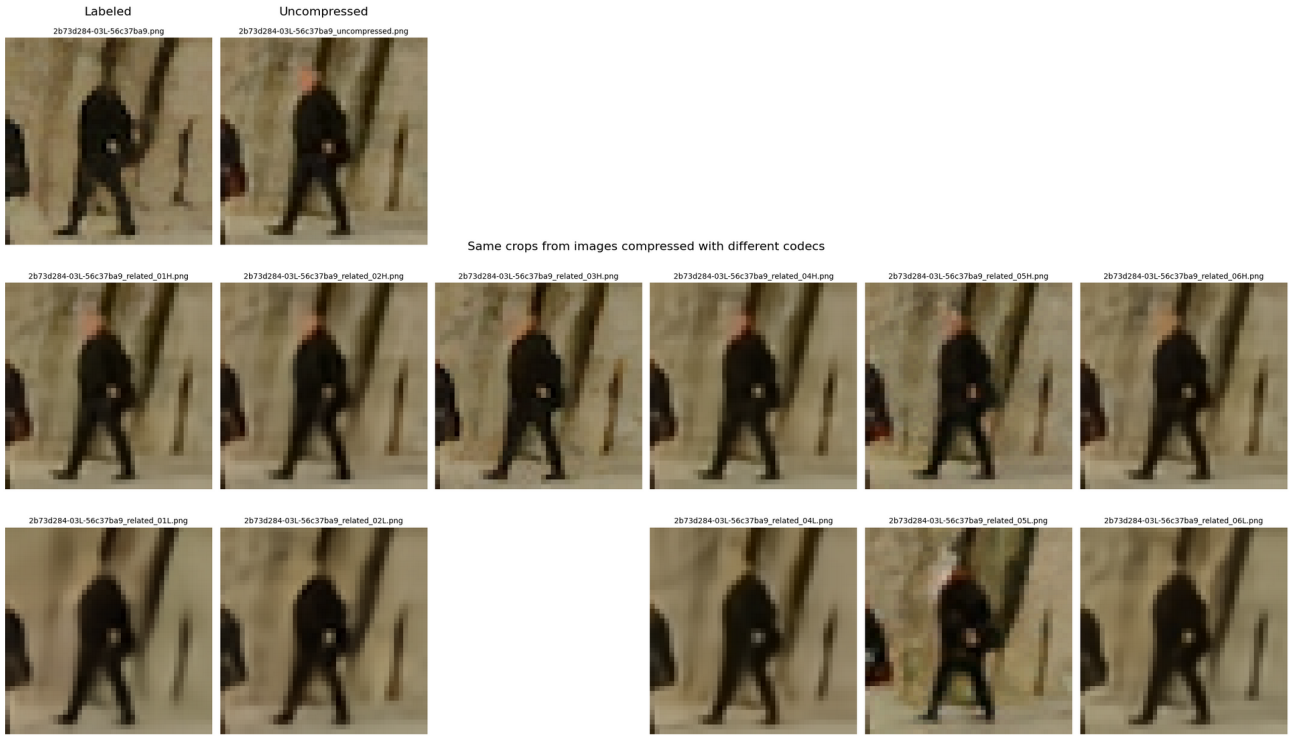


Figure 10: 2b73d284-03L-56c37ba9.

Figure 11: 328f0a7f-05L-6e11a50c.

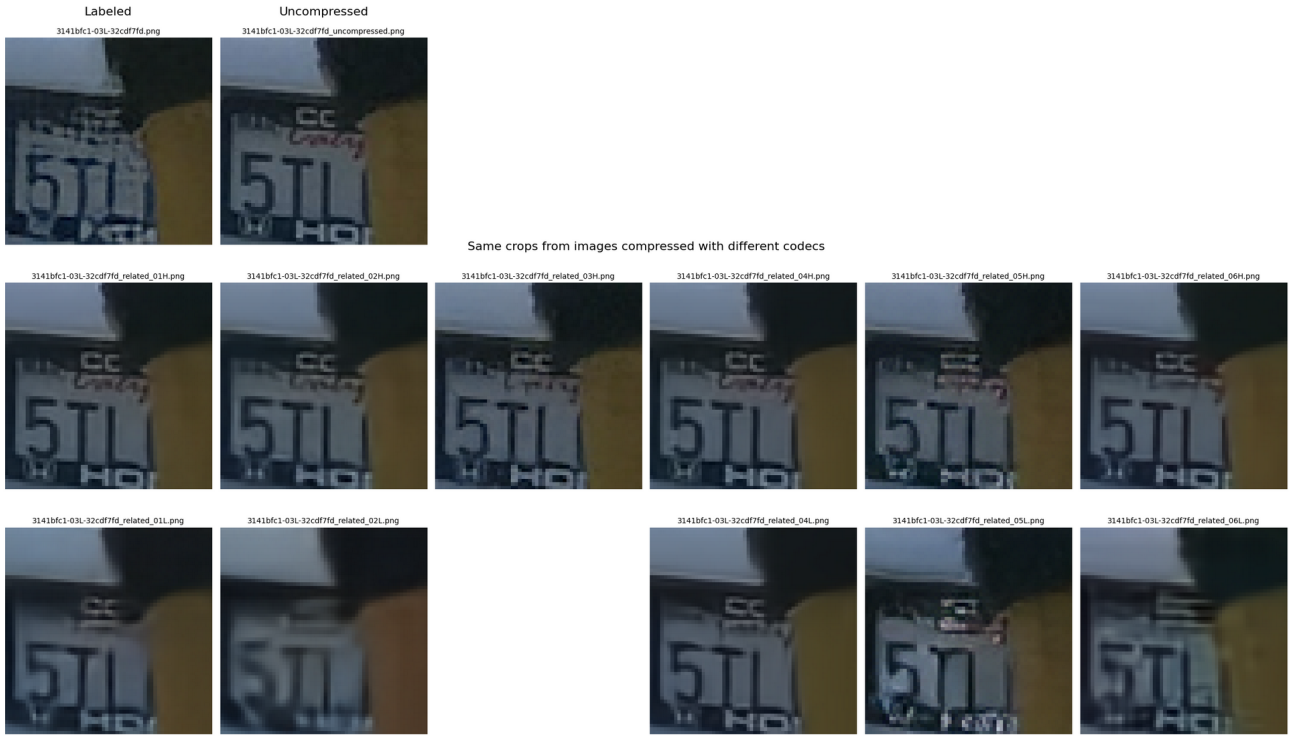


Figure 12: 3141bfc1-03L-32cdf7fd.

Figure 13: 4154727c-06L-4ffe5f30.

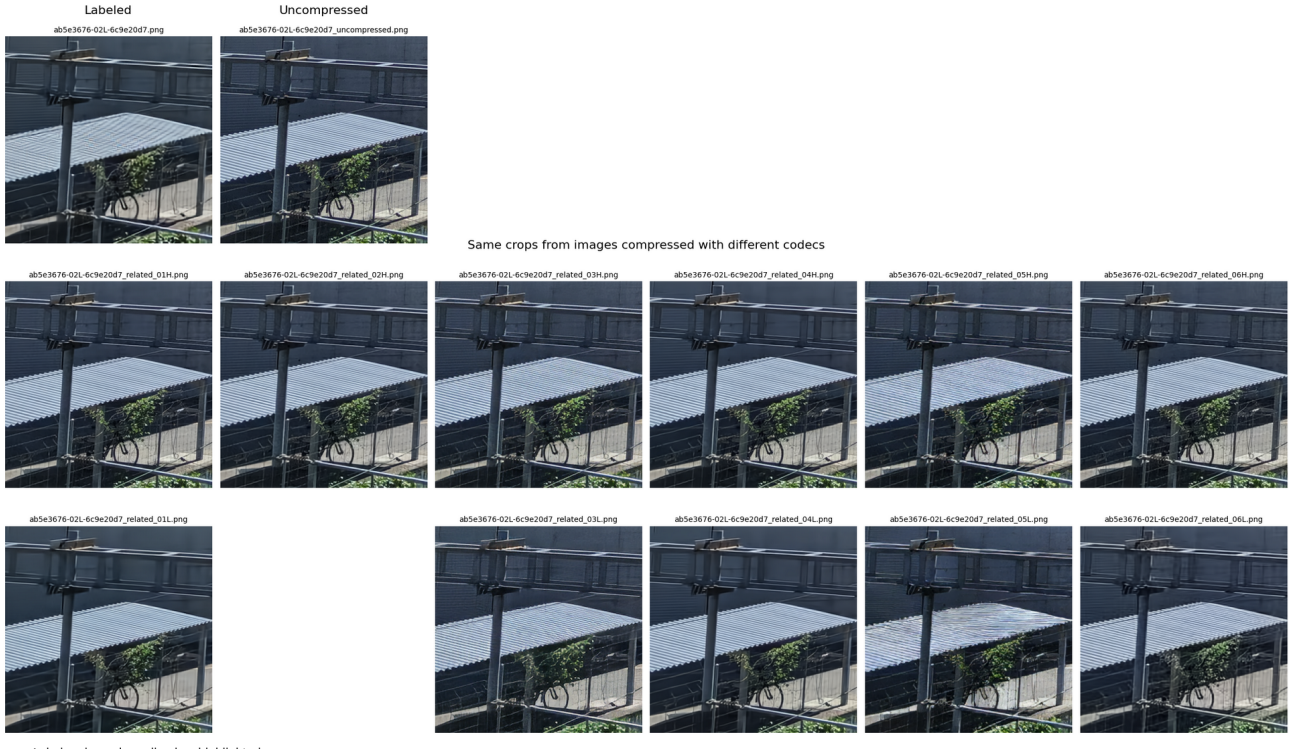


Figure 14: ab5e3676-02L-6c9e20d7.

Figure 15: ad865bfc-02H-eaaaac43.

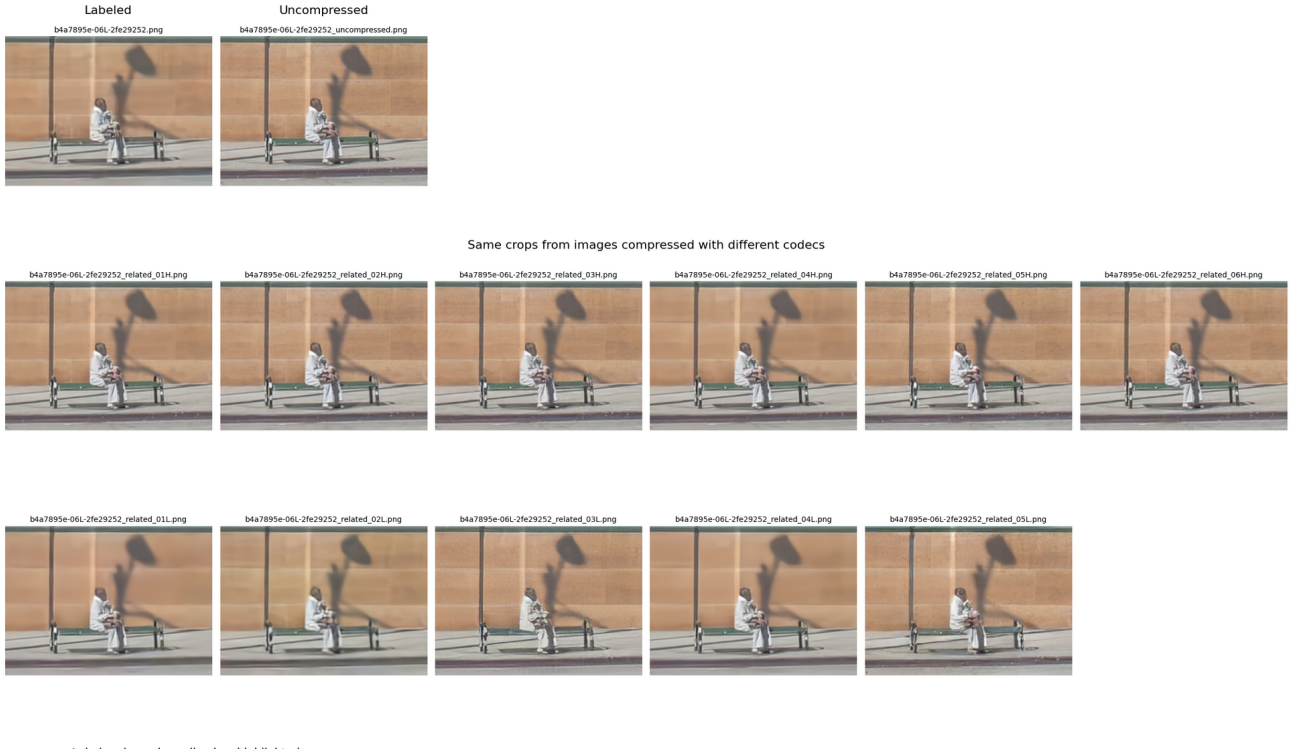


Figure 16: b4a7895e-06L-2fe29252.

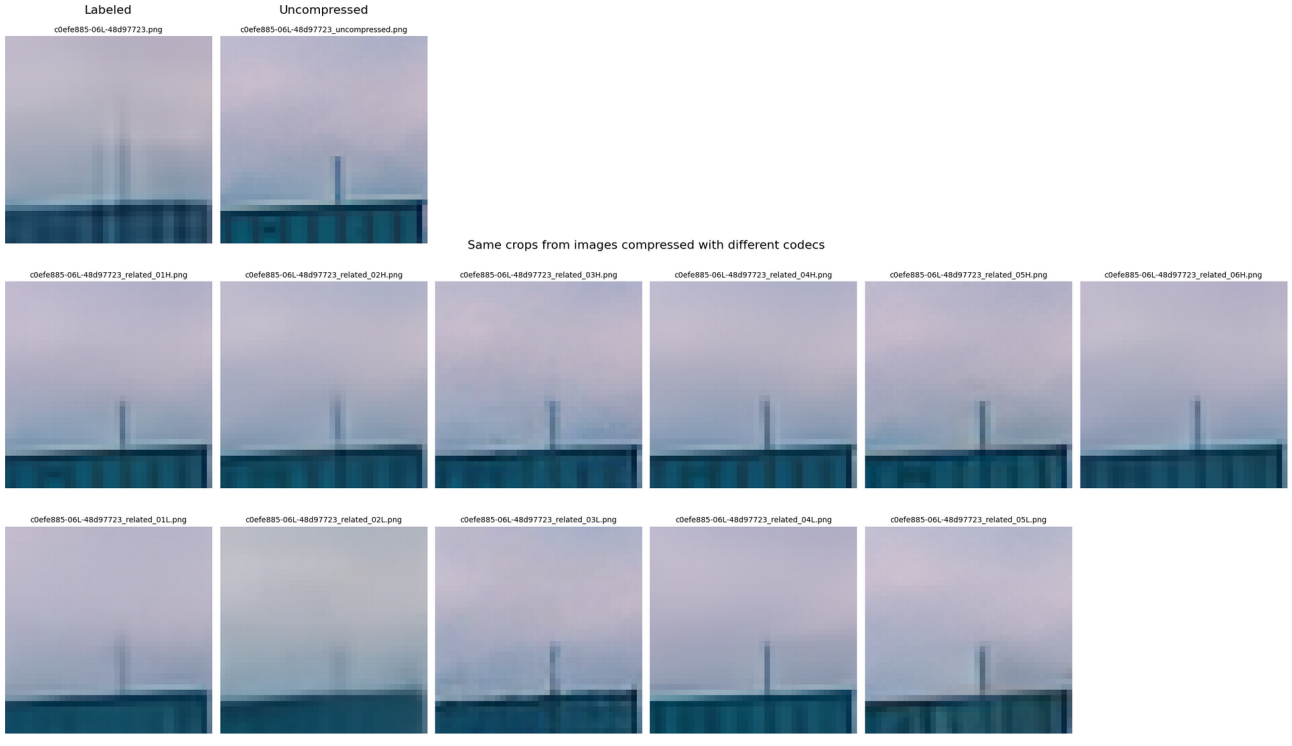


Figure 17: c0efe885-06L-48d97723.

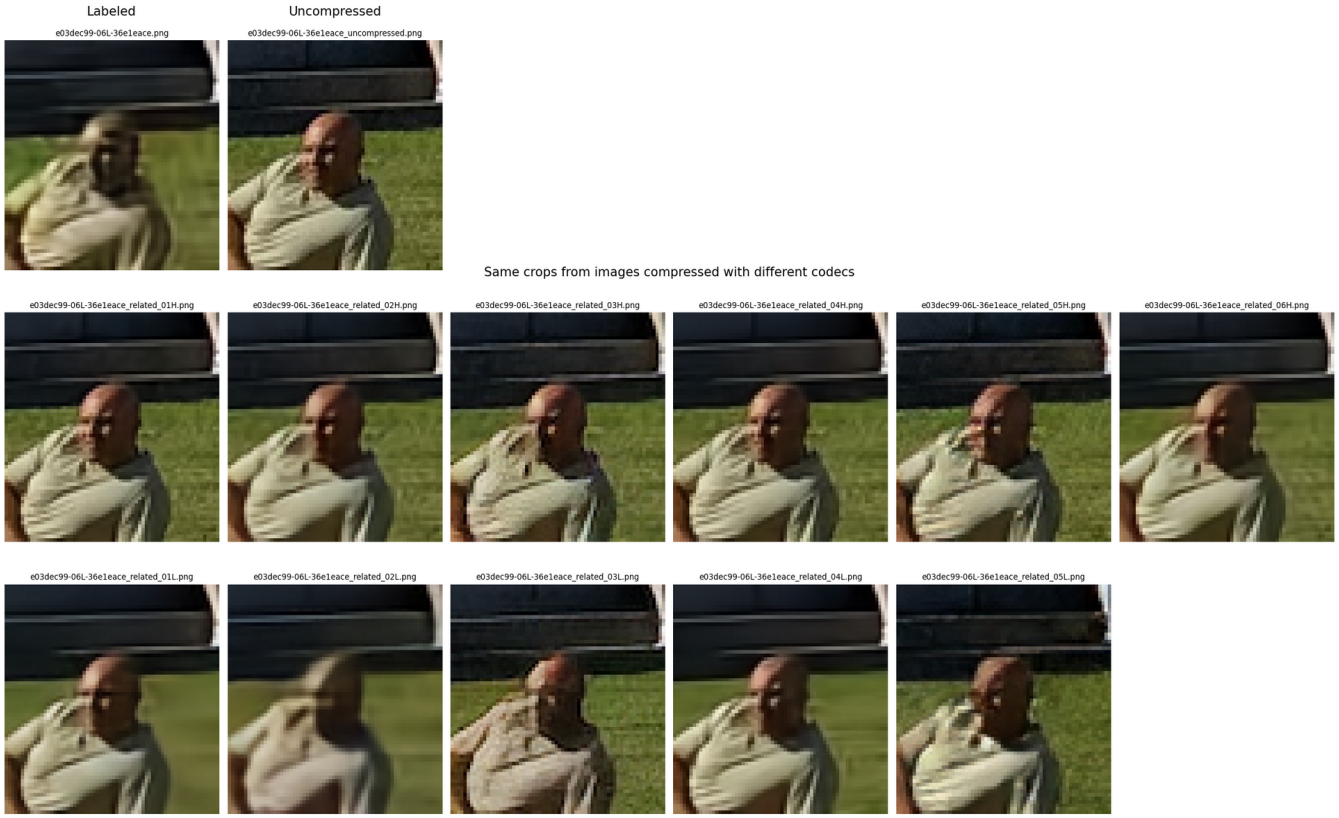


Figure 18: `e03dec99-06L-36e1eace`. Prior work has analyzed racial bias of neural image compression [9].