



# **Landscape More Secure Than Portrait? Zooming Into the Directionality of Digital Images With Security Implications**

Benedikt Lorch and Rainer Böhme, *University of Innsbruck*

<https://www.usenix.org/conference/usenixsecurity24/presentation/lorch>

**This paper is included in the Proceedings of the  
33rd USENIX Security Symposium.**

**August 14-16, 2024 • Philadelphia, PA, USA**

978-1-939133-44-1

**Open access to the Proceedings of the  
33rd USENIX Security Symposium  
is sponsored by USENIX.**

# Landscape More Secure Than Portrait ?

## Zooming Into the Directionality of Digital Images With Security Implications

Benedikt Lorch  
University of Innsbruck

Rainer Böhme  
University of Innsbruck

### Abstract

The orientation in which a source image is captured can affect the resulting security in downstream applications. One reason for this is that many state-of-the-art methods in media security assume that image statistics are similar in the horizontal and vertical directions, allowing them to reduce the number of features (or trainable weights) by merging coefficients. We show that this artificial symmetrization tends to suppress important properties of natural images and common processing operations, causing a loss of performance. We also observe the opposite problem, where unaddressed directionality causes learning-based methods to overfit to a single orientation. These are vulnerable to manipulation if an adversary chooses inputs with the less common orientation. This paper takes a comprehensive approach, identifies and systematizes causes of directionality at several stages of a typical acquisition pipeline, measures their effect, and demonstrates for three selected security applications (steganalysis, forensic source identification, and the detection of synthetic images) how the performance of state-of-the-art methods can be improved by properly accounting for directionality.

## 1 Introduction

Images are *directional* if horizontal pixels sequences have different statistical properties than vertical sequences. Directionality has multiple causes. It can originate from scene content, unintentional asymmetries in image processing, or technological choices that appeal to human visual perception. While the effect of directionality has so far been neglected in security applications that depend on images, we show that directionality impacts the security of image steganography, forensic source identification, and the detection of synthetic images (aka “fake-or-real”).

Figure 1 showcases the loss of not considering directionality in these three applications. The task of *steganalysis* is to attack steganography [92] by detecting hidden messages embedded in innocuous cover images [84]. We embed 0.4

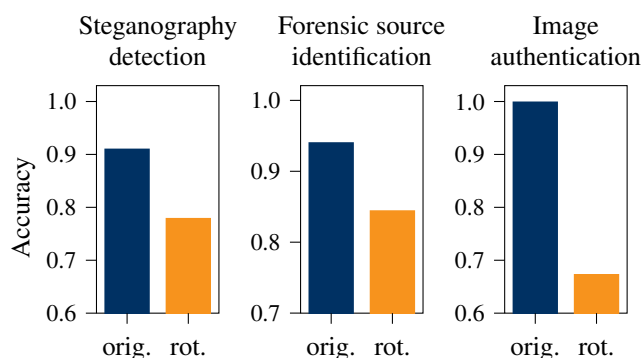


Figure 1: Directionality matters across security applications. The figure summarizes the performance loss of selected state-of-the-art classification tasks when the (square-shaped) test images are rotated by 90 degrees. Note the different scales.

bit per usable sample into JPEG images of size  $512^2$  using J-UNIWARD [49], a state-of-the-art steganography method, resulting in an average payload ratio of 4.3%. The leftmost graph shows the detection accuracy of an EfficientNet-B0 [97] detector with 4 M parameters trained on the dataset from the ALASKA2 steganalysis competition [16]. It achieves 91% accuracy (blue bar) for the binary task of distinguishing between stego images and pristine covers. This accuracy drops to 78% (orange bar) when the test images are rotated by 90 degrees, indicating a lack of robustness to directionality. In terms of error rates, this means that the probability of false accusations increases from 8% to 37%. (See Sec. 4 for more details.) A common task in *forensic source identification* is to identify the camera model that was used to acquire a given image [61, 73]. The center graph shows the accuracy of EfficientNet-B5 trained as multi-class classifier on scenes from the Forchheim database [44], which contains images acquired with 27 different smartphone models. Again, the performance drops from 94% to 85% when the square-shaped test images are rotated by 90 degrees. (See Sec. 5 for details.) Finally, *image authentication* concerns the task of

distinguishing real images from forgeries [28]. An increasingly important subtask is the detection of synthetic images, which has received considerable attention through large-scale challenges including ones hosted by Facebook [26] and the IEEE [22]. The rightmost graph shows the performance of EfficientNet-B0 trained on the binary classification task of distinguishing between real images from the ALASKA2 dataset and generated images using the Stable Diffusion XL [81] autoencoder. Once again, the accuracy drops significantly from 100 % to 67 % if the orientation of the test images does not match the training images. (See Sec. 6 for details.) In all examples, we replicate a common experimental setup and train a state-of-the-art CNN detector. The results reveal that images contain directional statistics, which learning-based detectors tend to exploit. This calls for a spotlight on the effect of directionality in security applications.

To give one example for a typical cause of directional artifacts, consider JPEG compression [79]. While this popular [27, 50] lossy compression standard does not favor any direction, common parameter choices do leave directional artifacts. These include the practice to subsample the chroma channels in the horizontal direction only, or the asymmetry in the default quantization table. Some practices can be explained with technological requirements (e.g., interlacing on CRT monitors [56]); others originate from empirical measurements of human visual perception (e.g., contrast sensitivity by spatial frequency [71]). Strikingly, these historically justified practices prevail in contemporary cameras and software libraries. Multiple causes of directionality may coexist, interact with directional biases in the scene content, and therefore require careful analysis. There is no single measure of directionality and causal relationships are not always straightforward.

**Contributions** This paper takes a comprehensive approach to studying directionality of images in security applications. It makes the following specific contributions:

- We identify and systematize causes of directionality and relate them to technology and human perception.
- We present evidence for the effect of directionality in selected security applications: steganalysis, forensic source identification, and the detection of synthetic images.
- We show how the state of the art in each application can be improved by correctly accounting for directionality.
- We extract guidance on the use of rotation augmentation, which has the potential to generalize across applications.

**Implications** Our work has at least three important implications. First and foremost, if security techniques are biased towards one direction, then the opponent can have an advantage by choosing images of the opposite orientation. Second, defenders who want to consider directionality face the decision to incur higher cost and complexity (e.g., in terms of

trainable weights) or accept limited generalizability across image orientations. This applies in particular to deep learning-based methods, but also to hand-crafted features (as we show in Sec. 5). Lastly, some causes we identify are due to legacy decisions made for technology that is no longer in use (e.g., CRT monitors). Standardization bodies and developers of image processing libraries should revisit these conventions.

**Organization** Section 2 discusses how directionality can be measured. Section 3 systematizes the causes of directionality and relates them to technology and human perception. The three following sections evaluate the effect on selected security applications: steganalysis (Sec. 4), forensic camera model identification (Sec. 5), and synthetic image detection (Sec. 6). Section 7 reviews related work. Section 8 discusses implications and limitations, and Sec. 9 concludes.

## 2 Measuring directionality

Tools to detect directional artifacts in images and, where possible, to quantify their strength are a prerequisite for evaluating the effects of directionality. Directional artifacts occur at different stages of the image acquisition pipeline and manifest themselves in different statistical properties. To deal with this variety, we use several methods to measure directionality. We discuss them below, from coarse to fine.

**End-to-end effect** The three security applications mentioned in the introduction use a learning-based detector with measurable accuracy. An intuitive measure of directionality is to evaluate the detector's sensitivity to rotated test images. If the images were non-directional or directionality did not matter, the detector would naturally generalize to rotated test images. A significant performance drop indicates that the detector is sensitive to directional image statistics, and the magnitude of the drop can be interpreted as effect size.

An advantage of the end-to-end approach is that it measures the effect of directionality for the target application. As learning-based detectors tend to exploit small nuances in image statistics, this approach can pick up subtle directional artifacts, which are difficult to measure directly with other tools. At the same time, the indirect measurement through a specific detector has several limitations. A drop in performance is hardly comparable across applications. Moreover, this approach cannot reveal the cause of directionality.

**Visualization** Statistical patterns in single images are often difficult to see with bare eyes. A common technique in the forensics community is to visually inspect the power spectrum, where periodic patterns manifest themselves as peaks in the frequency domain [82]. Figure 2 shows an example image (left) and its power spectrum (center left). Horizontal motion

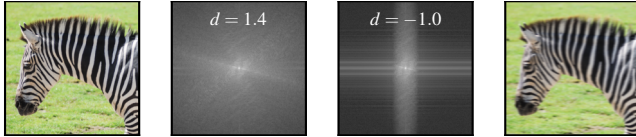


Figure 2: The power spectra (center panels) reveal that the right image has low energy in the high horizontal frequencies. (Motion blur has been applied for illustration.)

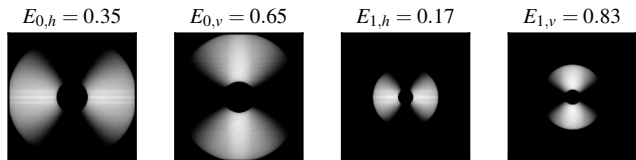


Figure 3: The steerable pyramids directionality score compares the horizontal and vertical energy on two scales.

blur (right) removes high horizontal frequencies, as can be seen in the corresponding power spectrum (center right).

Directional artifacts that are common and synchronous in all images from a particular source or processing operation can be amplified by averaging many images with diverse content. To further suppress the scene content, one can extract a noise residual. The noise residual is obtained by subtracting the image from a denoised version of it. We use the denoising CNN (DnDNN) [107] with pre-trained weights from [20]. Denoising and averaging can also be combined.

Visualization has the advantage that directionality can be visually perceived and its orientation can be determined. Limitations are that subtle artifacts may not be perceptible, and that it is not easy to quantify the strength of directionality.

**Quantification by steerable pyramids** Ideally, we seek to quantify the directionality of an individual image. A numeric directionality score should have three properties: (1) a score of zero means that directional statistics are unchanged when the image orientation is changed; (2) the sign of the score should indicate the dominant image direction; and (3) the absolute value of the score should reflect the strength of directionality.

The *steerable pyramids* method [93] has these properties. It decomposes the image into several oriented frequency bands at different scales, as shown in Fig. 3. For each scale  $s$ , it calculates the difference in energy between the two horizontal wedges  $E_{s,h}$  and the two vertical wedges  $E_{s,v}$ , normalized by their sum. The resulting scores are summed over all scales. Using two scales gives a directionality score in the range  $[-2, 2]$ . A positive score indicates that the image contains more energy in its horizontal frequencies than in its vertical frequencies. Note that summing over the scales may cancel out directional differences at different scales. Summing over absolute values would avoid this problem, but loses the sign of the dominant image direction.

We also experimented with alternative measurements. An intuitive approach is to compute directional edges using first-order derivative filters, such as the Sobel operator [94]. A directionality score can be defined as the difference between the horizontal and vertical absolute-valued derivative image. We found that both detectors can indicate the dominant image direction, but the steerable pyramids are more precise. The comparison is reported in Appendix B.

Steerable pyramids are applicable to individual images and produce a numerical score  $d$ , which can be compared to other measurements. For example, the left image in Fig. 2 has a directionality score of  $d = 1.4$ . The prevalence of horizontal frequencies corresponds to the Zebra’s vertical stripes. The right image has a score of  $d = -1.0$ , because the horizontal blur attenuates the horizontal frequencies and adds vertical frequencies. A limitation is that the method aggregates over the entire spectrum and is therefore less sensitive to directional artifacts in narrow subbands. By design, it cannot detect higher-order dependencies with directionality, e.g., pairwise correlation between horizontally (but not vertically) adjacent pixels that only occurs in certain environments. Only the learning-based detectors of the end-to-end approach are able to exploit this. This justifies our combination of approaches.

### 3 Causes of directionality

Directionality in images has multiple causes. It can be inherent to the image content or introduced at several stages during the image acquisition, as illustrated in Fig. 4. During image acquisition, the scene is projected through lenses onto a sensor. The sensor turns the incoming light into an electrical current, which is amplified and converted to a raw digital signal. The signal then undergoes a number of raw-to-image conversion steps including demosaicing, denoising, sharpening, color adjustments, gamma correction, quantization, and optionally lens correction. The final image is usually stored in a compressed format, with JPEG still being most popular [27, 51, 78]. All stages can introduce directional artifacts.

The following subsections walk through these stages and explain causes of directionality. Each subsection starts with an observation in a gray box, then presents links to technology, human perception, and prior evidence. At the end of each subsection, we present our own evidence in support of hypotheses derived from previous work. Appendix A gives an overview of the image sources used for the validation experiments.

#### 3.1 Scene content

**Scene content.** Edges in natural images are predominantly horizontal and vertical. In popular image datasets, horizontal edges are more prevalent than vertical edges.

**Background** Most photographs show visual recordings of our world and therefore inherit the orientation characteristics

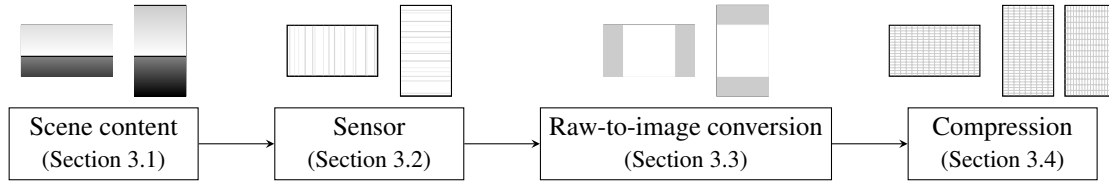


Figure 4: Causes of directionality are present in several stages of the image acquisition.

of the environment. Since the 1970s, researchers have measured the distribution of oriented contours in indoor, outdoor, and natural scene environments. They have found that contours near the cardinal axes, i.e., vertical and horizontal edges, are more prevalent than contours with oblique orientation angles [18, 96]. This holds especially for urban outdoor and indoor scenes, but also for natural scenes. Extensive analyses of horizontal vs. vertical differences support the finding that horizontal lines are more prevalent than vertical lines [45, 58].

The prevalence of cardinal contours is also reflected in human perception. Neurophysiologists have found that humans perceive horizontal and vertical lines more readily than oblique lines. This is known as the *oblique effect* [3]. This bias can be linked to the physiology of the visual cortex, which has more cells processing horizontal and vertical than oblique information [13]. Other work discusses the interaction between the living environment and the development of visual perception. The so-called carpentered world hypothesis suggests that people in urban environments develop stronger preferences for cardinal lines than people from rural areas [89]. Largely independent of the living environment is the visual perception of human faces. A recent study presented filtered versions of human faces to subjects in fMRI brain scanners. It found that stimuli with horizontal frequencies activated the primary visual cortex more than vertical or oblique frequencies [41]. (Examples of the stimulus material are in Appendix F.)

The prevalence of cardinal contours is also present in the art. Analyses of 20th century paintings show that artists prefer horizontal and vertical over oblique lines, which may suggest that human observers find cardinal lines more aesthetically pleasing [66]. In particular, paintings in portrait format contain more vertical lines while paintings in landscape format contain more horizontal lines [66].

**Validation** We analyze the relative distribution of horizontal and vertical scene content in the BOSSBase, ALASKA2, and ImageNet databases using steerable pyramids. Figure 5 shows the distribution of directionality in the three databases. The directionality score  $d$  on the x-axis ranges between  $-2$  and  $+2$ . Images with negative  $d$  contain more horizontal edges, images with positive  $d$  contain more vertical edges. While all databases contain diverse images, the distribution is skewed towards images with horizontal edges. These findings are consistent with related work [45]. The number of images

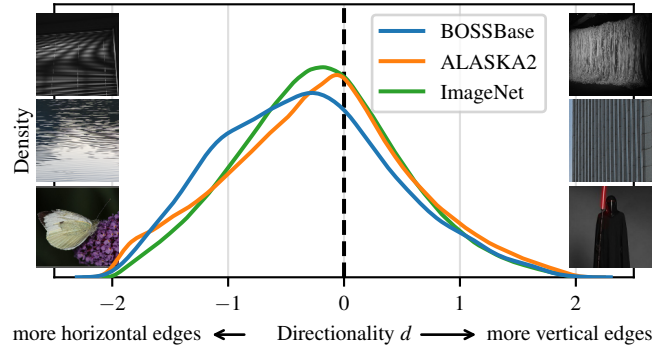


Figure 5: The relative distribution of horizontal and vertical scene content is skewed towards images with more horizontal edges. On the sides are example images from the three databases with very high absolute directionality scores.

with  $|d| > 0.5$  ranges between 49% and 57%. Through manual inspection, we observe that BOSSBase contains many seascapes with a clear horizon and horizontal edges.

### 3.2 Sensor noise

The sensor consists of a rectangular array of photo diodes, which convert incoming photons to an electric charge. The charges are collected, read from the sensor, amplified, and eventually converted to a digital signal [34]. Imperfections in this process introduce noise into the resulting output image. The exact sources of noise and its visual manifestation depend on the sensor design, which is usually proprietary. While some sources of noise cause barely perceptible white noise, others lead to banding, which is more perceptible to the human eye. To give one example, we illustrate how the readout of CMOS sensors produces column patterns [2, 37].

**Readout noise.** Differences between individual column amplifiers in CMOS sensors produce a column pattern.

**Background** In CMOS sensors, each individual cell converts the charge to a voltage. The voltages are then read line by line after shifting the charges through the column amplifier. Deviations in the column amplifiers introduce a column pattern. This column pattern can be best seen from bias frames.

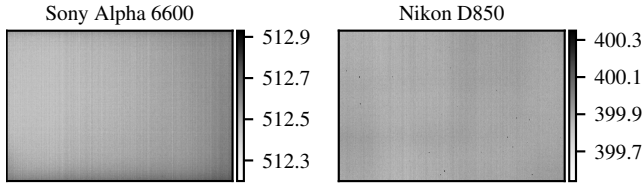


Figure 6: Physical differences between column amplifiers in CMOS sensors produce column artifacts.

**Validation** To illustrate the column pattern, we recorded 200 raw images with a Sony Alpha 6600 and a Nikon D850 camera in a dark room with the lens cap on. Both cameras save 14 bits per pixel. The exposure time was set to the minimum of  $1/4000s$  and  $1/8000s$ , respectively. The short exposure time keeps the influence of dark current low and therefore allows us to measure the readout noise only. Figure 6 shows the average over the 200 frames. The column pattern is clearly visible in the average frame, however, its amplitude is rather small. The values fluctuate around 512 and 400, which are the two cameras’ black levels.

### 3.3 Raw-to-image conversion

**Linear pattern.** Camera model-specific processing steps leave characteristic horizontal and vertical traces.

**Background** The raw sensor measurements undergo a sequence of processing steps to produce a final image. They include color filter array (CFA) demosaicing, tone mapping, white balancing, denoising, sharpening, and often compression. While the exact process in commercial cameras is proprietary and thus difficult to analyze, its effect can be observed by averaging many frames of the same camera model.

The sensor together with the raw-to-image conversion introduce a so-called linear pattern, which can be obtained as a byproduct of the camera fingerprint estimation. Camera fingerprints are estimated by averaging noise residuals from dozens of images, followed by zero-meaning the rows and columns separately for each channel in the color filter array, and then applying a Wiener filter [34]. The difference between the average noise residual and the final fingerprint gives the linear pattern [15]. It contains artifacts which are shared by cameras of the same model, unlike the fingerprint, which is intended to uniquely identify each sensor [29].

**Validation** The flat-field frames in the VISION database are particularly suited for studying linear patterns of different cameras [91]. Figure 7 shows four selected examples. Observe that the Apple iPhone 4 leaves strong horizontal banding artifacts. Apple’s iPhone 6+ shows an oval shape in the center. The Huawei P9 shows irregular vertical structures, while the Samsung Galaxy Tab A leaves strong vertical

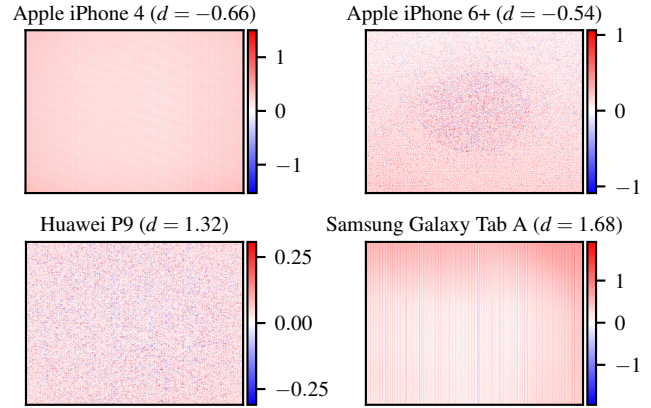


Figure 7: Spatial representation of linear patterns and directionality score  $d$  of four cameras in the VISION database.

banding artifacts. The directionality score  $d$ , calculated with steerable pyramids, confirms the visual impression. Although it is difficult to pinpoint the exact causes in these proprietary devices, these artifacts contribute to directional statistics in final images. We conjecture that the oval shape of the iPhone 6+ might be related to the correction of geometric lens distortion, another cause of directional artifacts which we discuss next.

**Geometric distortion correction.** Correcting geometric lens distortion introduces resampling artifacts, which are strongest at the longer ends of rectangular images.

**Background** An optical lens focuses incoming light onto the camera sensor. An ideal lens would map straight lines onto straight lines. Most practical lenses are not ideal. Their magnification changes with the distance from the optical center [46, Ch. 6.3]. As a result, straight lines appear curved. We distinguish barrel from pincushion distortion, where the magnification decreases or increases with distance from the optical axis, respectively. While this geometric distortion is radial (i.e., non-directional), rectangular sensors capture more distortion at the longer ends of the frame. This raises the question why most cameras have rectangular sensors. For example, DSLR full frame sensors typically measure  $36 \times 24$  mm. This 3:2 aspect ratio has likely persisted from the 35 mm film.

A simple calculation also shows that rectangular sensors are more economical when photographers prefer rectangular aspect ratios.<sup>1</sup> Lenses produce a spherical projection. Since spherical sensors are difficult to manufacture, a square shape would maximize the area within a spherical projection. However, cropping the square frame to the desired rectangular

<sup>1</sup>Rectangular formats can be related to artistic rules, like the rule of thirds, and the golden ratio. In historical paintings, where artists were allegedly free to choose the aspect ratio, the majority of paintings are rectangular rather than square. <https://blog.wolfram.com/2015/11/18/aspect-ratio-s-in-art-what-is-better-than-being-golden-being-plastic-rooted-or-just-rational-investigating-aspect-ratios-of-old-vs-modern-paintings/> (accessed on 2 Feb, 2024).

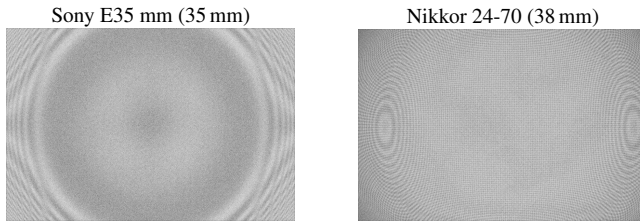


Figure 8: Algorithmic correction of geometric lens distortion may introduce directionality as it treats the long ends of rectangular frames differently than the center area. The figure illustrates this by showing the amount of local correlation for each pixel (brighter means stronger). See text for details.

shape would imply losing resolution. Directly fitting a 3:2 rectangle into the spherical lens projection gives 38% more area than fitting a square into the same size and cropping to the 3:2 format afterwards.

Most cameras accept geometric distortion on the optical side and correct for it using signal processing. This involves unwarping the image, a geometric transformation that requires interpolation [28, Ch. 4.2]. Interpolation, in turn, leaves statistical measurable correlations between adjacent pixels [82].

**Validation** For illustration, we take two raw flat-field images with a Sony E 35 mm prime lens and a Nikkor 24–70 zoom lens at 38 mm, and correct for geometric distortion in Adobe Lightroom Classic 12.3. The corrected images are analyzed with a forensic resampling detector [82]. Figure 8 shows the resulting  $p$ -maps. Brighter shades in  $p$ -maps mean that pixels can be better predicted from their local neighborhood. Known variations in local predictability can be exploited for more reliable forensics and steganalysis [64].

### 3.4 JPEG compression

JPEG is still the most popular image format supporting lossy compression [27]. It achieves its storage efficiency by discarding visually unimportant information. JPEG compression proceeds as follows [79]. RGB pixel values are converted to the luminance and chrominance color space. The chrominance channels are optionally subsampled. Every channel is split into non-overlapping  $8 \times 8$  blocks, each of which is transformed to the frequency domain using the discrete cosine transform (DCT). The resulting DCT coefficients are quantized by division with subband-specific quantization factors and subsequent rounding. The 64 quantization factors form a quantization table (QT), which can be different per channel, and is typically derived using the quality factor (QF) parameter in the (useful) range between 50 (low quality, small files) and 100 (all factors set to 1, high quality, but still not lossless). The quantized DCT coefficients are stored using lossless entropy encoding. JPEG decompression works in reverse order. Information discarded after rounding is not recoverable.

Although the standard does not introduce any directional artifacts, we find several sources of directionality in common parameter choices and in popular implementations.

**Chroma subsampling.** Horizontal chroma subsampling is more popular than vertical chroma subsampling. Moreover, the popular libraries *libjpeg-turbo* and *MozJPEG* introduce horizontal high-frequency artifacts.

**Background** Human perception is less sensitive to changes in color than to changes in brightness [87, Sec. 4.3.4, p. 221]. JPEG compression exploits this property by optionally reducing the resolution of the chroma channels. There are several common subsampling factors, usually denoted in J:a:b notation [83, Ch. 10]. Here it is sufficient that 4:2:0 halves the chroma channel resolution in both directions, i.e., to  $1/4$  of the pixels; horizontal subsampling (4:2:2) halves the horizontal resolution while keeping the vertical resolution; and vertical subsampling (4:4:0) halves the vertical resolution while keeping the horizontal resolution.

Among the directional parameter choices, horizontal subsampling (4:2:2) is more common than vertical (4:4:0). This is probably due to the use of interlacing on cathode ray tube (CRT) monitors, and in accordance with the ITU-R BT.601 recommendation in favor of horizontal chroma subsampling [54]. Interlacing was adopted in the 1930s to double the frame rate without increasing the bandwidth requirements, thereby reducing temporal flickering from low frame rates [11, pp. 426–427]. Interlacing splits the frame into two fields with alternating rows. Only one field is updated at a time. Decompressing a horizontally subsampled field is easy by replicating the previous chroma sample in a row. By contrast, decompressing and displaying a vertically subsampled frame is more involved. As CRT monitors update row by row, additional memory would be required to buffer a row of chroma samples.<sup>2</sup> These are all historical reasons. Today’s displays do not use interlacing, memory has become cheaper, and bandwidth has multiplied.

*Libjpeg* [52], the open-source reference implementation, introduced a high-frequency horizontal artifact during chroma subsampling caused by integer rounding [72]. This artifact, called *chroma wrinkle*, was found in version 6b, which has been in use for about a decade (1998–2009) [7]. From version 7 onwards, *libjpeg* resolved this problem by changing the chroma subsampling implementation to DCT scaling. Yet, the artifact persists until today in popular forks like *libjpeg-turbo* [68], which uses SIMD, and *MozJPEG* [77], which is optimized for web publishers [47].

**Validation** We analyze the distribution of chroma sampling patterns in native camera images from forensic image

<sup>2</sup>Personal communication with Eric Bourguignat of (formerly) CCETT, a research center in Rennes, France, that contributed to the JPEG standard.

Table 1: Number of cameras in forensic image datasets split by their use of chroma subsampling. Cameras in the “mix” column produce JPEGs with 4:2:0 and 4:2:2 subsampling.

Database	4:4:4	4:2:0	4:2:2 (directional)	mix
Dresden	–	1	26	–
VISION	–	32	2	1
Forchheim	–	21	5	1

databases. The Dresden database [39] contains JPEG images from DSLR and point-and-shoot cameras. The VISION [91] and Forchheim databases [44] contain smartphone photos. As shown in Tab. 1, 26 out of 27 cameras in the Dresden database use 4:2:2 subsampling. The only exception is the Kodak M1063, which uses 4:2:0 subsampling. In the VISION and Forchheim datasets, 4:2:0 subsampling is more prevalent. Still, some smartphones produce 4:2:2 subsampling.

*Libjpeg* and its derivatives include the command line tool `cjpeg` for compressing images. Interestingly, *MozJPEG*’s `cjpeg` chooses the chroma subsampling based on the given quality factor. For  $QF \geq 90$ , *MozJPEG* disables chroma subsampling (4:4:4), for  $80 \leq QF < 90$ , it subsamples horizontally (4:2:2), and for  $QF < 80$ , *MozJPEG* defaults to 4:2:0.

**DCT implementation.** The integer DCT implementations in *libjpeg* introduce directional rounding artifacts.

**Background** Since the 2-D DCT is linearly separable, it is commonly implemented as a row-wise 1-D DCT followed by a column-wise 1-D DCT [1]. *Libjpeg* supports three DCT methods. The default, called ISLOW, converts fractional constants to 13-bit integers and uses a fixed-point Loeffler–Ligtenberg–Moschytz (LLM) algorithm, which saves multiplications by using a divide-and-conquer strategy [69]. Fixed-point arithmetic requires descaling (bitwise shifts to the right) followed by rounding.<sup>3</sup> To maintain precision, this happens only after the second (column-wise) DCT. A property of the LLM algorithm is that the 1st and 5th element of the row vectors are not multiplied by fractional constants. These are upscaled by only 2 bits after the first pass, which is why the final descaling leaves stronger rounding artifacts in the 1st and 5th row, but no artifacts in the 1st and 5th elements.

The faster but less accurate IFAST option uses the Arai–Agui–Nakajima (AAN) DCT [79, p. 52] algorithm with fixed-point arithmetic. The implementation processes columns first and rows second. Compared to the ISLOW implementation, fractional constants are converted to only 8-bit integers and intermediate results are descaled immediately.

The more accurate but slower IFLOAT DCT also uses the AAN DCT but with floats. We did not observe any notice-

<sup>3</sup>The scaling is explained in `jfdctint.c` in *libjpeg*.

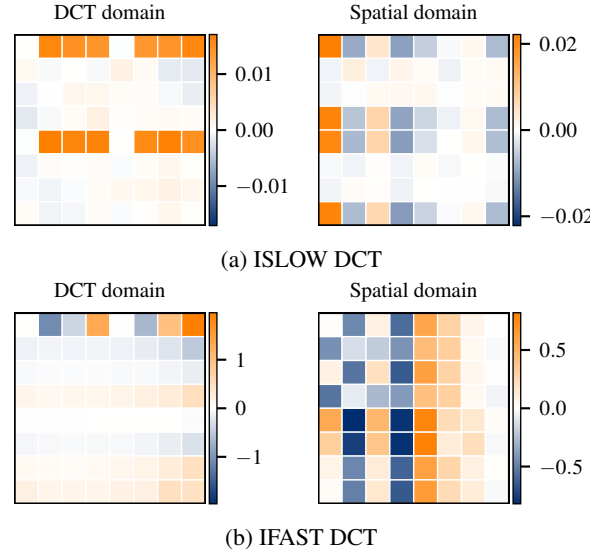


Figure 9: Practical DCT implementations in *libjpeg* leave directional rounding artifacts. The top row is the default setting.

able rounding errors. DCT rounding artifacts are a cause of directionality, which is strongest at quality factor 100 as quantization attenuates these weak traces.

**Validation** We draw 100 000 random symmetric  $8 \times 8$  blocks. Each block is DCT-transformed using the ISLOW and IFAST DCT methods. We measure the difference between the DCT block and a floating point version which does not have rounding artifacts. Figure 9 shows the average difference of the ISLOW DCT (top) and the IFAST DCT (bottom) in the DCT and the spatial domain. The DC difference is set to zero. Positive differences (orange) mean that the DCT implementation overestimates the corresponding coefficients. The ISLOW DCT overestimates rows 1 and 5 in the DCT domain. The differences are quite small. The IFAST DCT produces stronger artifacts, i.e., in spatial domain it underestimates columns 2 and 4 and overestimates column 5.

**Quantization tables.** Many quantization tables are asymmetric, including the ones *libjpeg* uses by default.

**Background** The idea of subband-specific quantization is to attenuate those spatial frequencies that are barely perceptible. The tables are stored in every JPEG file, therefore camera manufacturers and software developers can choose them.

In Annex K, the JPEG standard [53] provides example quantization tables (QTs) for luminance and chrominance components. They are the result of psychovisual thresholding experiments in which study participants were asked to indicate at what intensity the projection of each DCT basis function was discernible from a constant background [71]. Measurements from the original experiment are shown in



		Horizontal frequencies →							
Vertical frequencies ↓	16	11	10	16	24	40	51	61	
	12	12	14	19	26	58	60	55	
	14	13	16	24	40	57	69	56	
	14	17	22	29	51	87	80	62	
	18	22	37	56	68	109	103	77	
	24	35	55	64	81	104	113	92	
	49	64	78	87	103	121	120	101	
	72	92	95	98	112	100	103	99	

Figure 10: The luminance quantization table from Annex K of the JPEG standard [53] has several asymmetric features.

Table 2: Prevalence of standard quantization tables (QTs) and asymmetric luminance and chrominance QTs in native camera images from popular forensic datasets.

Dataset	Standard QTs	Asymmetric luma QT	Asymmetric chroma QT
Dresden	31 %	96 %	19 %
VISION	50 %	68 %	13 %
Forchheim	52 %	99 %	29 %

Appendix E. Although the JPEG standard emphasizes that these two tables are only examples, their inclusion as defaults in *libjpeg* has made them the de facto standard in many software implementations and in academic research. These QTs are still in use today, although the luminance table has been criticized since the 1990s for its asymmetry and because certain higher frequencies are less quantized than lower frequencies [62].

While the chrominance table is symmetric, the luminance table (shown in Fig. 10) has different values for horizontal and vertical frequencies. For example, at 2/2 cycles per block, the horizontal quantization factor is 10, whereas the vertical factor is 14 (orange circles). Conversely, at 5/2 cycles per block, the horizontal factor is 40, whereas the vertical one is 24 (blue). These seemingly minor differences affect JPEG steganography methods, such as J-UNIWARD [49] and UERD [43], which embed more into coefficients with lower quantization factors.

**Validation** We analyze the JPEG images of three forensic image databases<sup>4</sup> for the ratio of asymmetric QTs. Table 2 shows that across all databases, the majority of luminance tables are asymmetric. Conversely, the majority of chrominance tables are symmetric. Furthermore, we had access to 99 677 user-supplied JPEG images [78]. Our own reanalysis revealed 1 169 distinct luminance QTs including the standard tables

<sup>4</sup>We do not include the steganalysis databases because their original images are never compressed.

ranging from QF 1 to QF 100. Out of these 1 169 luminance QTs, 997 (85 %) are asymmetric.

*Libjpeg* calculates QTs by scaling the standard table [105]. The only QF leading to a symmetric QT is 100, where all entries are 1. *Libjpeg-turbo* uses the same luminance QTs. *MozJPEG* supports several QTs, including the standard tables. However, the default has changed to a symmetric table,<sup>5</sup> which refines a previous proposal [62]. On Ubuntu systems, the popular image libraries Pillow and OpenCV link against *libjpeg* version 6b and *libjpeg-turbo* version 8. Hence, software built with these packages introduces chroma wrinkles and uses asymmetric QTs.

**Lossless rotation** Most cameras produce JPEGs with a fixed orientation and store the sensor orientation with respect to the ground as EXIF metadata. This lets the viewer rotate the image after decompression [57]. JPEG images can also be rotated without decompression, e.g., using the command line tool `jpegtran`. This also rotates the directional traces.

## 4 Effect on steganalysis

Having discussed the causes of directionality, we move on to implications for security applications. We start with steganalysis, the task of detecting steganography [8]. We show for several steganography methods that state-of-the-art CNN-based detectors may fail if the stego images are rotated (Sec. 4.1). We present and evaluate mitigations (Sec. 4.2), and link the effect to specific causes of directionality (Sec. 4.3).

### 4.1 Sensitivity to rotated test images

The ALASKA2 database is split into three disjoint subsets: 80 % of the images are used for training, 10 % for validation, and 10 % for testing. All images have a resolution of 512<sup>2</sup> pixels and are compressed using *libjpeg-turbo* 2.1.0 with quality factor 75 and no chroma subsampling (4:4:4). The corresponding QT is roughly the standard QT shown in Fig. 10 divided by two. Steganography is simulated using the three state-of-the-art JPEG steganography methods nsF5 [36], UERD [43], and J-UNIWARD [49] with embedding rate 0.4 bits per non-zero AC coefficient (bpnzAC) into the luminance channel. This setup and rate corresponds to a commonly chosen evaluation setting in contemporary papers [17, 103, 104].

We use the AdamW optimizer to train an EfficientNet-B0 with ImageNet-pretrained weights [12], batch size 32, dropout rate 0.25, and learning rate 10<sup>-4</sup>. The training images in the *no-rot* setup were randomly flipped on the horizontal and vertical axis for data augmentation, but not rotated by 90 or 270 degrees in order to preserve the directionality.

<sup>5</sup><http://www.imagemagick.org/discourse-server/viewtopic.php?f=22&t=20333&p=98008#p98008>, accessed on 2 Feb, 2024

Table 3: Sensitivity of steganalysis to directionality (top section). The first two columns report the accuracy for quality factor 75, the last two columns for an artificially asymmetric QT blending quality factors 60 and 80.

Setup	Stego.	QF 75 (Section 4.1)		QF 80\60 (Section 4.2.2)	
		org.	rot.	org.	rot.
<i>no-rot</i>	nsF5	0.977	0.879	0.976	0.665
	UERD	0.922	0.893	0.927	0.557
	J-UNIWARD	0.884	0.801	0.893	0.492
<i>Effect of rotation augmentation</i> (Section 4.2.1)					
<i>aug-rot</i>	nsF5	0.978	0.976	0.977	0.975
	UERD	0.926	0.925	0.914	0.913
	J-UNIWARD	0.897	0.897	0.893	0.873
<i>base-rot</i>	nsF5	0.986	0.864	0.984	0.671
	UERD	0.927	0.876	0.938	0.543
	J-UNIWARD	0.911	0.780	0.908	0.505

The top section of Tab. 3 (*no-rot*, QF 75) reports the accuracy on the original test images (org.) and after rotating the test images by 90 degrees (rot.). Comparing the three steganography methods on the original orientation, nsF5 is least secure and J-UNIWARD is most difficult to detect. This confirms related studies [10, 12, 55]. Importantly, if the test images are rotated, the performance of all three detectors drops between 2.9 and 9.8 %-pts. This means that steganographers can gain an advantage by transmitting rotated images.

## 4.2 Mitigation with rotation augmentation

Data augmentation attempts to reduce overfitting by training on multiple variants of the existing samples. A popular augmentation practice is to rotate the decompressed training images by multiples of 90 degrees [10, 55, 101, 106]. Since image statistics are directional, rotating the test images can prevent a CNN from exploiting directional embedding traces. To validate this, we add two experimental setups.

### 4.2.1 Standard quantization table

The second setup, denoted *aug-rot*, is identical to the first, but the images are randomly rotated by multiples of 90 degrees in the spatial domain during training, as is typical for data augmentation. Hence, the detector must generalize to both the original and the transposed QT.

In the third setup, *base-rot*, we create three additional copies of the database, where the cover images are rotated by 90, 180, and 270 degrees prior to compression. The stego

images are created from the rotated covers. This dataset has four times the number of cover and stego images.

Table 3 reports the test accuracy for all setups and embedding methods. Both rotation setups slightly improve the detection accuracy over the *no-rot* setup, presumably because the CNNs are exposed to a greater variety of images. In the *aug-rot* setup, the detectors generalize to rotated images, because they have seen both orientations during training. This mitigates the weakness and removes the steganographer’s advantage from sending rotated images.

The highest accuracy on the original test images is achieved by training with the *base-rot* setup. The gain in accuracy compared to the *aug-rot* training ranges between 0.1 and 1.4 %-pts. Although our gains are small, they are within the range of improvements that have been reported in recent publications. For example, tweaking the EfficientNet-B0 architecture by removing pooling promises gains below 1 %-pt., but at the cost of tripling FLOPs and memory [103]. Unsurprisingly, the detectors in the *base-rot* setup experience the largest drop in accuracy when tested with rotated images. This is probably because they have learned directional traces in the training data that have been introduced by JPEG compression.

### 4.2.2 Artificial QT with amplified asymmetry

Although the standard QT contains many asymmetric features, the table as a whole does not exhibit a dominant direction. Stronger quantization in one horizontal coefficient might be compensated by another vertical coefficient. In order to link the drop in performance to directionality caused by JPEG compression, and to rule out alternative explanations, we repeat the experiments with a QT that has artificially strong directionality. Our artificial QT blends the QTs for QF 60 in the upper and QF 80 in the lower triangle, and their average on the diagonal. We chose this combination because it is close to QF 75. Note that the performance cannot be compared directly to the results for QF 75, because the difficulty of the steganalysis task significantly depends on the QT. Everything else is unchanged.

The two rightmost columns in Tab. 3 (QF 80\60) report the test accuracies for this experiment. As before, in the *no-rot* and *base-rot* setups, the CNNs do not generalize to rotated images. However, the performance drops are more pronounced. The accuracy on UERD and J-UNIWARD gets close to random guessing. We interpret this as evidence that the directionality picked up by the CNN is to a large extent introduced during JPEG compression, and specifically by the directionality in the QT. Again, the *base-rot* CNNs show improvements over the *aug-rot* CNNs between 0.7 and 2.4 %-pts.

## 4.3 Overfitting to dataset directionality

Popular image databases, including the steganalysis databases BOSSBase and ALASKA2, are dominated by images with

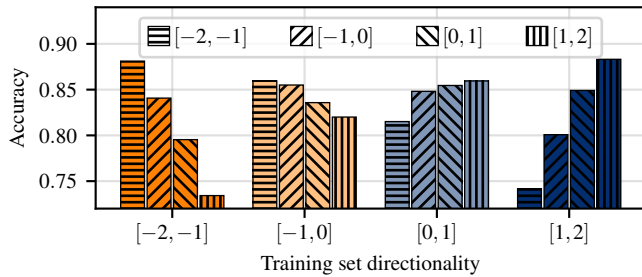


Figure 11: Effect of dataset directionality on steganalysis. Each set of bars refers to one detector trained on artificially biased data with directionality scores in the reported interval. Hatched bars report the resulting accuracy on test data with varying directionality. Observe the overfit on the outer bins.

more horizontal than vertical edges (cf. Fig. 5). When training on such biased databases, a learning-based steganalysis detector might overfit to the prevalent direction. We demonstrate this first on an artificial dataset and then on BOSSBase.

**Artificial dataset** For the first experiment, we chose the ALASKA2 database because it is larger than BOSSBase. The ALASKA2 database is split into 50 % for training and 50 % for testing. Both subsets are split by directionality into four bins with the directionality scores  $[-2, -1]$ ,  $[-1, 0]$ ,  $[0, 1]$ , and  $[1, 2]$ . From each bin, we randomly select the maximum number of images such that all bins have the same size (2484 training and 2592 test images). We then train a steganography detector with images from a single bin, and evaluate its test accuracy on all bins separately. To isolate the effect of the directionality of the scene content, we suppress the directional bias from JPEG compression by using a symmetrized variant of the QF-75 QT and the floating-point DCT method. Like before, we embed steganography into the luminance channel using J-UNIWARD with 0.4 bpnzAC. Since we have too few images for training a CNN-based detector, we use feature-based steganalysis. We extract the popular Gabor filter residual (GFR) [95] features from the luminance channel and train an ensemble classifier with  $\approx 100$  Fisher linear discriminants as base learners, as suggested in the literature [65].

Figure 11 shows the results. Each set of bars shows one ensemble classifier trained on one bin. The individual bars represent the accuracy on the individual test bins. When training with an extremely directional training set, i.e.,  $[-2, -1]$  and  $[1, 2]$ , the classifier achieves the highest accuracy on images with the same directionality. The accuracy drops when the test images contain a less pronounced or opposite directionality. When training with a moderately directional training set, i.e.,  $[-1, 0]$  and  $[0, 1]$ , the classifier shows higher accuracy for images of the same directionality than for the opposite direction. This experiment on an artificial subset of ALASKA2 demonstrates that even featured-based steganalysis is sensitive to a directionality bias in the training data.

Table 4: Effect of the directionality bias in the BOSSBase steganography benchmark dataset on detection accuracy.

Steganography method	Steganalysis features	Test all $d$	Test $d < 0$	Test $d > 0$
		# images	5000	3417
nsF5	GFR	0.993	0.993	0.991
	PHARM	0.987	0.989	0.980
UERD	GFR	0.894	0.900	0.879
	PHARM	0.860	0.868	0.841
J-UNIWARD	GFR	0.895	0.899	0.888
	PHARM	0.870	0.878	0.854

**BOSSBase** Typical datasets are more diverse in their directionality than the one in the previous experiment. Now we demonstrate that the directionality bias in BOSSBase may impair into a steganography detector’s generalizability. The BOSSBase images are split into 50 % for training and 50 % for testing. All images have a resolution of  $512^2$  pixels and are compressed using *libjpeg-turbo* 2.1.0 with the symmetric QF-75 QT and the floating-point DCT method. Steganography is simulated using nsF5 [36], UERD [43], and J-UNIWARD [49] with an embedding rate of 0.4 bpnzAC. In this experiment, we use two popular steganalysis feature descriptors: the phase-aware spatial rich models (PHARM) [48] and the Gabor filter residuals (GFR) [95]. Both feature sets do not merge horizontal and vertical orientations and can thus capture directional traces. We use the same ensemble classifier as before [65].

The left column in Tab. 4 shows the test accuracy over all test images. Additionally, we break down the test set by the directionality score  $d$ . The detector consistently achieves higher accuracy on test images with negative  $d$ . Although the difference is small, this result on a database that has served as a benchmark for many publications [10, 35, 43, 48, 49, 65, 95, 101] confirms that the detector works better on images that follow the database bias. We repeated this experiment on the original ALASKA2 database, but did not observe any significant differences in accuracy. This relates to the observation in Fig. 5, where we found that the directionality distribution in ALASKA2 is less skewed than in BOSSBase. ALASKA2 seems to be a better benchmark than BOSSBase in this regard.

## 5 Effect on forensic source identification

Linking a photo to a camera make or model is a common forensic task relevant to law enforcement [61]. This is possible because cameras differ in the technical implementation of the acquisition pipeline (cf. Fig. 4). While recent advances in camera model identification propose CNNs [9, 44, 86], feature-based classifiers [75, 76] are still used, presumably because

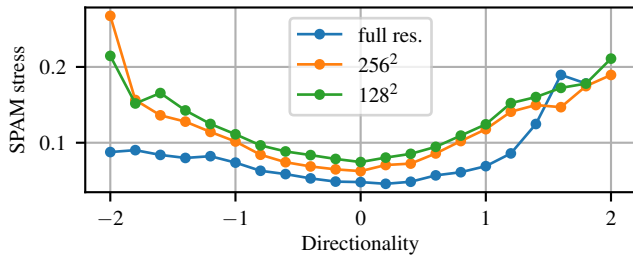


Figure 12: Relation between directionality and “stress” in the original SPAM feature space. Higher values indicate more information loss due to symmetrization. Our proposed variant removes this stress at the expense of greater dimensionality.

they are established, explainable (if necessary in court), and require fewer training images than CNNs. In this section, we show how considering directionality can improve the classification accuracy of feature-based methods. The results for the CNNs used in Fig. 1 are in Appendix C.

A popular feature descriptor for capturing camera model artifacts is the *subtractive pixel adjacency model (SPAM)* [75, 76]. SPAM features compute noise residuals and count co-occurrences between adjacent pixels. Both happens in horizontal and vertical direction. To decrease the feature dimensionality, certain horizontal and vertical co-occurrences are averaged [80]. This removes directional information. We show that feature-based camera model identification benefits from keeping directional image statistics separate.

Similar to previous work [75], we randomly select 10 camera models with at least two devices from the Dresden database [39]. To ensure that the classifier learns model-specific rather than device-specific traces, images from only one device are used for training. The other devices are used for testing. To properly account for potential lossless rotation (cf. Sec. 3.4), we verify that 17 (out of 18) camera models store the images with a canonical orientation. One camera model rotated the images prior to compression. We also use its images as stored, because this preserves the compression direction. We randomly select the same number of images from each camera model, such that training and test sets are class-balanced. To avoid leakage from the scene content, 80% of the scenes are exclusively used for training and the remaining 20% are used for testing.

We extract the SPAM features from each image. The original feature descriptor with directional averaging has 338 dimensions. Removing the directional averaging yields 676 dimensions. The feature extraction is repeated for the full-resolution images and for center crops with a size of  $256^2$  and  $128^2$ . Normalized features are used to train a linear support vector machine (SVM), as suggested in previous work [75]. The regularization hyper-parameter of the SVM is set via a grid search with five-fold cross validation.

Table 5 shows the classification accuracy for this 10-class task. The results are averaged over five training runs with ran-

Table 5: Effect of directionality on forensic camera model identification. The original SPAM features average horizontal and vertical information (symmetrization). Skipping this step improves the classification accuracy for small images.

Resolution	SPAM [80] (symmetrized)	Proposed variant (not symmetrized)
Full resolution	0.9972	0.9972
$256^2$	0.9722	0.9728
$128^2$	0.9041	0.9342

domly selected camera models. At the full image resolution, both feature descriptors attain the same near-perfect accuracy. For the more challenging task on smaller image patches, the classifier benefits from the directional information in our extended feature descriptor. At size  $128^2$ , our non-symmetrized variant outperforms the original by 3 %-pts. We observed similar results when using only the straight submodel or the diagonal submodel and with different classifiers.

An advantage of feature-based methods is that we can measure the information lost by symmetrization. Figure 12 plots the L2 distance between the submodels of different directionality that are averaged in the original SPAM features as a function of the directionality score. We call this distance “stress.” The two submodels are most similar for directionality scores close to zero. Stress increases with image directionality. Taken together, this provides the link between the directionality score, symmetrization in feature space, and accuracy.

## 6 Effect on synthetic image detection

GAN and diffusion-based synthetic image generators are known to introduce weak but detectable periodic artifacts [19]. They are explained by the repeated upsampling during image synthesis and therefore appear largely symmetrical [30]. Little attention has been paid to asymmetries in these artifacts. We show that synthetic images do contain directional traces, which CNN-based “fake-or-real” detectors tend to exploit.

We train CNN detectors for two synthetic image generators: DALL-E Mini [24] and Stable Diffusion XL (SDXL) [81]. For the real images, we use ALASKA2 and rotate each image by a random multiple of 90 degrees to remove directional bias. To create fake images, we pass the real images through the autoencoders of the two synthetic image generators. Compared to generating images from text prompts, this approach allows us to create real-fake pairs with matching scene content. This eliminates the possibility of spurious results due to shortcuts caused by a scene bias [38]. As detailed in Appendix D, this approach leaves the same synthesis artifacts as the typical text-to-image pipeline. 80% of the images are used for training, 10% for validation, and 10% for testing. We ensure that both real and fake images of the same scene are in the same

Table 6: CNN-based synthetic image detectors exploit directional artifacts introduced by the autoencoders of GANs.

Autoencoder	Orientation	Acc.	TNR	TPR
DALL-E Mini	original	1.000	1.000	0.999
	rotated	0.989	0.999	0.979
Stable Diffusion XL	original	1.000	1.000	1.000
	rotated	0.674	1.000	0.349

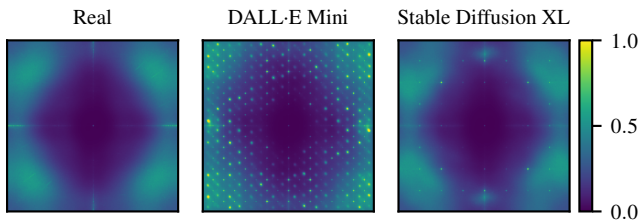


Figure 13: Power spectra of noise residuals from real images, the DALL-E Mini autoencoder, and the Stable Diffusion XL autoencoder. The last stages of generative models leave asymmetric artifacts that do not appear in real images.

split. We train an EfficientNet-B0 as binary classifier with cross-entropy loss, ImageNet pretraining, learning rate  $10^{-4}$ , batch size 32, and dropout rate 0.25.

Table 6 reports that both detectors achieve an accuracy of 1.0. The accuracy drops when the test images are rotated by 90 degrees. As can be seen from the true negative rate (TNR) and true positive rate (TPR), the performance loss is largely due to fake (positive) images being misclassified as real (negative). For the DALL-E Mini autoencoder, the TPR drops from 0.999 to 0.979, which means that 2.1 % of fake images are classified as real. For the SDXL autoencoder, the TPR even drops to 0.349, which means that almost two out of three fake images are classified as real. Rotated real images are correctly classified, i.e., the TNR remains near-perfect after rotation.

Directional artifacts can be observed in the power spectrum after averaging the noise residuals from 1 000 randomly selected images [19]. Figure 13 compares the randomly rotated ALASKA2 (left) to our DALL-E Mini (center) and SDXL (right) images. In ALASKA2, no directional pattern can be observed. Conversely, DALL-E Mini shows asymmetric dot patterns. SDXL introduces a high-frequency artifact on the vertical axis, which is not present on the horizontal axis.

Our experiments in this section demonstrate that synthetic images contain directional statistics. They are related to the autoencoder of synthetic image generators, but their root cause is yet unknown. As many recent detectors for synthetic images do not rotate their training images [21, 42, 90, 98, 99], these detectors may be less reliable on rotated test images. An attacker can generate an image of a rotated scene and manually rotate it back to reduce the chances of being caught.

## 7 Related work

Previous work in the security field has touched on the directionality of images, but the results are scattered. Westfeld [100] reports improved steganalysis by scanning images along a space-filling Hilbert curve. While it breaks up directionality, the method is justified by increased locality. Franz et al. [31–33] develop statistical models of flatbed scanner images. They observe directional differences between successive scans and attribute it to the variability of line sensor elements and the mechanical tolerances. None of this applies to the area sensors in today’s smartphones and digital cameras. A common forensic task on these sensors is to determine the configuration of the color filter array. The fact that this pattern depends on the camera orientation has been studied [60]. Recent work observed directional artifacts in DALL-E 2, but did not investigate further [4].

Popular feature descriptors in steganalysis [35, 63, 80] and image forensics [14] use directional high-pass filters to extract noise residuals or count co-occurrences in horizontal and vertical directions. They exploit sign and mirror symmetries to obtain a compact feature vector. Several early works made the simplifying assumption that image statistics are non-directional; examples include the subtractive pixel adjacency model (SPAM) (“the effect of portrait/landscape orientation is negligible” [80, p. 3]), the spatial rich model (“the statistics of natural images do not change after rotating the image by 90 degrees” [35, p. 5]), and the JPEG rich model (“assuming the statistics of natural images do not change after mirroring about the main diagonal” [63, p. 2]). Such assumptions can be traced back to early quantitative detectors of LSB replacement steganography (“What holds when scanning an image horizontally ought to hold, in general, when scanning vertically.” [59, p. 13]). While a lower feature dimensionality reduces the risk of overfitting and multicollinearity [102], it ignores the natural directional properties of images. Two more recent steganalysis feature descriptors, the phase-aware rich models (PHARM) [48] and the Gabor filter residuals (GFR) [95], do not average over orientations, but without giving a reason or supporting evidence. Some applications may ignore directionality when the orientation is known a priori or when the error rate is low enough to analyze the image in all four orientations, e.g., in PRNU matching.

The practices regarding rotation augmentation are inconsistent among subfields. In steganalysis, rotation augmentation is common [10, 55, 101]. In camera model identification and synthetic image detection, the majority of related work avoids rotation, with two exceptions [44, 74]. We did not find reflections on these choices, let alone comparisons. Rotation augmentation has been adopted from computer vision [67], where images are often downsampled and rotated by only a few degrees. However, these operations involve resampling which suppresses the signal of interest for security applications [55].

Overall, directionality of image statistics seems tangen-

tial to all of these works. While the security implications of certain image processing operations have received some attention recently (e.g., adversarial scaling [85]), a comprehensive treatment of directionality has been lacking.

## 8 Discussion

Directionality matters. This is the two-word summary of our multi-domain, multi-method study on the causes and effects of directionality in digital images in security applications.

This research has led to a number of new insights, primarily into the effects but also the causes of directionality. We are the first to report directional bias in popular benchmark image databases. We also point out that DCT implementations in popular JPEG libraries leave directional traces. For color images, directional chroma subsampling and asymmetric quantization tables add more directional traces. Across the three security applications studied, detectors trained on one image orientation do not naturally generalize to rotated images. In steganalysis, we found that the main cause for this is asymmetric quantization tables, followed by directional scene bias in the dataset. CNN-based fake-or-real detectors are susceptible to pick up directional traces left by the autoencoders in the last stage of generative models. Several feature descriptors used in steganalysis and image forensics average horizontal and vertical statistics. We found that a forensic camera model classifier benefits from keeping the directions separate, especially for small images.

Rotation augmentation requires more thought than is devoted in the extant literature. Our results lead to the following recommendations. If generalization to rotated images is desired, use rotation augmentation. To maximize performance on a directional dataset, avoid rotation augmentation. On small datasets, rotation augmentation can reduce overfitting but directionality-preserving transformations are preferable.

A qualitative observation is that many causes of directionality relate to legacy conventions and practices. While old standards “never die,” and are notoriously hard to replace, contemporary implementations of standards could depart from outdated defaults and instead use settings that do not leave new or amplify existing directional traces.

**Limitations** While we claim to have captured the dominant causes of directionality in raw-to-image conversion, our stock-taking is probably incomplete. A careful study of the literature reveals more idiosyncrasies of specific devices that may cause directionality. For example, the meanwhile historical Fujifilm J50 reportedly shifts the PRNU by seven pixels in horizontal direction depending on the exposure time [40].

Although the experimental setups in Sections 4, 5, and 6 closely replicate selected related work published in prominent venues, these scenarios only evaluate a single dataset, compression setting, and detector architecture. The effect of directionality may be larger or nonexistent in different

scenarios. Additionally, we trained and evaluated our CNNs with JPEG images that were *decompressed* with default settings, i.e., the ISLOW inverse DCT. Switching to the float DCT would slow down the training and the numerical differences are tiny. Real-world applications built with the popular TensorFlow framework might deviate much more since `tf.io.decode_jpeg` defaults to the IFAST DCT method.<sup>6</sup> The CNN detectors in Sec. 6 are tailored to *uncompressed* synthetic images. In practical applications, suspect images are often post-processed, scaled, and compressed. We intentionally chose uncompressed images to avoid potential inferences from directional effects of subsequent processing. Further experiments with more diverse setups are needed to quantify which factors interact with directionality.

**Possible extensions** Directionality relates to rotation by 90 degrees. Is there a similar paper to be written on horizontal and vertical flipping? Clearly, natural images may contain biases, e.g., a bright sky at the top and darker areas at the bottom. Some processes intentionally flip images, e.g., webcams make users feel like they are looking into a mirror. To show that directionality is still special, we retrain all CNNs from Fig. 1 without flipping augmentation, and evaluate their accuracy with flipped images. In all cases, the performance on flipped images matches the performance on the original images. This suggests that flipping matters less than rotation.

**Reproducibility** Our code for measuring directionality is available at <https://github.com/uibk-uncover/directionality>.

## 9 Concluding remark

We close with the conjecture that gravitation is the root cause of directionality in digital images. Human preference for landscape orientation has impacted scene composition as well as technology. This is likely due to the horizontal arrangement of mammalian eyes, which itself might have evolved to make our prehistoric ancestors more secure against predators that move on the surface [5].

## Acknowledgements

This work is funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 101021687 (UNCOVER). The numerical results were calculated on the University of Innsbruck’s LEO5 cluster and the Vienna Scientific Cluster (VSC). We thank our colleagues Nora Hofer, Martin Beneš, and Verena Lachner for helpful comments on the study, and Alain Léger, Eric Bourguignat, and Herbert Lohscheller for discussions on the JPEG history.

<sup>6</sup>See [https://github.com/tensorflow/tensorflow/blob/907415fea3e5e8ae2afee09c42f3cc1152fe63fd/tensorflow/core/kernels/image/decode\\_image\\_op.cc#L127](https://github.com/tensorflow/tensorflow/blob/907415fea3e5e8ae2afee09c42f3cc1152fe63fd/tensorflow/core/kernels/image/decode_image_op.cc#L127), accessed on 31 Jan, 2024

## References

- [1] S. Agarwal and H. Farid. Photo forensics from rounding artifacts. In *ACM Workshop Inf. Hiding MMSec.*, 2020.
- [2] C. Aguerrebere, J. Delon, Y. Gousseau, and P. Musé. Study of the digital camera acquisition process and statistical modeling of the sensor raw data. Technical report, 2012.
- [3] S. Appelle. Perception and discrimination as a function of stimulus orientation: The "oblique effect" in man and animals. *Psychol. Bull.*, 78(4):266–278, 1972.
- [4] Q. Bammey. Synthbuster: Towards detection of diffusion model generated images. *IEEE Open J. Signal Process.*, 5, 2024.
- [5] M. Banks, W. Sprague, J. Schmoll, J. Parnell, and G. Love. Why do animal eyes have pupils of different shapes? *Science Advances*, 1(7), 2015.
- [6] P. Bas, T. Filler, and T. Pevný. "Break our steganographic system": The ins and outs of organizing BOSS. In *Inf. Hiding*, volume 6958, pages 59–70. Springer, 2022.
- [7] M. Beneš, N. Hofer, and R. Böhme. Know your library: How the libjpeg version influences compression and decompression results. In *ACM Workshop Inf. Hiding MMSec.*, 2022.
- [8] R. Böhme. *Advanced Statistical Steganalysis*. Springer, 2010.
- [9] L. Bondi, D. Güera, L. Baroffio, P. Bestagini, E. Delp, and S. Tubaro. A preliminary study on convolutional neural networks for camera model identification. *Elect. Imaging*, 29(7):67–67, 2017.
- [10] M. Boroumand, M. Chen, and J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Trans. Inf. Forensics Sec.*, 14(5):1181–1193, 2019.
- [11] R. Burns. *Television: An International History of the Formative Years*. IET, 1998.
- [12] J. Butora, Y. Yousfi, and J. Fridrich. How to pretrain for steganalysis. In *ACM Workshop Inf. Hiding MMSec.*, 2021.
- [13] B. Chapman, M. Stryker, and T. Bonhoeffer. Development of orientation preference maps in ferret primary visual cortex. *J. Neuroscience*, 16(20):6443–6453, 1996.
- [14] C. Chen and M. Stamm. Camera model identification framework using an ensemble of demosaicing features. In *IEEE Int. Workshop Inf. Forensics Sec.*, 2015.
- [15] M. Chen, J. Fridrich, and M. Goljan. Digital imaging sensor identification (further study). In *Sec., stego., waterm. multimedia contents IX*, volume 6505, pages 258–270. SPIE, 2007.
- [16] R. Cogranne, Q. Giboulot, and P. Bas. ALASKA#2: Challenging academic research on steganalysis with realistic images. In *IEEE Int. Workshop Inf. Forensics Secur.*, 2020.
- [17] R. Cogranne, Q. Giboulot, and P. Bas. Efficient steganography in JPEG images by minimizing performance of optimal detector. *IEEE Trans. Inf. Forensics Sec.*, 17, 2022.
- [18] D. Coppola, H. Purves, A. McCoy, and D. Purves. The distribution of oriented contours in the real world. *Nat. Acad. Sciences*, 95(7):4002–4006, 1998.
- [19] R. Corvi, D. Cozzolino, G. Poggi, K. Nagano, and L. Verdoliva. Intriguing properties of synthetic images: From generative adversarial networks to diffusion models. In *IEEE/CVF Conf. Comp. Vis. Pattern Recognit. Workshops*, pages 973–982, 2023.
- [20] R. Corvi, D. Cozzolino, G. Zingarini, G. Poggi, K. Nagano, and L. Verdoliva. On the detection of synthetic images generated by diffusion models. In *IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023.
- [21] D. Cozzolino, D. Gragnaniello, G. Poggi, and L. Verdoliva. Towards universal GAN image detection. In *Int. Conf. Vis. Commun. Image Process.*, 2021.
- [22] D. Cozzolino, K. Nagano, L. Thomaz, A. Majumdar, and L. Verdoliva. Synthetic image detection: Highlights from the IEEE video and image processing cup 2022 student competition. *IEEE Signal Process. Mag.*, 40(7):94–100, 2023.
- [23] D. Cozzolino, G. Poggi, R. Corvi, M. Nießner, and L. Verdoliva. Raising the bar of AI-generated image detection with CLIP. In *CVPR Workshop Media Forensics*, 2024.
- [24] B. Dayma, S. Patil, P. Cuenca, K. Saifullah, T. Abraham, P. Lê Khac, L. Melas, and R. Ghosh. DALL-E mini. [github.com/borisdyma/dalle-mini](https://github.com/borisdyma/dalle-mini), 2021. accessed on 25 Jan, 2024.
- [25] J. Deng, W. Dong, R. Socher, L. Li, L. Kai, and F. Li. ImageNet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 248–255, 2009.
- [26] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. Canton Ferrer. The DeepFake Detection Challenge Dataset. *arXiv e-prints*, page arXiv:2006.07397, 2020.
- [27] B. Dornauer and M. Felderer. Web image formats: Assessment of their real-world-usage and performance across popular web browsers. In *Product-Focused Software Process Improvement*, pages 132–147, 2024.
- [28] H. Farid. *Photo Forensics*. MIT Press, 2016.
- [29] T. Filler, J. Fridrich, and M. Goljan. Using sensor pattern noise for camera model identification. In *IEEE Int. Conf. Image Process.*, pages 1296–1299, 2008.
- [30] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz. Leveraging frequency analysis for deep fake image recognition. In *Int. Conf. Mach. Learn.*, volume 119, pages 3247–3258, 2020.
- [31] E. Franz. *Steganographie durch Nachbildung plausibler Änderungen*. Dissertation, TU Dresden, 2002.
- [32] E. Franz. Embedding considering dependencies between pixels. In *Sec., Forensics, Stego., Waterm. Multimedia Contents X*, volume 6819, 2008.
- [33] E. Franz and A. Pfitzmann. Steganography secure against cover-stego-attacks. In *Inf. Hiding*, pages 29–46, 2000.
- [34] J. Fridrich. *Sensor Defects in Digital Image Forensic*, pages 179–218. 2013.
- [35] J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. *IEEE Trans. Inf. Forensics Sec.*, 7(3):868–882, 2012.

- [36] J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography: dead ends challenges, and opportunities. In *Workshop MMSec.*, pages 3–14, 2007.
- [37] A. El Gamal, B. Fowler, H. Min, and X. Liu. Modeling and estimation of FPN components in CMOS image sensors. In *Solid State Sensor Arrays: Develop. Appl.* 2, volume 3301, pages 168–177, 1998.
- [38] R. Geirhos, J. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. Wichmann. Shortcut learning in deep neural networks. *Nature Mach. Intelligence*, 2(11), 2020.
- [39] T. Gloe and R. Böhme. The ‘Dresden image database’ for benchmarking digital image forensics. In *ACM Symp. Applied Comp.*, pages 1584–1590, 2010.
- [40] T. Gloe, S. Pfennig, and M. Kirchner. Unexpected artefacts in PRNU-based camera identification: A ‘Dresden image database’ case-study. In *MMSec. Workshop*, 2012.
- [41] V. Goffaux, F. Duecker, L. Hausfeld, C. Schiltz, and R. Goebel. Horizontal tuning for faces originates in high-level fusiform face area. *Neuropsychologia*, 81:1–11, 2016.
- [42] D. Gragnaniello, D. Cozzolino, F. Marra, G. Poggi, and L. Verdoliva. Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. In *IEEE Int. Conf. Multimedia Expo*, 2021.
- [43] L. Guo, J. Ni, W. Su, C. Tang, and Y. Shi. Using statistical image model for JPEG steganography: Uniform embedding revisited. *IEEE Trans. Inf. Forensics Sec.*, 10(12), 2015.
- [44] B. Hadwiger and C. Riess. The Forchheim image database for camera identification in the wild. In *Int. Conf. Pattern Recognit. Workshops*, volume 12666, pages 500–515, 2020.
- [45] B. Hansen and E. Essock. A horizontal bias in human visual processing of orientation and its correspondence to the structural components of natural scenes. *J. Vis.*, 4(12):5, 2004.
- [46] E. Hecht. *Optics*. Pearson Education, 5 edition, 2016.
- [47] N. Hofer and R. Böhme. Progressive JPEGs in the wild: Implications for information hiding and forensics. In *ACM Workshop Inf. Hiding MMSec.*, pages 47–58, 2023.
- [48] V. Holub and J. Fridrich. Phase-aware projection model for steganalysis of JPEG images. In *Media Waterm., Sec., Forensics*, volume 9409, 2015.
- [49] V. Holub, J. Fridrich, and T. Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP J. Inf. Sec.*, 2014:1, 2014.
- [50] G. Hudson, A. Léger, B. Niss, I. Sebestyén, and J. Vaaben. JPEG-1 standard 25 years: past, present, and future reasons for a success. *J. Elect. Imag.*, 27(4):040901, 2018.
- [51] G. Hudson, A. Léger, B. Niss, and I. Sebestyén. JPEG at 25: Still going strong. *IEEE Multimedia*, 24(2):96–103, 2017.
- [52] Independent JPEG Group. libjpeg. [ijg.org](http://ijg.org). Accessed on 25 Jan, 2024.
- [53] International Telecommunication Union. ISO/IEC 10918-1: 1993(E) CCIT Recommendation T.81. [w3.org/Graphics/JPEG/itu-t81.pdf](http://w3.org/Graphics/JPEG/itu-t81.pdf), 1993.
- [54] International Telecommunication Union. ITU-R BT.601. [itu.int/rec/R-REC-BT.601](http://itu.int/rec/R-REC-BT.601), 2011.
- [55] T. Itzhaki, Y. Yousfi, and J. Fridrich. Data augmentation for JPEG steganalysis. In *IEEE Int. Workshop Inf. Forensics Sec.*, 2021.
- [56] M. Jancovic. *Interlacing: The First Video Compression Method*, pages 77–119. 2023.
- [57] JPEGClub. Exif orientation tag. [jpegclub.org/exif\\_orientation.html](http://jpegclub.org/exif_orientation.html), 2002. Accessed on 26 Jan, 2024.
- [58] M. Keil and G. Cristóbal. Separating the chaff from the wheat: Possible origins of the oblique effect. *J. Optical Society America A*, 17(4):697–710, 2000.
- [59] A. Ker. Quantitative evaluation of pairs and RS steganalysis. In *Sec., Stego., Waterm. Multimedia Contents VI*, volume 5306, pages 83–97, 2004.
- [60] M. Kirchner. Efficient estimation of CFA pattern configuration in digital camera images. In *Media Forensics and Security II*, volume 7541, 2010.
- [61] M. Kirchner and T. Gloe. *Forensic Camera Model Identification*, pages 329–374. Wiley, 2015.
- [62] S. Klein, D. Silverstein, and T. Carney. Relevance of human vision to JPEG-DCT compression. In *Human Vision, Visual Processing, and Digital Display III*, volume 1666, 1992.
- [63] J. Kodovský and J. Fridrich. Steganalysis of JPEG images using rich models. In *Media Waterm., Sec., Forensics*, volume 8303, 2012.
- [64] J. Kodovský and J. Fridrich. Steganalysis in resized images. In *IEEE Int. Conf. Acoust., Speech, Signal Process.*, pages 2857–2861, 2013.
- [65] J. Kodovský, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Trans. Inf. Forensics Sec.*, 7(2):432–444, 2012.
- [66] R. Latto and K. Russell-Duff. An oblique effect in the selection of line orientation by twentieth century painters. *Empirical studies of the arts*, 20(1):49–60, 2002.
- [67] Y. LeCun, L. Jackel, L. Bottou, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261(276):2, 1995.
- [68] libjpeg-turbo. Main libjpeg-turbo repository. [github.com/libjpeg-turbo/libjpeg-turbo](https://github.com/libjpeg-turbo/libjpeg-turbo). Accessed on 25 Jan, 2024.
- [69] C. Loeffler, A. Ligtenberg, and G. Moschytz. Practical fast 1-D DCT algorithms with 11 multiplications. In *IEEE Int. Conf. Acoust., Speech, Signal Process.*, pages 988–991, 1989.
- [70] H. Lohscheller. *Einzelbildübertragung mit wachsender Auflösung*. Dissertation, Technische Hochschule Aachen, 1982.
- [71] H. Lohscheller. A subjectively adapted image communication system. *IEEE Trans. Commun.*, 32(12):1316–1322, 1984.
- [72] B. Lorch and C. Riess. Image forensics from chroma subsampling of high-quality JPEG images. In *ACM Workshop Inf. Hiding MMSec.*, pages 101–106, 2019.



- [73] S. Mandelli, N. Bonettini, and P. Bestagini. *Source Camera Model Identification*, pages 133–173. 2022.
- [74] S. Mandelli, N. Bonettini, P. Bestagini, and S. Tubaro. Detecting GAN-generated images by orthogonal training of multiple CNNs. In *IEEE Int. Conf. Image Process.*, 2022.
- [75] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva. Evaluation of residual-based local features for camera model identification. In *Int. Conf. Image Anal. Process. Workshops*, 2015.
- [76] P. Mendes Júnior, L. Bondi, P. Bestagini, S. Tubaro, and A. Rocha. An in-depth study on open-set camera model identification. *IEEE Access*, 7:180713–180726, 2019.
- [77] Mozilla Foundation. MozJPEG: Improved JPEG encoder. [github.com/mozilla/mozjpeg](https://github.com/mozilla/mozjpeg), 2014. Accessed on 25 Jan, 2024.
- [78] J. Park, D. Cho, W. Ahn, and H. Lee. Double JPEG detection in mixed JPEG quality factors using deep convolutional neural network. In *Eur. Conf. Comp. Vis.*, 2018.
- [79] W. Pennebaker and J. Mitchell. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, 1992.
- [80] T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans. Inf. Forensics Sec.*, 5(2):215–224, 2010.
- [81] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. *arXiv e-prints*, page arXiv:2307.01952, 2023.
- [82] A. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Trans. Signal Process.*, 53(2):758–767, 2005.
- [83] C. Poynton. *Digital Video and HDTV: Algorithms and Interfaces*. Morgan Kaufmann Publishers, 2003.
- [84] N. Provos. Defending against statistical steganalysis. In *USENIX Sec. Symp.*, 2001.
- [85] E. Quiring, D. Klein, D. Arp, M. Johns, and K. Rieck. Adversarial preprocessing: Understanding and preventing image-scaling attacks in machine learning. In *USENIX Sec. Symp.*, pages 1363–1380, 2020.
- [86] A. Rafi, T. Tonmoy, U. Kamal, Q. Wu, and M. Hasan. RemNet: remnant convolutional neural network for camera model identification. *Neural Comput. Appl.*, 33(8):3655–3670, 2020.
- [87] E. Reinhard, E. Khan, A. Akyz, and G. Johnson. *Color Imaging: Fundamentals and Applications*. A. K. Peters, 2008.
- [88] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conf. Comp. Vis. Pattern Recognit.*, pages 10674–10685, 2022.
- [89] M. Segall, D. Campbell, and M. Herskovits. *The influence of culture on visual perception*, volume 310. Bobbs-Merrill, 1966.
- [90] K. Shiohara and T. Yamasaki. Detecting deepfakes with self-blended images. In *IEEE/CVF Conf. Comp. Vis. Pattern Recognit.*, pages 18699–18708, 2022.
- [91] D. Shullani, M. Fontani, M. Iuliani, O. Al Shaya, and A. Piva. VISION: A video and image dataset for source identification. *EURASIP J. Inf. Sec.*, 2017:15, 2017.
- [92] G. Simmons. *The Prisoners’ Problem and the Subliminal Channel*, pages 51–67. 1984.
- [93] E. Simoncelli and W. Freeman. The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *IEEE Int. Conf. Image Process.*, volume 3, 1995.
- [94] I. Sobel and G. Feldman. An isotropic 3x3 image gradient operator. Technical report, 2014.
- [95] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang. Steganalysis of adaptive JPEG steganography using 2D Gabor filters. In *ACM Workshop Inf. Hiding MMSec.*, pages 15–23, 2015.
- [96] E. Switkes, M. Mayer, and J. Sloan. Spatial frequency analysis of the visual environment: Anisotropy and the carpentered environment hypothesis. *Vis. Res.*, 18(10):1393–1399, 1978.
- [97] M. Tan and Q. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Int. Conf. Mach. Learn.*, volume 97, pages 6105–6114, 2019.
- [98] S. Wang, O. Wang, R. Zhang, A. Owens, and A. Efros. CNN-generated images are surprisingly easy to spot... for now. In *IEEE/CVF Conf. Comp. Vis. Pattern Recognit.*, pages 8692–8701, 2020.
- [99] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen, and H. Li. DIRE for diffusion-generated image detection. In *IEEE/CVF Int. Conf. Comp. Vis.*, pages 22388–22398, 2023.
- [100] A. Westfeld. Space filling curves in steganalysis. In *Sec., Stego., Waterm. Multimedia Contents*, volume 5681, 2005.
- [101] J. Ye, J. Ni, and Y. Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Trans. Inf. Forensics Sec.*, 12(11):2545–2557, 2017.
- [102] M. Yedroudj, M. Chaumont, and F. Comby. How to augment a small learning set for improving the performances of a CNN-based steganalyzer? In *Media Waterm., Sec., Forensics*, 2018.
- [103] Y. Yousfi, J. Butora, J. Fridrich, and C. Fuji Tsang. Improving EfficientNet for JPEG steganalysis. In *ACM Workshop Inf. Hiding MMSec.*, pages 149–157, 2021.
- [104] Y. Yousfi, J. Butora, E. Khvedchenya, and J. Fridrich. Imagenet pre-trained CNNs for JPEG steganalysis. In *IEEE Int. Workshop Inf. Forensics Sec.*, 2020.
- [105] Y. Yousfi and J. Fridrich. JPEG steganalysis detectors scalable with respect to compression quality. In *Media Waterm., Sec., Forensics*, 2020.
- [106] J. Zhang, K. Chen, C. Qin, W. Zhang, and N. Yu. Distribution-preserving-based automatic data augmentation for deep image steganalysis. *IEEE Trans. Multimedia*, 24:4538–4550, 2022.
- [107] K. Zhang, W. Zuo, Y. Chen, Y. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.*, 26(7), 2017.

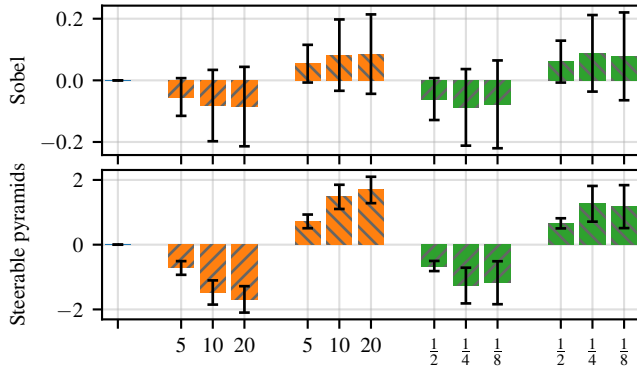


Figure 14: Comparison of Sobel filters (top) and steerable pyramids (bottom) for five types of test data. Both detectors indicate the dominant image direction, but the steerable pyramids are more precise, as indicated by the tighter error bars.

## A Data

The experiments in this paper draw on multiple databases, summarized in Tab. 7. BOSSBase (grayscale) and ALASKA2 (color) are popular in the steganography and steganalysis literature. From ImageNet, we use training set. The Dresden, VISION, and Forchheim databases are popular in the forensics community for benchmarking camera device or model identification. Camera annotations make them ideal for camera-specific statistics and the study of prevalent compression patterns. From the Dresden database, we use the natural scene subset. From the Forchheim database, we use the camera-original subset. Additionally, we acquired 201 flat-field frames with a Sony Alpha 6600 and a Nikon D850.

Table 7: Databases used for illustrating the causes and effects of directionality

Name	Purpose	Format	# images
BOSSBase [6]	Steganalysis	NC	10 000
ALASKA2 [16]	Steganalysis	NC	80 005
ImageNet [25]	Image recog.	JPEG	1 281 141
Dresden [39]	Forensics	JPEG	16 961
VISION (flat) [91]	Forensics	JPEG	4 167
VISION (nat) [91]	Forensics	JPEG	7 565
Forchheim [44]	Forensics	JPEG	3 851
Own (flat)	-	NC	402

NC = never compressed

## B Comparison of directionality scores<sup>7</sup>

This section compares two directionality detectors based on the Sobel filter and steerable pyramids. We generate 1 000

<sup>7</sup>A longer description is available in an arXiv version of this paper.

Table 8: A CNN multi-class classifier for forensic camera model identification cannot maintain its accuracy on rotated test images. Rotation augmentation overcomes this limitation.

Training setup	Accuracy original	Accuracy rotated
<i>no-rot</i>	0.941	0.845
<i>aug-rot</i>	0.932	0.929

*symmetric* synthetic images of size  $512^2$ , consisting of cosine patterns with randomized amplitude and additive noise. We produce four directional variants using horizontal/vertical smoothing and horizontal/vertical downscaling.

Figure 14 shows the average directionality scores of the Sobel (top) and the steerable pyramids directionality detectors (bottom). As expected, horizontal/vertical smoothing (orange) or downscaling (green) leads to negative/positive directionality scores. The errors bars show that steerable pyramids are more precise than the Sobel-based detector.

## C CNN-based camera model identification

This part has been moved to the appendix to make space for the SPAM feature descriptor, which allows us to directly measure the “stress” caused by explicit symmetrization. The headline result in Fig. 1 use a CNN-based method for better comparability across applications. Here we describe the experiment and report on the effect of rotation augmentation.

We replicate the setup of [44], who train an EfficientNet-B5 as 27-class classifier on the Forchheim dataset, and pay particular attention to factors that affect directionality. We verified that 22 cameras stored their images in canonical orientation and set the EXIF orientation flag (cf. Sec. 3.4). The other cameras stored rotated images without orientation metadata. We use all images as stored to preserve the compression direction.

Each image was subdivided into non-overlapping patches of size  $256^2$ . From each image, we select the 100 patches with the highest quality according to a heuristic quality metric [9]. This approach discards non-informative regions, e.g., consisting of homogeneous pixels. We trained on randomly cropped sub-patches of size  $64^2$ , using an initial learning rate of  $10^{-5}$ , and a dropout rate of 0.25. The learning rate halves whenever the validation loss stagnates for 500 epochs. Training ends when the validation loss does not improve for 1000 epochs.

We evaluate two setups. In the *no-rot* setup, the training patches are randomly flipped along the vertical and horizontal axis. In the *aug-rot* setup, the training patches are additionally rotated by a random multiple of 90 degrees, as in [44].

Table 8 reports the results. In the *no-rot* setup, the CNN achieves an accuracy of 0.941. The rightmost column reports the accuracy after rotating the test images by 90 degrees in counter-clockwise direction. The drop in accuracy by 9.6%pts. shows that the images contain directional information,

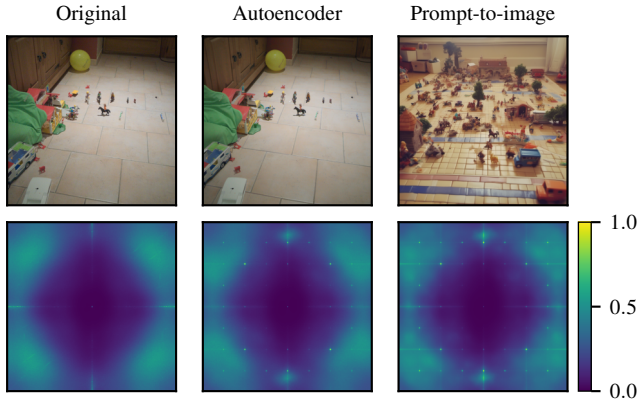


Figure 15: Passing a real image through SDXL’s autoencoder inserts the same frequency-domain artifacts as in images generated from text prompts while preserving the scene content.

which the CNN has learned to exploit. In the *aug-rot* setup, the CNN achieves a slightly lower accuracy of 0.932 but generalizes to rotated test images.

## D SDXL autoencoder vs. text-to-image

Today’s synthetic image generators split the image generation into a semantic and a perceptual part [88]. The perceptual part consists of an autoencoder that provides a compact image representation and can reconstruct images from this representation. The semantic part consists of a generative model that synthesizes the semantic content of the image in the autoencoder’s compact representation. The autoencoder’s decoder fills in the perceptual details while decompressing the generated representation to an image.

We considered compressing the ALASKA2 dataset to text prompts with the CLIP interrogator<sup>8</sup> and reconstructing a synthetic ALASKA2 dataset from the text prompts. Working with a small subset, we found that some of the generated images were surprisingly close to the original images. In many cases, however, the text prompts encouraged an artistic style or hallucinated animals. We were concerned that this might introduce scene bias and discarded this approach.

Our fake images in Sec. 6 were created by passing real images through the generators’ autoencoder. Figure 15 demonstrates that this approach leaves the same artifacts as synthesizing images from text prompts. The top row shows a real image from ALASKA2 (left), the image after compression and decompression via the SDXL autoencoder (center), and an image synthesized from a text prompt (right). The bottom row shows the power spectrum after averaging 1 000 noise residuals. The autoencoder image has the same scene content as the original image, but shows the same artifacts as images

<sup>8</sup><https://github.com/pharmapsychotic/clip-interrogator/> with the CLIP model *ViT-g-14/laion2B-s34B-b88K*.

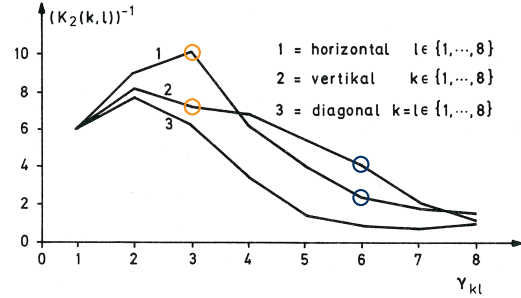


Figure 16: Psychovisual thresholding experiments revealed different contrast sensitivity profiles for horizontal, vertical, and diagonal DCT basis functions. Figure reproduced from [70, p. 138]. The circles are our own annotations, relating to the quantization factors highlighted in Fig. 10.

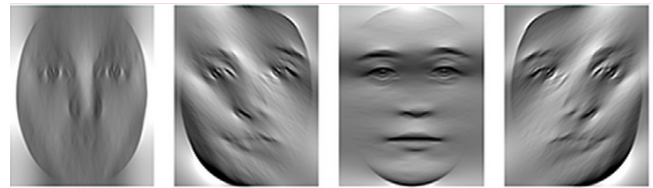


Figure 17: Orientation-filtered stimuli used in neuroscience. Images reproduced with kind permission of the authors [41].

synthesized from text prompts. The artifacts in the right column appear slightly sharper because SDXL produces 1024<sup>2</sup> images while the autoencoder maintains the 512<sup>2</sup> image resolution. This shows that synthesis artifacts are introduced by the autoencoder, as also observed in related work [23].

## E Measurement of human contrast sensitivity

Human contrast sensitivity to the 64 DCT basis functions was measured by Lohscheller in the early 1980s [70, pp. 122–142]. Figure 16 shows the measured contrast sensitivity for the horizontal, vertical, and diagonal DCT basis functions. The values are averages of 18 human subjects. Note that horizontal and vertical sensitivity are different and the order changes between low and high frequencies. These measurements are reflected in the standard JPEG quantization table (Fig. 10), which is still widely used today.

## F Horizontal edges in human face perception

Goffaux et al. confirmed that human brains prefer horizontal contours when recognizing human faces [41]. They presented orientation-filtered stimuli images to study participants in fMRI scanners. These images are reproduced in Fig. 17. The strongest activity in high-level face-specialized ventral regions of the human brain was observed in the condition where horizontal edges were preserved, as in the third image.