Detecting CNN-Generated Facial Images in Real-World Scenarios

Nils Hulzebosch[1,2] Sarah Ibrahimi[1,2] Marcel Worring[1]
[1]University of Amsterdam [2]Dutch National Police

Abstract

Artificial, CNN-generated images are now of such high quality that humans have trouble distinguishing them from real images. Several algorithmic detection methods have been proposed, but these appear to generalize poorly to data from unknown sources, making them infeasible for real-world scenarios. In this work, we present a framework for evaluating detection methods under real-world conditions, consisting of cross-model, cross-data, and post-processing evaluation, and we evaluate state-of-the-art detection methods using the proposed framework. Furthermore, we examine the usefulness of commonly used image pre-processing methods. Lastly, we evaluate human performance on detecting CNN-generated images, along with factors that influence this performance, by conducting an online survey. Our results suggest that CNN-based detection methods are not yet robust enough to be used in real-world scenarios.

1. Introduction

Recently, state-of-the-art CNN-based generative models have radically improved the visual quality of generated images [21, 22]. Combined with an increasing ease of using such models by non-experts through user friendly applications (e.g. [8, 10, 39]), there is sufficient reason to be cautious about its use by people with harmful intents. The malicious use of technologies employing generative models has been demonstrated with DeepFakes in the form of (revenge) pornography, where faces of women are mapped to pornographic videos [8], and with DeepNude by undressing women [10]. The potential of DeepFakes for political purposes has also been demonstrated in [9, 15, 36], and has the capability to become a significant problem in terms of fake news and propaganda. Current state-of-the-art generative models [21, 22] go one step further and are capable of creating fully-generated realistic images of human faces. The development of image generation techniques will likely have ethical, moral, and legal consequences.

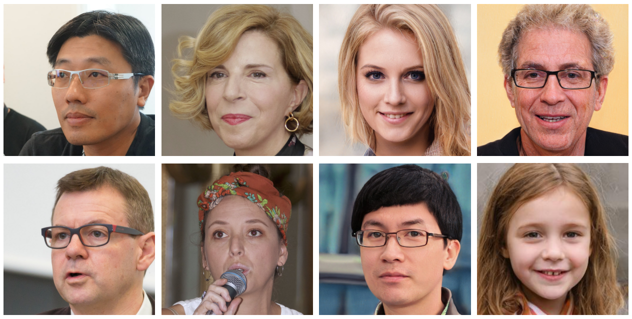

Figure 1: Can you distinguish fake from real images? The answers are shown below.[1]

Generative Adversarial Networks (GANs) [16] could be regarded as the most promising and widely used type of generative models for image creation and manipulation. In only a few years of existence, many features such as: visual image quality; image resolution; range of control over the output; and ease of training these models have been improved. Recently, [21] proposed StyleGAN, which is able to generate nearly photo-realistic facial images of 1024x1024 resolution, along with some stylistic control over the output, as presented in Figure 1. [22] has proposed an improved version with reduced visual artefacts. To counteract the development of generative models, automatic fake imagery detection methods have gained increasing interest. Many works focus on learning-based detection, using Convolutional Neural Networks (CNNs). They work well on data similar to that seen during training, but often fail when images are generated by other GANs [13] or when images are post-processed [28].

Deviations in data sources and post-processing techniques are inconvenient in real-world scenarios. In this work, we refer to real-world scenarios as scenarios where an image encountered has an unknown source and possibly underwent unknown forms of post-processing after its creation. Furthermore, an image should be of reasonable size and should have no clearly visible alterations that lowers its credibility of being authentic. An example of such a real-world scenario is a forensic setting where the authenticity of an image must be determined. It is desirable that a

[1] Left four are *real* (from the FFHQ dataset), and right four are *generated* by StyleGAN (trained on the FFHQ dataset).

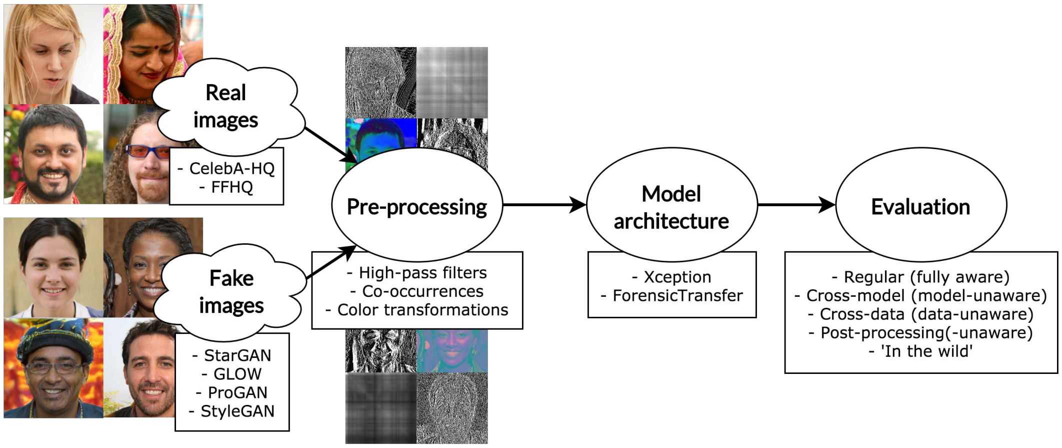

Figure 2: Overview of our experimental pipeline. Two state-of-the-art detection models are evaluated under real-world scenarios with a focus on cross-model, cross-data and post-processing scenarios. Pre-processing techniques are examined for generalizability.

detection method works well, independently of the type of model that generated or manipulated the encountered image. Another example is images encountered on social media pages, which may be unintentionally or deliberately altered. Examples of unintentional alterations are compression and resampling (or resizing), which often happen when uploading images onto social media or viewing images in a web browser. Blurring, adding noise, and adjusting colors are examples of deliberate alterations. We also assume that in real-world scenarios, the majority of users that encounter these images are not trained to detect fake images. Based on the trend of applications using DeepFakes and the advanced techniques to create realistic fully-generated images, we expect that the use of fully-generated images will be the next trend for new applications. For this reason, it is important to take both real-world conditions and fully-generated images into consideration when evaluating detection methods.

In this work, we aim to evaluate state-of-the-art image generation models under an approximation of real-world conditions, using the following three categories: 1) a *cross-model scenario*, where the type of model used to generate an image is unknown, 2) a *cross-data scenario*, where the data used to train a generative model is unknown, and 3) a *post-processing scenario*, where an image is modified with an unknown type of post-processing. For each category we examine whether the generalizability of learning-based methods could be improved using commonly used pre-processing methods. Our work focuses on facial images, since most applications are targeted on facial generation or manipulation.

Our main contributions are the following:
1) We propose a framework, presented in Figure 2, consisting of three types of evaluation required for robust evaluation under real-world conditions: cross-model, cross-data, and post-processing evaluation;
2) We evaluate the most promising state-of-the-art model architectures and pre-processing methods;
3) We perform a user study with 496 participants and measure human performance of detecting state-of-the-art generated images and factors that influence this performance.

2. Related work

In this section, we review methods for CNN-generated image detection, image pre-processing and human detection of image forgery. GANs [16] have recently emerged as the state-of-the-art in generating realistic imagery, in terms of image resolution and visual quality. Recent works have been able to generate nearly photo-realistic facial images [21, 22]. Other works focus on more control over the output images, mainly in the fields of stylistic manipulation [6, 18, 55] or semantic manipulation [17, 35, 37, 50]. Our work includes models capable of unconditional generation [20, 21] and conditional stylistic manipulation [6, 21, 23] of human faces.

CNN-generated image detection Early work of CNN-generated image detection uses handcrafted features based on domain knowledge. Two examples of domain knowledge that could be exploited are image color information and human facial appearances [24, 30, 31, 49]. While these methods have reasonable performance, such handcrafted features are less applicable in real-world scenarios, where images often do not adhere to some of the assumptions made for these methods *e.g.* when faces are partially covered. In this case, the methods of [30, 49] might not work.

Following these works, learning-based methods have been proposed, using CNNs to automatically learn features of real and generated images [1, 3, 7, 12, 13, 40]. [13] presents ForensicTransfer and achieves state-of-the-art results on detecting CNN-inpainted images [17, 51] and fully CNN-generated images [6, 19, 20, 23]. Another commonly used architecture for CNN-generated image detection is the Xception model [7], originally proposed as image classification model trained on ImageNet [43]. [28, 42] both evaluate several models and show that Xception yields the best overall performance across regular and compressed images in detecting fully-generated images [28] and CNN-manipulated images [42]. Futhermore, the evaluation of [13] shows good performance of Xception, in some evaluation setups outperforming the ForensicTransfer model. [48] proposes a model, along with several data augmentation procedures, to detect fully-generated images of unknown sources. The results suggest that increasing the number of image classes, as well as randomly blurring and compressing images during training, increases the robustness of CNN-based detectors, yielding good results in cross-model and post-processing scenarios. [27] finds that real and fake images have textural differences and exploit this by proposing a *Gram-Net* model architecture to focus on global image textures, yielding good results in cross-model, cross-data, and post-processing scenarios.

We select ForensicTransfer [13] and Xception [7] for our evaluation. We did not take the architectures of [27] and [48] into account, since they were not published yet at the time this research was conducted. Given that these works are extensions of the state-of-the-art models we expect our results are still valid since our work focuses on different types of pre-processing techniques and datasets. We also evaluate an *in the wild* scenario exclusively for facial images. Additionally, we are the first to perform a large scale user study that compare human performance under realistic conditions to model performance.

Image pre-processing Pre-processing an image before passing it to a CNN-based model is not uncommon in the field of image forgery detection and has been studied by several works [5, 11, 12, 13, 24, 25, 33, 41]. The motivation is to enrich or focus on specific information in the image, such that learning the difference between real and generated (or manipulated) might be more fruitful. As shown by [2, 29, 52], CNN-generated images have pixel patterns dissimilar to real images, which might become more distinctive by learning more intrinsic (pixel-level) image features, such that detection models might generalize better to unseen (*e.g.* model-unaware) fake images.

Several works on image forgery detection [12, 13, 25, 32, 41] include high-pass filters as a way to accentuate the high-frequency structure of an image. Another type of pre-processing is color transformation, where non-RGB color information is used to detect forgeries. [24] has shown the effectiveness of detecting generated images, using HSV (hue, saturation, value) and YCbCr (luma, red-chroma difference, and blue-chroma difference) color information along with a feature-based approach. Lastly, several works use co-occurrence matrices to focus on irregularities in pixel-patterns, for example in steganalysis [12, 14, 38, 45, 46] and detection of forged images [11, 12]. Recently, [33] has used this approach for detecting CNN-generated images, suggesting good performance in several evaluation scenarios. Most works seem to evaluate one type or class of pre-processing method(s) with one model architecture [13, 32, 33]. The interaction between pre-processing methods and model architectures remains unclear as well as the benefits of pre-processing methods. In our work, we focus on these interactions by examining three common types of pre-processing: 1) high-pass filters, 2) co-occurrence matrices, and 3) color transformations.

Human detection of image forgery Humans have trouble distinguishing forged images from authentic images, especially when no comparison material is provided to them [34, 44, 53]. Examples include detection of erase-fill, copy-move, cut-paste, and changes in reflections. [42] shows that humans have trouble detecting CNN-modified images. Recent work by [54] addresses human performance on fully-generated GAN-images specifically. However, their aim is to evaluate the quality of GAN-images, not the human detection capabilities. Their results show that StyleGAN images generated using the *truncation trick* are perceived as more realistic [54]. The truncation trick refers to how far away a latent style vector is sampled from the average latent style vector, which determines the amount of variety in the generated image. Furthermore, images of 64x64 resolution are harder to distinguish from real than 1024x1024 images. However, images of this small size do not occur often in real-world scenarios. Lastly, [27] examines human performance of detecting GAN-generated images as a direct comparison with algorithmic detection. Therefore, they train humans by showing many examples, and then test them with novel examples, resulting in an average classification score of 63.9% for the FFHQ vs StyleGAN$_{FFHQ}$ scenario. While this yields an indication for upper bound performance of humans, it does not examine performance of untrained humans, and factors that influence performance, making it difficult to project the results to real-world scenarios.

This work attempts to determine human performance under an approximation of real-world conditions. It differs from [27] since we do not pre-train participants, and measure the performance related to intermediate feedback. Moreover, it differs from [54] since we do not include any time constraints or training phase and evaluate more logical image resolutions. Lastly, we examine the influence of AI-experience on human performance, and image cues humans use to recognise generated images.

3. Methods & Experimental Setup

Figure 2 gives an overview of our method. Each component will be discussed next.

3.1. Datasets

Real images CelebA-HQ (CAHQ) [20] and Flickr-Faces-HQ (FFHQ) [21] are selected as datasets for real images. The first is a high-quality version of the original CelebA dataset [26], consisting of 30K front view facial pictures of celebrities. Note that high-quality refers to several processing steps as discussed by [20], yielding high-resolution and visually appealing images. The second is a dataset with 70K high-quality front view pictures of ordinary people, of which the first 30K are selected.

Generated (fake) images We use five datasets of generated images for evaluation under real-world conditions: 1) StarGAN$_{CAHQ}$ [6], 2) GLOW$_{CAHQ}$ [23], 3) ProGAN$_{CAHQ}$ [20], 4) StyleGAN$_{CAHQ}$ [21], and 5) StyleGAN$_{FFHQ}$ [21].

The first two datasets are provided by [13]. StarGAN and GLOW are conditional generative models that transform the style of an input image to some desired style. The datasets are created by taking a CAHQ image as input, randomly selecting a facial attribute out of a small set of attributes (*e.g.* hair color), and generating the corresponding image with either the StarGAN or GLOW model. GLOW is not a GAN but a flow-based deep generative model. The third dataset consists of images generated by ProGAN, an unconditional GAN that generates high-resolution facial images. We use the dataset provided by [20].

For the last two datasets, we use images generated by StyleGAN. StyleGAN could be regarded as the state-of-the-art GAN in terms of visual quality [54], strengthened by high-resolution images and some stylistic control over the output. We use two variants of StyleGAN images to evaluate cross-data performance. For the first variant, we use the dataset by [21]. From the available sets of images generated with different amounts of truncation, we select the set generated using $\psi = 0.5$. Note that these images are generated by a model trained on FFHQ images. There is no public StyleGAN$_{CAHQ}$ dataset, thus we generate images using a model pre-trained on CAHQ images (with $\psi = 0.5$). The motivation for selecting $\psi = 0.5$ and the creation of StyleGAN$_{CAHQ}$ are discussed in further detail in Section A.1 of the supplementary material.

For each dataset, we use 30K images, split into training (70%), validation (20%), and test (10%) sets. The amount of real and fake images seen during training and testing is equal. During training, images are rescaled to match the corresponding input layer size of both models.

3.2. Pre-processing

For pre-processing techniques we use high-pass filters, co-occurrence matrices, and color transformations, since these have recently been demonstrated to work well in CNN-generated image detection [13, 33, 24]. For each of these three categories, one or multiple variants have been experimented with. We select the best performing methods to be included in the results. A visualization of these methods is shown in Figure 3. **Res1** is a first-order derivative filter: $[1 \quad -1]$ [4, 14]. It is included as baseline high-pass filter. Similar to [13], this filter is applied in horizontal and vertical direction in parallel and the resulting channels are concatenated, yielding 6 image channels. **Res3** is a third-order derivative filter: $[1 \quad -3 \quad 3 \quad -1]$ [12, 13]. Again, it is applied similarly to Res1 and is equal to the *RES* filter used by [13]. Note that we have experimented with other implementations (*i.e.* applying the filter horizontally and vertically in sequence, yielding three channels) but these performed worse, thus choosing the implementation by [13]. **Cooc** calculates the co-occurrence matrix of an input image, similar to [33]. This is done by a matrix multiplication of the original image with its transpose, resulting in three image channels. **HSV** converts to the hue, saturation, value (HSV) color space, resulting in three image channels. This is inspired by [24], who use HSV and YCbCr color spaces, as discussed in Section 2. Our initial experiments showed better performance of HSV, so YCbCr is not considered.

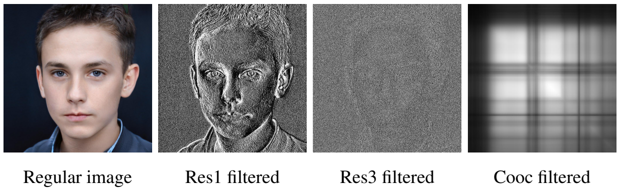
Regular image
Res1 filtered
Res3 filtered
Cooc filtered

Figure 3: Visualization of several pre-processing methods using a StyleGAN$_{FFHQ}$ image. Note that HSV is not visualized since it is not meaningful to display using RGB color conventions.

3.3. Model architectures

Based on the work of [13, 28, 42], we select Xception [7] and ForensicTransfer [13] as state-of-the-art model architectures for CNN-generated image detection. **Xception (X)** [7] is a deep CNN with depth-wise separable convolutions [7], inspired by Inception modules [47], and has shown good performance in multiple image forgery detection tasks [13, 28, 42], both for regular and compressed images. **ForensicTransfer (FT)** is a CNN-based encoder-decoder architecture, which learns to encode the properties of fake and real images in latent space, outperforming several other methods when combined with high-pass filtering the images, or using transfer learning for few-shot adaptation to unknown classes [13]. Images are classified as real if the *real partition* in latent space is more active than the *fake partition*, and vice versa. The training procedures for both models are described in Section A.2 of the supplementary material.

3.4. Evaluation

To examine the performance of detection methods under real-world conditions, we include five types of evaluation.

Default (fully aware) In the easiest setup, test images are created by the same generative model as train images and are from the same data distribution. These test images are not further manipulated. This setup gives an upper bound on the performance of a detection method, but has no correspondence to a real-world scenario. We test this for StyleGAN$_{CAHQ}$ and StyleGAN$_{FFHQ}$.

Cross-model (model-unaware) In a real-world scenario, many generative models exist and new models will be created in the future. In this setup, test images are generated by one or multiple different models than images in the training set. The detection model has no examples of similar test images. In our work, we evaluate the performance of 1) detecting StarGAN$_{CAHQ}$, GLOW$_{CAHQ}$, and ProGAN$_{CAHQ}$ when trained on StyleGAN$_{CAHQ}$, and 2) detecting StyleGAN$_{FFHQ}$ with $\psi \in [0.7, 1.0]$ when trained on $\psi = 0.5$.

Cross-data (data-unaware) In a real-world setting, numerous different datasets could be used to train a generative model, each with their own biases and pre-processing methods, which have a large impact on the generated images. Thus, it is needed to evaluate how detection models can generalize to unknown images used for training a generative model. In this setup, the data used for generating training images differs from the data used for generating test images. The model may be equal or different. In our work, we evaluate the performance of detecting StyleGAN$_{FFHQ}$ test images when trained on StyleGAN$_{CAHQ}$ images and vice versa.

Post-processing(-unaware) When images are uploaded to and downloaded from the internet, they are likely to undergo several types of post-processing, such as compression and resampling. On the other hand, images could be manipulated to make them less detectable, for example with blur and noise addition. In our work, we select two types of techniques, JPEG compression and Gaussian blurring, and evaluate how different amounts of post-processing influence the detection of StyleGAN$_{FFHQ}$ images. We evaluate several degrees ranging from hardly visible to clearly visible to the human observer.

In the wild This mimics a real-world scenario where a detection model has access to all currently known state-of-the-art models and encounters images generated by a newer model. In our case, one detection model is trained on multiple known sources (StarGAN$_{CAHQ}$, GLOW$_{CAHQ}$ and ProGAN$_{CAHQ}$), and evaluated on unknown sources of higher visual quality (StyleGAN$_{CAHQ}$ and StyleGAN$_{FFHQ}$).

4. Online survey

To examine how well humans can identify state-of-the-art fake images, we conduct a user study with 496 participants. We also study what influences their performance. A schematic overview of the survey is given in Figure 4. Each participant is randomly assigned to the control-group or feedback group. It then sees 18 images sequentially of varying resolutions and must decide for each image if it is real or fake. The whole process of survey design is described in Section B of the supplementary material.

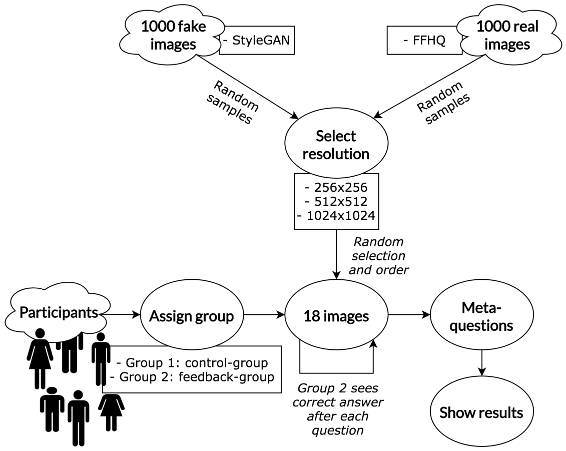

Figure 4: Each participant is randomly assigned to the control-group or feedback group. Then he/she sees 18 images sequentially of varying resolutions and must decide for each image whether it is real or fake.

Note that our experiments aim to estimate how well humans would perform in real-world scenarios, by examining several real-world factors that could influence this performance. These include 1) image resolution (measured with three resolutions), 2) how well people are trained (measured with a feedback and control group), and 3) AI-experience (measured with a question after completing the survey).

5. Results

5.1. Algorithmic detection

Table 1 shows results of training on StyleGAN$_{CAHQ}$ (left) and StyleGAN$_{FFHQ}$ (right) images, along with cross-model and cross-data performance. It is important to note that we show the accuracy per dataset, and not the average accuracy of real and fake images combined. For this reason, the performance on the real and fake datasets do not add up to 100%. For example, when a model is not able to detect fake images and classifies every image as real, the accuracy we report for the dataset with fake images will be 0%. This is done to get a better understanding of how well each model can detect generated images, since real images are, with few exceptions, detected with high accuracies.

Default (fully aware) In the *Default* columns of Table 1 we see that both Xception and ForensicTransfer have a nearly perfect performance for both fake images (StyleGAN$_{CAHQ}$/StyleGAN$_{FFHQ}$) and real images

Model		Default		Cross-model			Cross-data		Default		Cross-model		Cross-data	
Pre-process	Arch.	StyleG (CAHQ)	*CAHQ*	GLOW (CAHQ)	ProG (CAHQ)	StarG (CAHQ)	StyleG (FFHQ)	*FFHQ*	StyleG (FFHQ)	*FFHQ*	StyleG $\psi = 0.7$	StyleG $\psi = 1.0$	StyleG (CAHQ)	*CAHQ*
–	X	99.6	**99.8**	0.3	1.0	0.2	5.9	**99.8**	**99.9**	**100**	97.3	84.1	0.2	**100**
	FT	98.3	99.3	0.3	**88.9**	97.5	44.7	60.7	99.2	**100**	97.8	**94.3**	0.01	**100**
Res1	X	91.6	96.8	0.9	65.2	37.8	37.8	69.2	91.2	**100**	91.8	84.5	0.2	99.9
	FT	99.5	95.4	31.2	**88.9**	**100**	**90.2**	28.4	91.3	90.9	89.1	83.5	8.8	93.9
Res3	X	72.2	45.6	**49.9**	65.7	57.8	62.1	39.8	54.5	98.6	54.4	51.2	2.5	98.7
	FT	93.3	89.9	36.6	65.2	99.5	87.8	36.7	72.5	75.4	70.4	67.1	**30.8**	84.0
Cooc	X	95.0	96.4	2.1	12.9	2.8	31.2	95.3	93.6	97.4	63.5	18.6	13.7	98.7
	FT	79.6	77.8	2.3	37.3	26.8	32.4	83.8	80.4	91.5	52.9	23.7	20.6	91.8
HSV	X	**99.9**	**99.8**	3.0	63.6	12.2	44.7	87.6	**99.9**	99.9	**98.3**	87.9	0.2	99.9
	FT	93.7	97.9	33.0	79.5	81.8	46.8	56.8	98.7	99.9	96.7	91.3	0.1	**100**

Table 1: Evaluation of default, cross-model, and cross-data performance. The first setup uses StyleGAN$_{CAHQ}$ ($\psi = 0.5$) as a training dataset and tests on 1) StyleGAN$_{CAHQ}$ images (default evaluation), 2) GLOW$_{CAHQ}$, ProGAN$_{CAHQ}$ and StarGAN$_{CAHQ}$ images (cross-model), and 3) StyleGAN$_{FFHQ}$ images (cross-data). The second setup uses StyleGAN$_{FFHQ}$ ($\psi = 0.5$) as a training dataset and tests on 1) StyleGAN$_{FFHQ}$ images (default evaluation), 2) StyleGAN$_{FFHQ}$ ($\psi = 0.7$ and $\psi = 1.0$) images (cross-model), and 3) StyleGAN$_{CAHQ}$ images (cross-data). On the left side, we denote the type of pre-processing and model architecture, where X denotes Xception and FT denotes ForensicTransfer. We have also abbreviated StyleG(AN)$_{CAHQ}$, ProG(AN)$_{CAHQ}$, and StarG(AN)$_{CAHQ}$ for visualization purposes. Real image datasets are *cursive* (CAHQ and FFHQ). Best accuracies per dataset (*i.e.* column) are **bold**. Accuracies are averaged over 5 runs.

(CAHQ/FFHQ). The third-order derivative filter seems to harm the performance the most for both model architectures and both datasets.

Cross-model (model-unaware) For the cross-model setup, we see in Table 1 that ForensicTransfer retrieves high performance for detecting ProGAN$_{CAHQ}$ (88.9%) and StarGAN$_{CAHQ}$ (100%) images. For ProGAN$_{CAHQ}$ and StarGAN$_{CAHQ}$, the first order derivative filter yields slightly better results than using no filter. In our second setup, we train on StyleGAN$_{FFHQ}$ and evaluate cross-model (parameter) evaluation. Note that this type of cross-model evaluation refers to the same model ($\psi = 0.5$), but using another truncation ($\psi = 0.7$ and $\psi = 1.0$) for generating images, results in differences between training and testing images. The larger the difference between the ψ values for training and testing, the lower the accuracy for detecting fake images. We also see this trend for different pre-processing techniques. For example, the performance for using the co-occurence matrix results in a drop of 45% for Xception and 29% for ForensicTransfer.

Cross-data (data-unaware) In Table 1 for both Xception as well as ForensicTransfer there is a trade-off between detecting fake and real images. Xception labels all images (99.8%) as true when no pre-processing is used. In the same scenario, ForensicTransfer is able to detect fake StyleGAN$_{FFHQ}$ images in 45% of the cases, but as a consequence detecting FFHQ as real drops to 60%. Using first or third order derivative filters for ForensicTransfer increases the performance for generated images, but decreases the performance for real images. For cross-data performance in our second setup, there is a an increase in performance of detecting StyleGAN$_{CAHQ}$ images, when using ForensicTransfer together with third order derivative filters or the co-occurrence matrix. This increase from 0 to 20-30% is still far from a good performance. Based on our results, there is no clear model or pre-processing method that stands out as best. ForensicTransfer has relatively high cross-model and cross-data performance, and seems to benefit slightly from high-pass filters, at the cost of a small drop in default performance. However, high-pass filters decrease performance for Xception, which seems to benefit slightly from HSV transformation.

Post-processing(-unaware) This evaluation consists of three levels of Gaussian blur, from a standard normal distribution with different kernel sizes, and three levels of JPEG compression using different quality factors. The results are presented in Table 2. For each type of evaluation, the difference between training and testing images increases gradually to the right (*e.g.* QF=90 is almost no compression, and QF=10 is severe compression). Without pre-processing techniques, Xception is much more robust to blur and compression, and shows nearly no drop in performance for the smallest amounts. For example, when using a 3x3 kernel for Gaussian blur Xception is able to detect StyleGAN$_{FFHQ}$ images with 98.9% accuracy, while ForensicTransfer only detects 18.1% of the cases. Again, ForensicTransfer seems to benefit slightly from high-pass filters, while these deteriorate Xception performance. In this setup, HSV does not benefit Xception as much, making performance on cross-model and post-processing worse. Cooc shows no evident pattern in performance.

In the wild As shown in Table 3, cross-model detection is still low when training on images generated by different models. When examining the average accuracy of real images and unseen generated images (StyleGAN$_{CAHQ}$/StyleGAN$_{FFHQ}$), we observe that Xception without pre-processing performs best (62.4%), followed by X-Cooc (61.2%) and FT-Res1 (60.5%). The other methods

Model		Default	Post-processing					
			Gaussian blur			JPEG compression		
Pre-proc.	Arch.	StyleG (FFHQ)	3x3 kernel	9x9 kernel	15x15 kernel	QF=90	QF=50	QF=10
–	X	**99.9**	**98.9**	1.7	0.0	**99.5**	**95.8**	28.4
	FT	99.2	18.1	0.0	0.0	0.1	0.2	0.1
Res1	X	91.2	71.8	0.2	0.0	43.0	5.3	0.4
	FT	91.3	79.5	21.9	12.4	8.0	1.4	0.9
Res3	X	54.5	11.0	0.8	1.3	16.9	6.3	5.7
	FT	72.5	66.4	**65.7**	**64.1**	17.7	5.9	3.8
Cooc	X	93.6	92.3	39.6	0.8	93.0	91.9	**75.5**
	FT	80.4	79.1	60.4	56.1	72.9	51.3	6.4
HSV	X	**99.9**	91.9	1.9	0.1	79.0	55.3	11.2
	FT	98.7	17.7	0.0	0.0	0.1	0.0	0.0

Table 2: Evaluation of post-processing evaluation techniques using StyleGAN$_{FFHQ}$ as a training dataset and testing on 1) StyleGAN$_{FFHQ}$ images (default), 2) StyleGAN$_{FFHQ}$ images with different amounts of Gaussian blur (three kernel sizes), and 3) StyleGAN$_{FFHQ}$ images with different amounts of JPEG compression (three quality factors). The layout is similar to the previous table.

Model		Default					Cross-model		
Pre-proc.	Arch.	*FFHQ*	*CAHQ*	StarG (CAHQ)	GLOW (CAHQ)	ProG (CAHQ)	StyleG (CAHQ)	StyleG (FFHQ)	Avg*
–	X	**99.9**	99.9	**100**	**100**	**99.7**	49.5	0.1	**62.4**
	FT	89.3	99.1	**100**	99.9	78.8	7.9	10.4	51.7
Res1	X	99.6	99.7	97.6	98.5	97.6	2.9	0.2	50.6
	FT	65.8	86.6	**100**	**100**	84.4	50.0	39.5	60.5
Res3	X	43.0	41.0	83.5	81.6	78.8	52.8	58.0	48.7
	FT	49.7	43.6	**100**	**100**	76.3	76.3	45.6	53.8
Cooc	X	97.3	96.4	99.2	99.6	94.3	50.1	0.8	61.2
	FT	27.6	15.7	88.4	85.7	86.8	**86.3**	**69.3**	49.7
HSV	X	**99.9**	**100**	**100**	**100**	**99.7**	12.5	0.02	53.1
	FT	91.2	94.3	**100**	**100**	82.8	28.3	6.1	55.0

Table 3: Evaluation of 'in the wild' scenario. The models are trained on two datasets of real images (CAHQ and FFHQ) and three datasets of generated images (StarGAN$_{CAHQ}$, GLOW$_{CAHQ}$, and ProGAN$_{CAHQ}$). They are tested on two versions of StyleGAN images, that are not seen during training. The layout is similar to the previous tables. * Average of FFHQ, CAHQ, StyleGAN$_{CAHQ}$ and StyleGAN$_{FFHQ}$, with an equal amount of real and generated images.

	Total avg	Intermediate Feedback		Image resolution			AI-experience	
		No	Yes	1024^2	512^2	256^2	Little	Much
Real images	74.8	74.8	74.9	78.0	75.0	71.6	66.4	82.2
Fake images	65.3	60.4	70.9	76.5	65.5	54.0	57.1	72.6
All images	70.1	67.6	72.9	77.2	70.2	62.8	61.7	77.4

Table 4: Average accuracies of labelling real and fake images among 1) all participants, 2) participants without/with intermediate feedback, 3) images of different resolution, and 4) participants with little/much AI-experience.

yield an average accuracy close to 50% (with a balanced amount of real and generated images). Lastly, some pre-processing methods seem to decrease default performance.

5.2. Human performance

In Table 4, the results of our survey with 496 participants are presented. Note that in all tables, *real* refers to an authentic image from the FFHQ dataset, while *fake* refers to a generated image from the StyleGAN$_{FFHQ}$ dataset. Out of all images, 70.1% are labelled correctly. For real images, the average accuracy is 74.8%, while for fake images it is 65.3%. In the following, we examine the results of 1) intermediate feedback, 2) resolution, 3) AI-experience, and 4) upper and lower bound. The cues humans use to distinguish these images are analysed in Section C of the supplementary results.

Feedback Table 4 shows the average results of the group with intermediate feedback (N=233) and the group without (N=263). As shown, performance on real images is nearly identical, while performance on fake images is roughly 10% higher, suggesting that participants can better learn to recognize fake images when receiving intermediate feedback. This is supported by an independent samples t-test, yielding a p-value of < 0.005 (with a t-statistic of 3.3). Note that only 18 images are evaluated in total, and this effect might be larger with more images. As a sanity check, the distribution of AI-experience among both groups is examined, which is nearly equal.

Image Resolution When comparing performance with different image resolutions, Table 4 shows that average detection accuracy of real and fake images decreases when images of lower resolution are presented. However, for real images this decrease is small, while for fake images the difference between highest and lowest resolution is 22.5%. Note that each participant sees 3 real and 3 fake images of each resolution, but the selected images and order of presenting are completely random, excluding the possible influence of learning. The differences between resolution are tested with a one-way ANOVA test, yielding a p-value of $<< 0.001$ (with F-statistic 49.7). When performing post-hoc evaluation, we see that all group means differ much more than the standard error, suggesting that a lower image resolution makes an image significantly harder to classify, for the resolution tested in our survey. This is likely due to details and artefacts being less visible on smaller scales.

AI-experience Table 4 shows the detection accuracies among two groups of participants with different levels of AI-experience. The first group (N=259) has *much* AI-experience, and consists of AI-students, teachers, and professionals. The second group (N=218) has *little* AI-experience and consists of all others. As shown, the average level of AI-experience within a group seems to have a large influence on detection performance. For real and fake images combined, the difference between little and much AI-experience is roughly 15%. This difference is supported by an independent samples t-test, yielding a p-value of $<< 0.001$ (with a t-statistic of 10.7). Note that people with little AI-experience recognize fake images correctly in 57.1% of the cases, which is slightly better than random.

Upper and Lower Bound The upper and lower bound of human performance is examined in Table 5. This is done

	Upper bound		Lower bound	
	Much AI experience	Little AI experience	Much AI experience	Little AI experience
Real images	85.4	69.6	78.9	61.0
Fake images	86.7	76.6	54.9	37.0
All images	86.0	73.1	66.9	49.0

Table 5: Average accuracies of labelling real and fake images in different setups, ranging from the most easy setup (left columns), which denote average performance of participants with feedback for 1024x1024 images, to the most difficult setup (right columns), which denote average performance of participants without feedback for 256x256 images. Within both groups, the performance of participants with little or much AI experience is shown.

by evaluating the easiest scenario (*i.e.* with feedback and 1024-res. images) and hardest scenario (*i.e.* without feedback and 256-res. images). Within both scenarios, the difference between little and much AI-experience is examined. As becomes clear in Table 5, the highest average detection accuracy for fake images is 86.7% and the lowest is 37.0%.

Comparison to algorithmic performance A comparison of algorithmic and human performance on StyleGAN$_{FFHQ}$ data is presented in Figure 5. The *upper bound* scenario approximates the most easy setup for both. For algorithmic detection this is the case when the model is trained and tested on the same dataset (StyleGAN$_{FFHQ}$). For humans this is the upper bound as shown in Table 5. The *realistic* scenario approximates real-world conditions. For algorithmic detection, we formulate this as the 'in the wild' scenario as shown in Table 3, where only StyleGAN$_{FFHQ}$ results are used. For humans it includes three variants (displayed from left to right in Figure 5): 1) an *optimistic* realistic scenario, assuming humans have average AI-experience, learn to recognise fake images with feedback, and mainly see high-resolution images (512 and 1024), 2) an *average* realistic scenario (estimated by the average of all survey results), and 3) a *pessimistic* realistic scenario, assuming humans have low AI-experience, do not receive feedback, and see images of all resolutions. Lastly, the *lower bound* scenario presents the results of the most difficult setup. For humans this is the lower bound as shown in 5. For algorithmic detection, the lower bound is set at 50%, which is a random guess in our two-class classification task with balanced class sizes. Note that its performance in the realistic scenario is already close to 50%.

6. Conclusion & Discussion

Our work has evaluated two state-of-the-art models for detecting CNN-generated images, and has proposed three types of evaluation, along with an 'in the wild' setup, for mimicking real-world conditions in which such detection models will be used. Furthermore, we evaluated the benefits of several commonly used pre-processing methods.

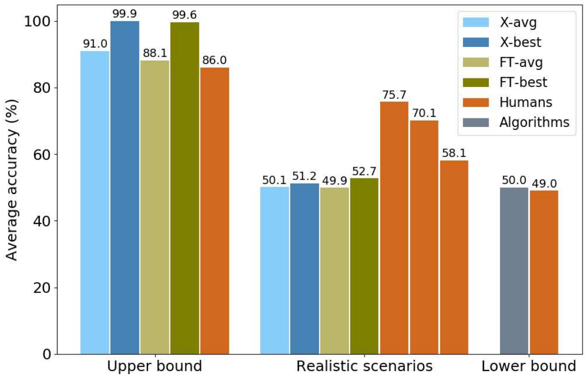

Figure 5: Comparison of algorithm and human performance in different scenarios.

Based on our algorithmic experiments, we can conclude that performance in the easiest (default) scenario doesn't generalize well to other evaluation scenarios. ForensicTransfer seems more robust in cross-model performance, whereas Xception seems more robust in post-processing performance. Unfortunately, there is no single type of pre-processing that increases performance in multiple scenarios, and an increase in one evaluation setup is often paired with a decrease in other setups. Furthermore, the benefits of pre-processing methods are not guaranteed for both models; *i.e.* high-pass filters work much better for ForensicTransfer than for Xception. Our results emphasize the importance of evaluating multiple scenarios. We emphasize the need for a benchmark dataset including images generated by multiple models, such that these types of evaluation can be performed and compared to related work.

The results of the survey suggest that humans have trouble recognizing state-of-the-art fake images, which are correctly classified in roughly two-thirds of the cases. Our results suggest that the capability of detecting fake images could be influenced by several factors that may be of importance in real-world scenarios, such as AI-experience, image resolution, and feedback. When combining these factors, we see large differences between the best and the worst case (86.7% as opposed to 37.0% of fake images correctly recognized). These results emphasize the need for algorithmic detection methods to support humans in recognizing such images, as well as more research into the factors that influence human performance. Based on our comparison between algorithms and humans, we see that humans perform better than our models in the realistic scenario. However, from our upper bound performance we can conclude that models can outperform humans when trained and employed correctly. We encourage future work to pay more attention to extensiveness of evaluation which will result in more robust models for real-world scenarios.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *WIFS*, 2018. 2

[2] Michael Albright and Scott McCloskey. Source generator attribution via inversion. In *CVPR Workshop on Media Forensics*, pages 96–103, 2019. 3

[3] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *ACM IHMS*, 2016. 2

[4] Bolin Chen, Haodong Li, and Weiqi Luo. Image processing operations identification via convolutional neural network. *arXiv preprint arXiv:1709.02908*, 2017. 4

[5] Jiansheng Chen, Xiangui Kang, Ye Liu, and Z Jane Wang. Median filtering forensics based on convolutional neural networks. *IEEE Signal Processing Letters*, 22(11):1849–1853, 2015. 3

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 2, 4

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 2, 3, 4

[8] Samantha Cole. We are truly fucked: Everyone is making ai-generated fake porn now [blog post], 2018. Accessed April 17, 2020. 1

[9] Samantha Cole. Deepfake of boris johnson wants to warn you about deepfakes [blog post], 2019. Accessed April 17, 2020. 1

[10] Samantha Cole, Emanuel Maiberg, and Jason Koebler. This horrifying app undresses a photo of any woman with a single click [blog post], 2018. Accessed April 17, 2020. 1

[11] Davide Cozzolino, Diego Gragnaniello, and Luisa Verdoliva. Image forgery detection through residual-based local descriptors and block-matching. In *ICIP*, 2014. 3

[12] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In *ACM IHMS*, 2017. 2, 3, 4

[13] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510*, 2018. 1, 2, 3, 4

[14] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *TIFS*, 7(3):868–882, 2012. 3, 4

[15] Marcus Gilmer. As concern over deepfakes shifts to politics, detection software tries to keep up [blog post], 2019. Accessed April 17, 2020. 1

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 1, 2

[17] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017. 2

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2

[19] Xiaohan Jin, Ye Qi, and Shangxuan Wu. Cyclegan face-off. *arXiv preprint arXiv:1712.03451*, 2017. 2

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 2, 4

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1, 2, 4

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv:1912.04958*, 2019. 1, 2

[23] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018. 2, 4

[24] Haodong Li, Bin Li, Shunquan Tan, and Jiwu Huang. Detection of deep network generated images using disparities in color components. *arXiv preprint arXiv:1808.07276*, 2018. 2, 3, 4

[25] Haodong Li, Weiqi Luo, Xiaoqing Qiu, and Jiwu Huang. Identification of various image operations using residual-based features. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(1):31–45, 2018. 3

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15:2018, 2018. 4

[27] Zhengzhe Liu, Xiaojuan Qi, Jiaya Jia, and Philip Torr. Global texture enhancement for fake face detection in the wild. In *CVPR*, 2020. 3

[28] Francesco Marra, Diego Gragnaniello, Davide Cozzolino, and Luisa Verdoliva. Detection of gan-generated fake images over social networks. In *MIPR*, 2018. 1, 3, 4

[29] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *MIPR*, 2019. 3

[30] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *WACVW*, 2019. 2

[31] Scott McCloskey and Michael Albright. Detecting gan-generated imagery using color cues. *arXiv preprint arXiv:1812.08247*, 2018. 2

[32] Huaxiao Mo, Bolin Chen, and Weiqi Luo. Fake faces identification via convolutional neural network. In *ACM IHMS*, 2018. 3

[33] Lakshmanan Nataraj, Tajuddin Manhar Mohammed, BS Manjunath, Shivkumar Chandrasekaran, Arjuna Flenner, Jawadul H Bappy, and Amit K Roy-Chowdhury. Detecting gan generated fake images using co-occurrence matrices. *Electronic Imaging*, 2019(5):532–1, 2019. 3, 4

[34] Sophie J Nightingale, Kimberley A Wade, and Derrick G Watson. Can people identify original and manipulated photos of real-world scenes? *Cognitive research: principles and implications*, 2(1):30, 2017. 3

[35] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 2

[36] Simon Parkin. The rise of the deepfake and the threat to democracy [blog post], 2019. Accessed August 5, 2019. 1

[37] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2

[38] Tomáš Pevny, Patrick Bas, and Jessica Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on information Forensics and Security*, 5(2):215–224, 2010. 3

[39] Jon Porter. Another convincing deepfake app goes viral prompting [blog post], 2019. Accessed April 17, 2020. 1

[40] Nicolas Rahmouni, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Distinguishing computer graphics from natural images using convolution neural networks. In *WIFS*, 2017. 2

[41] Yuan Rao and Jiangqun Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In *WIFS*, 2016. 3

[42] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, 2019. 3, 4

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 3

[44] Victor Schetinger, Manuel M Oliveira, Roberto da Silva, and Tiago J Carvalho. Humans are easily fooled by digital images. *Computers & Graphics*, 68:142–151, 2017. 3

[45] Kenneth Sullivan, Upamanyu Madhow, Shivkumar Chandrasekaran, and BS Manjunath. Steganalysis for markov cover data with applications to images. *IEEE Transactions on Information Forensics and Security*, 1(2):275–287, 2006. 3

[46] Kenneth Sullivan, Upamanyu Madhow, Shivkumar Chandrasekaran, and Bangalore S Manjunath. Steganalysis of spread spectrum data hiding exploiting cover memory. In *Security, Steganography, and Watermarking of Multimedia Contents VII*, 2005. 3

[47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 4

[48] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, 2020. 3

[49] Xin Yang, Yuezun Li, Honggang Qi, and Siwei Lyu. Exposing gan-synthesized faces using landmark locations. In *ACM IHMS*, 2019. 2

[50] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, 2017. 2

[51] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018. 2

[52] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *CVPR*, 2019. 3

[53] Lilei Zheng, Ying Zhang, and Vrizlynn LL Thing. A survey on image tampering and its detection in real-world photos. *Journal of Visual Communication and Image Representation*, 58:380–399, 2019. 3

[54] Sharon Zhou, Mitchell Gordon, Ranjay Krishna, Austin Narcomey, Li F Fei-Fei, and Michael Bernstein. Hype: A benchmark for human eye perceptual evaluation of generative models. In *NeurIPS*, 2019. 3, 4

[55] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2