Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations

Falko Matern Christian Riess Marc Stamminger
Friedrich-Alexander University Erlangen-Nuremberg
falko.matern@fau.de christian.riess@fau.de marc.stamminger@fau.de

Abstract

High quality face editing in videos is a growing concern and spreads distrust in video content. However, upon closer examination, many face editing algorithms exhibit artifacts that resemble classical computer vision issues that stem from face tracking and editing. As a consequence, we wonder how difficult it is to expose artificial faces from current generators? To this end, we review current facial editing methods and several characteristic artifacts from their processing pipelines. We also show that relatively simple visual artifacts can be already quite effective in exposing such manipulations, including Deepfakes and Face2Face. Since the methods are based on visual features, they are easily explicable also to non-technical experts. The methods are easy to implement and offer capabilities for rapid adjustment to new manipulation types with little data available. Despite their simplicity, the methods are able to achieve AUC values of up to 0.866.

1. Introduction

In the past two years, algorithms for automated face editing like Face2Face and Deepfakes went through the public media [36, 16]. In particular, questions whether "We can (from now on) trust any video at all?" and "How can I, as a normal citizen, detect Deepfakes?" were regularly raised by journalists. While we would like to expressedly abstain from the sometimes alarmistic tone in these questions, it falls within the field of image and video forensics to provide technical answers for detecting the state of the art in automated video editing or generation of faces.

The main research progress in the creation of artificial faces is performed in the computer graphics and computer vision communities. Particular focus of this research is to create visually plausible video content. Impressive progress has recently been made, as is shown in Fig. 1. This figure shows three example faces. From left to right, these faces are an artificial image from ProGAN [18], a real photograph, and an artificial image from Glow [21]. It is reasonable to acknowledge that even a well-trained viewer has difficulties to distinguish artificial from real faces.

Figure 1. Only one image shows an actual person. The images on the left (ProGAN [18]) and right (Glow [21]) are generated. The image in the middle is a crop from the CelebA dataset [27].

However, having such automated tools is just one component in creating a manipulation. The goal of a malicious manipulation is to convey a (semantic) message with the video that is not communicated in the original material. This imposes constraints on the manipulator, e.g., to arrange specific people in a specific scene, such that the visual message fits the overall story of the manipulator. Additionally, the material has to be consistent with side information that may be available to the viewer or an independent analyst. From a technical perspective, this requires highly robust methods for video editing: it is reasonable to assume that a manipulator has only few candidate scenes in which he or she can convey the intended message, and the method for editing a face has to create a visually plausible result in that particular situation.

In this work, we survey landmark works in automatic video generation. While current computer vision and computer graphic works clearly exhibit excellent results in relatively free scenarios, we argue that most of these methods have limitations if applied in specific, pre-defined scenarios that may be relevant to a manipulator. This leads to characteristic artifacts in the generated content. The good news is, from the perspective of the forensic analyst, that it does not necessarily require sophisticated tools to detect such artifacts. We demonstrate this by several visual features that focus on the eyes, teeth, facial contours. In the future, we expect these visual features to be diminished in novel computer vision algorithms, such that probably only statistical forensics tools have a reasonable chance to de-

978-1-7281-1392-0/19/$31.00 ©2019 IEEE
DOI 10.1109/WACVW.2019.00020

tect facial manipulations. However, until sufficiently general statistical video forensics methods are developed, these visual features can also serve as an easy-to-implement, well feasible bridge technology to detect current manipulations.

This paper is organized as follows. In Sec. 1.1, we review related work in image forensics. In Sec. 2, we present a selection of methods for automated generation and editing of faces. In Sec. 3, we discuss artifacts that might arise from common challenges of these methods. Experiments and results with visual features are presented in Sec. 4. The findings are discussed and concluded in Sec. 5.

1.1. Related Work

Traditional methods in image forensics search either for physical or statistical image artifacts to validate the content of an image. An overview of such methods can be found, e.g., in recent surveys or books [32, 10]. Examples for physics-based methods are inconsistencies in lighting [19] or reflections [17, 29]. Recent statistical methods form statistical fingerprints on the residuals of an image to detect manipulations [11, 6, 7], validate noise priors from metadata [14], or learn specific manipulation traces, such as recoloring [3] or recompression [28].

With the availability of the face swap app, forensics researcher investigated specific solutions to detect such manipulations. For example, Zhang *et al*. proposed a bag of words classifier [40], and Zhou *et al*. proposed a two-stream neural network for this task [41]. Other recent methods aim to distinguish computer generated from natural images [31, 5]. These methods for example rely on a convolutional neural network (CNN) or the detection of small fluctuations caused by the human pulse to tell the content apart. For the detection of faces completely generated by deep-learning methods Tariq *et al*. [35] propose a CNN and Li *et al*. [24] use statistical differences in color components to distinguish the images. Detectors for Deepfakes mostly rely on deep-learning. For example, the work by Rössler *et al*. [33] proposes a large dataset containg Face2Face manipulations and the detection based on CNNs. Afchar *et al*. [1] propose two CNNs trained on scene content instead of noise to detect Deepfake and Face2Face manipulations. Li *et al*. [25] propose to expose fake faces by detecting eye-blinking which according to the authors tends to be missing in Deepfake videos. Güera and Delp [13] propose a recurrent neural network incorporating temporal information to detect Deepfake videos. Another approach by Li *et al*. [26] proposes to train a CNN to detect warping artifacts for the detection of Deepfake videos.

2. Manipulation Methods

Research on generation and manipulation of images and videos gained considerable momentum with the advent of deep-learning. The current progress in the field is so quick that a complete overview is beyond the scope of this work. Instead, we highlight example methods that we consider highly relevant in the context of image forensics.

2.1. Generated Faces

Generative adversarial networks (GAN) and variational autoencoders (VAE) [22] are a powerful tool for generating image content [12]. However, early implementations produce images of low resolution that oftentimes exhibit blur, which allows to easily identify them as generated. Karras *et al*. [18] overcame this limitation by demonstrating the generation of high-resolution images of up to 1024×1024 pixels in the so-called ProGAN. The images are generated by progressively growing the generator and discriminator of the model. The results show convincing faces generated from random numbers. Another way of generating such images are flow-based generative models [8, 9]. Kingma and Dhariwal [21] propose a flow-based method able to generate convincing faces at a resolution of 256×256 pixels by incorporating invertible 1×1 convolutions in the so-called Glow network. Other methods generate images from labels as input and are able to produce impressive results. Isola *et al*. [15] propose a general-purpose solution for such image-to-image translation problems, but the resulting images have a relatively low resolution. The method is improved by Wang *et al*. [39] by incorporating multi-scale generators and discriminators. The results have a resolution of up to 2048×1024 pixels. Wang *et al*. [38] further extend this method to video-to-video translation problems with impressive results.

2.2. Manipulation of Facial Attributes

Another popular research topic is the manipulation of certain facial attributes or the reenactment of faces. The method Face2Face by Thies *et al*. [37] is able to transfer facial expressions from a source to a target video in real-time. The method relies on fitting a 3-D morphable face model and estimating illumination which is approximated by spherical harmonic coefficients. The final result is rendered into the target video. A state-of-the-art method that uses similar methods can be found in [42]. Newer methods mostly rely on deep-learning models. The method proposed by Kim *et al*. [20] extends these approaches by allowing to manipulate the 3-D head position, head rotation, face expression, eye gaze, and eye blinking using a generative neural network. The method by Bansal *et al*. [2] is able to transfer video content from one domain to another, which can be applied to face-to-face scenarios. Some methods focus on changing certain facial attributes such as hair-color or age in single images. The method by Choi *et al*. [4] combines multiple domains into a single model. The method is therefore able to alter multiple attributes and facial expressions with one model. The method by Pumarola *et al*. [30] is able

to animate facial expressions in a convincing manner, given a single input image.

2.3. Deepfakes

Face swapping is the replacement of one face with another. This type of manipulation is commonly offered by smartphone apps. The generation of such data and its possible detection is covered by Zhou *et al*. [41]. Deepfakes are in principle a convincing face swap for videos. The face in a video is replaced by a different one, while the remaining original scene content and the original facial expressions are preserved. Examples of such manipulations are shown in Fig. 8. Such content can be created with two auto-encoders which are trained for the two faces. The weights for the encoders are shared, whereas each decoder is trained individually. This enables the use of both decoders with the encoder. The generation of Deepfake videos is explained in more detail in related work that aims to detect such manipulations [1, 13]. However, the state of the available tools and methods is dynamic. The methods are not strictly defined and might not necessarily be scientifically covered as there are multiple projects[1] [2] driven by community development. Since the methods are made publicly available with step-by-step instructions, they are usable by a wide audience and manipulated videos with impressive results are shared in social media.

3. Manipulation Artifacts

The results of methods as described in Sec. 2 are impressive, and research progress makes it visually increasingly difficult to see differences to real images. Nevertheless, some visual artifacts can still be identified, which is shown for a selection of example manipulation methods. The artifacts are categorized into different types of long-standing computer vision problems that are still not fully solved.

3.1. Global Consistency

Generative methods from Sec. 2.1 can be used to smoothly interpolate between given faces using the latent space of the underlying trained models. They can also be applied to generate new faces from random numbers. Two examples of such "hallucinated" faces are shown in Fig. 1. When the methods are used to interpolate between faces, the data points for generation are meaningful and the results usually plausible. For the generation of new faces, the data points supporting the interpolation of the image are random and not necessarily meaningful. While the results can generally be described as a qualitatively harmonious mixture of different faces, they seem to lack global consistency. A lot of samples can be observed to have a high variance in color between the left and right eye. Examples taken from three different methods are shown in Fig. 2. The phenomenon of differently colored irises is called heterochromia and is in reality rare for humans [34]. The severeness of this artifact in generated faces varies and not all samples are necessarily affected.

Figure 2. Samples of different methods displaying difference between color of the left and right eye. (Top to bottom: [18], [21], image taken from [39])

Figure 3. Example from FaceForensics [33] showing shading artifacts arising from illumination estimation and imprecise geometry of the nose.

3.2. Illumination Estimation

For re-rendering a face with different attributes, the incident illumination has to be transfered from the original image to the forgery. For methods such as Face2Face [37] the process of estimating geometry, estimating illumination and rendering is explicitly modeled. In deep-learning based methods, this model is usually implicitly learned from the data. A wrong or imprecise estimation of the incident illumination will lead to related artifacts.

Diffuse reflection is usually convincingly reproduced. Especially for manipulations that are generated with deep-learning techniques, we were not able to spot related artifacts. In some cases of Face2Face manipulations, shading artifacts can be spotted. The artifacts usually appear in the area of the nose, where one side is rendered too dark. An example of this artifact with a comparison to the original image is shown in Fig. 3. We hypothesize that these artifacts may be caused by the limited illumination model of Face2Face, which does not take interreflections into account.

[1] https://github.com/iperov/DeepFaceLab
[2] https://github.com/shaoanlu/faceswap-GAN

Figure 4. Deepfake examples showing missing reflection details in eyes. Samples from the dataset proposed in Sec. 4.1.

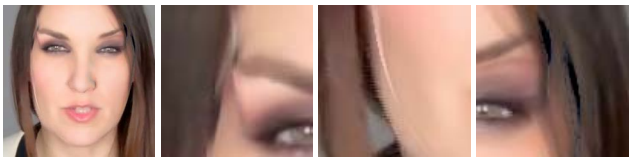


Figure 5. Examples from FaceForensics [33] showing artifacts from imprecise geometry estimation.

Specular reflection in faces is most noticable in the eyes. A lot of samples generated by Deepfake techniques show unconvincing specular reflections. Reflections in the eyes are either missing or appear simplified as a white blob. This artifact leads to an overall dull appearance of the eyes. Examples with comparisons to the unaltered content are shown in Fig. 4.

3.3. Geometry Estimation

Facial geometry has to be estimated to generate the manipulations. Analogously to the case of illumination, Face2Face models the geometry estimation explicitly by fitting a morphable model to images. Deep-learning based techniques implicitly learn the underlying model from data.

For the Face2Face data, we can spot artifacts arising from an imprecise estimation of the underlying geometry. The original image is overlaid with a face mask. If the geometry estimation is not perfect, artifacts along the boundary of the mask appear. This often becomes apparent in the area of the nose (Fig. 3), around the occlusion border of the face, and the eyebrows. Such artifacts are shown in Fig. 5. The imprecise geometry leads to blending artifacts, which appear as strong edges or high-contrast spots. Additionally, partly occluded face parts such as strands of hair, are not modeled correctly and can lead to "holes" as shown in the image to the right in Fig. 5. On Deepfake samples that are currently circulating in social media, we can frequently spot that some geometry is missing. Specifically, teeth are often not modeled at all. This is clearly visible in a lot of videos where teeth appear as a single white blob instead of individual teeth. An example is shown in Fig. 6.

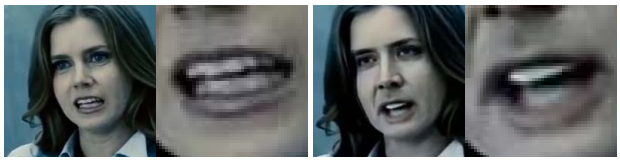


Figure 6. Missing geometry in Deepfakes. Teeth are generated as a structureless white blob. Samples from the dataset in Sec. 4.1.

4. Classification Based on Visual Artifacts

The visual appearance of artifacts is not always as obvious as in the shown examples. However, we show that relatively simple features can be used to model these observations. These features can be used to detect generated or manipulated faces. Specifically, we propose algorithms to detect completely generated faces, Deepfakes which are currently circulating in social media and images manipulated by Face2Face. In this Section, we first describe the generation of the required data and the methods to build descriptors for automated detection. Finally, we present an evaluation of the proposed methods.

4.1. Datasets

Evaluation is performed on a dataset of generated faces, on collected Deepfakes and on images showing Face2Face manipulations.

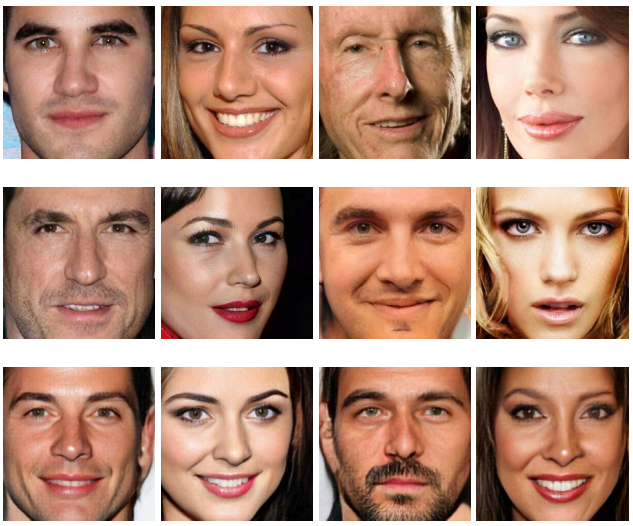


Figure 7. Example crops of the test dataset for generated faces. Top to bottom: CelebA [27], ProGAN [18], Glow [21].

Generated Faces. We collect a dataset for the classification of generated faces. For faces of real persons, we extract face crops from the CelebA dataset [27]. To generate samples which qualitatively match the results of the generation methods, we only extract crops with a height of at least 400 pixels. Additionally, we only take faces into account that

are classified as frontal and with high confidence. For generated faces, we extract face crops from the provided samples[3] of ProGAN [18]. The generated images of the method have a resolution of 1024×1024 pixels. A face detector is applied to crop the images to the face region, removing the background. After cropping, the image height varies between 446 and 642 pixels. We extract 2000 samples for the real and generated class as training and development data and 1000 samples for each class as testing data. As additional test data and to evaluate generalization to a different method, we generate 1000 random faces with the demo code[4] of Glow [21]. The generated images have a resolution of 256×256 pixels. After cropping, the height varies between 124 and 180 pixels. Some example images contained in the test data are shown in Fig.7. For classification, we denote generated faces as the positive class, and original faces as the negative class.

Deepfakes. We create a dataset to represent forgeries of the quality in which they can currently be found in social media. To that end, we collect Deepfake videos on YouTube. To simplify the dataset generation and to align scene content between classes, we select videos showing side-by-side comparisons between altered and unaltered material. We download four collection videos[5] containing multiple scenes and three videos[6] containing a single scene each. Although the videos show side-by-side comparisons, the origin and prior processing of the individual source videos is unknown. Dependent on availability, we download the videos as 1080p or 720p mp4 files. To presort the samples, the videos are split into a real and fake side and single frames are extracted. For each frame, the faces are detected and cropped. The mouth and eyes are classified as opened or closed based on a simple check of ratios between nearby facial landmarks of the face detector. Only frames with eyes and mouth classified as open are used, as these potentially show the artifacts described in Sec. 3. The pre-sorted samples are examined manually to delete obvious detection failures and overly blurry scene parts. Due to differences in face detection and classification of the eye and mouth regions, there is no strict one-to-one relation between frames of the fake and real class. The data is split into different scenes to avoid similar frames in the train and test set. The number of frames per scene in each class varies. The training data contains frames of 16 different scenes, the test data of four. Example images of the four scenes contained in the test set are shown in Fig. 8. The dataset contains a total of 5330 samples. The number of samples per class for each set are shown in Tab. 1. Additionally, we evaluate on the dataset proposed by Afchar *et al*. [1], which was created in a similar way but without using side-by-side comparisons and pre-classified mouth and eye regions.

[3] https://github.com/tkarras/progressive_growing_of_gans

[4] https://github.com/openai/glow

[5] https://www.youtube.com/Life2Coding

[6] https://www.youtube.com/derpfakes

Table 1. Number of samples in the proposed Deepfake dataset.

Set	Fake	Real
Train	2070	2440
Test	375	445

Figure 8. Scenes of the proposed Deepfake test data.

Face2Face. For evaluation on Face2Face manipulations, we extract samples from the publicly available FaceForensics dataset [33]. We limit our evaluations to the so called "self-reenactment" examples. For this case, Face2Face is applied to re-render the input videos without further manipulating the facial expressions. The dataset already provides the video frames cropped to the face region. We randomly select 10000 frames of the training data for each the original and altered class and 5000 samples each class of the testing data.

4.2. Detection Pipelines

We propose a set of straightforward features for detecting generated faces, Deepfakes, and Face2Face images.

Generated Faces. Differences in eye color are used to detect generated faces. To extract color features of each eye, we use commonly available computer vision methods. Facial landmarks are detected for each input image. After detection, the images are cropped to the face region and resized to 768 pixels in height, such that all samples are processed at a fixed resolution. Iris pixels need to be detected to compute eye color features. The iris usually has a high contrast to the sclera. For segmentation, we try to detect the iris as a circle inside of the landmarks for the eye. To that end, Canny edge detection and a Hough Circle Transformation are applied. The segmentation is further refined by thresholding on dark pixels that likely belong to the pupil and on bright pixels that likely stem from reflections or an inaccurate segmentation. An example segmentation result

Figure 9. Example result of the iris segmentation.

with the main steps of the pipeline is shown in Fig. 9. Two consistency checks help to identify failure cases in the iris detection: the distance of the center of the iris and the center of the eye (according to the landmarks) should be similar for the left and right eye. Additionally, both irises are expected to have similar radii. To improve the confidence in segmentation, samples violating these assumptions are discarded. This also helps to sort out implausible samples and failure cases of the generated faces.

We define multiple features to characterize the dissimilarity in color of the left and right eye. First, the colors are transformed into HSV color space and averaged for the segmented pixels of the left and right eye. The differences between the averaged H,S,V values of the left eye l_H, l_S, l_V and right eye r_H, r_S, r_V are computed as

$$\begin{aligned} \text{Dist}_H &= \min(|l_H - r_H|, 360 - |l_H - r_H|) \\ \text{Dist}_S &= |l_S - r_S| \\ \text{Dist}_V &= |l_V - r_V| \\ \text{Dist}_{HSV} &= \text{Dist}_H + \text{Dist}_S + \text{Dist}_V \quad . \end{aligned} \tag{1}$$

Additionally, normalized 64-bin histograms of the RGB values are computed for each iris. For each color channel, the correlation between the left and right eye is calculated individually, resulting in the features Correl_R, Correl_G, Correl_B. For classification without the need of any training data, the HSV distance Dist_{HSV} can be used directly. To further improve classification performance, we combine the described features into a six-dimensional feature vector

$$F = (\text{Dist}_H, \text{Dist}_S, \text{Dist}_V, \text{Correl}_R, \text{Correl}_G, \text{Correl}_B) \quad . \tag{2}$$

These features are passed to a bagged version of a k-nearest-neighbor classifier with $k = 20$ using Euclidean distances.

Deepfakes. We exploit missing reflections, and missing details in the eye and teeth areas for Deepfake detection. We again detect facial landmarks and crop the input image to the facial region. To accomodate varying resolutions of the input data, all data is resized to 256 pixels in height. The eye region is segmented by considering the pixels in the convex hull of the associated landmarks. To segment the teeth, the image is converted to grayscale. The pixels contained in the convex hull of the mouth landmarks are clustered into a bright and dark cluster by K-Means clustering. All pixels in the bright cluster are considered as belonging to teeth. The sample is discarded if less than 1% of mouth pixels is classified as teeth. Examples of the processed crops and resulting

Figure 10. Example of face crops and segmentations as used in the processing pipeline for the Deepfake data.

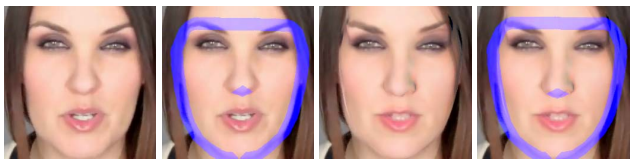
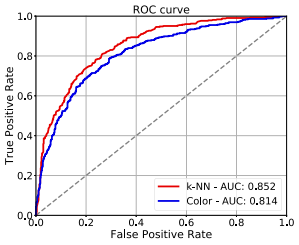


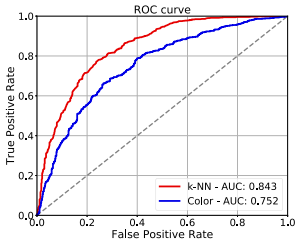
Figure 11. Example of face crops and segmentations as used in the processing pipeline for the Face2Face data.

segmentation are shown in Fig. 10. We choose the texture energy approach [23] by Laws to generate features that describe the complexity of the texture. The texture features are computed by 16 fixed 5×5 convolution masks designed to extract specific texture characteristics. As proposed by Laws [23], the local average of each pixel is subtracted with a kernel size of 15×15 before calculating the energy maps. The 16 energy maps resulting from filtering with the fixed kernels are averaged with a 10×10 kernel, and symmetric pairs are combined. This results in nine texture features per pixel. To generate feature vectors for each sample, we average the nine features for all pixels within the eyes, teeth and the whole image, respectively. These descriptors are classified with two different models. We fit a logistic regression model as an exemplary "off-the-shelf" classifier. As a higher capacity model we train a small neural network with low requirements regarding training time and hyperparameter tuning. The neural network is fully connected, contains three layers with 64 nodes and ReLU activation functions. It is trained with an ADAM solver and regularized by a L_2 penalty with an alpha value of 0.1. We train the classifiers on just the eye feature vector, just the teeth feature vector, a 16-dimensional feature vector containing the eye and teeth features, and on the feature vector extracted from the full face crop.

Face2Face. For the Face2Face data we leverage the same classifiers as described for Deepfakes, but different features. Instead of segmenting eyes and teeth we calculate the features for the face border and nose tip. Again, the segmentation is simply done by using detected facial landmarks. The face border is extracted by generating the convex hull around all landmarks and taking a ten pixel wide area around the edge. The nose is segmented by taking the convex hull around associated landmarks. An example of the segmentation is shown in Fig. 11. The computations of the features, the classifiers and their hyper-parameters



(a) Faces generated by ProGAN



(b) Faces generated by Glow

Figure 12. Classification of test data with high confidence in iris segmentation.

Table 2. ROC curve AUC values for the classification of the test data as proposed in Sec. 4.1.

Data	AUC (k-NN)	AUC (Color)	P	N
ProGAN	**0.852**	0.814	424	580
Glow	0.843	0.752	716	580
ProGAN	0.802	0.764	1000	1000
Glow	0.796	0.704	1000	1000

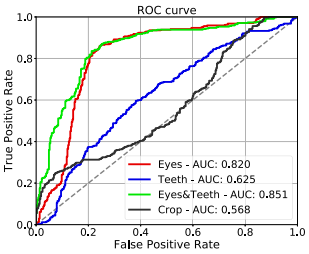
are as described before. We train the classifiers on just the nose feature vector, just the face border feature vector, a 16-dimensional feature vector containing both, and on the feature vector extracted from the full face crop.

4.3. Classification Results

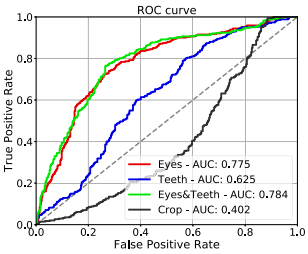
The proposed features are evaluated on the test data for all three types of manipulations.

Generated Faces. We generate two receiver operating characteristic (ROC) curves. Once directly for the difference in color Dist_{HSV}, and once for the probability returned by the trained k-nearest-neighbor classifier. We refer to these classifiers as "Color" and "k-NN", respectively. Figure 12 shows the ROC curves for the classification of the test data. Only images with high confidence in iris segmentation were evaluated for the curves in Fig. 12. The area under the curve (AUC) values and number of evaluated positive and negative samples are shown in Tab. 2. The best result, with an AUC of 0.852, is achieved by the k-NN classifier for the ProGAN test data with high confidence in segmentation. Classification using the difference in color Dist_{HSV} directly without the use of any training data at all leads to an AUC of up to 0.814. Row three and four of Tab. 2 display the AUC values for classifying the test sets without discarding any samples. The performance only worsens slightly, indicating a high robustness to segmentation faults. Overall performance varies between 0.764-0.852 for the ProGAN data and 0.704-0.843 for the Glow data.

Deepfakes. 342 fake and 367 real Deepfake samples are processed after discarding some samples for missing segmentations. Figure 13 shows the resulting ROC curves of the proposed classifiers. All AUC values are shown in Tab. 3. We refer to the neural network classifier as MLP and the logistic regression model as LogReg. The results show that the classifiers are able to distinguish a large amount of samples. Classification only on the features generated from teeth performs relatively poorly, with an AUC of 0.625 for both classifiers. The features extracted from the eye region lead to much better performances of 0.820 and 0.784. The best result with an AUC of 0.851 is achieved by the neural network using the combined feature vector. The results for using the features for the whole crop are close to guessing or even slightly negative for the logistic regression. This is a good indication that the eye and teeth features are in fact meaningful and the results are not caused by differences between the images of the real and fake class.



(a) MLP



(b) Logistic Regression

Figure 13. Classification of the proposed Deepfake test data with different features and classifiers.

Table 3. ROC curve AUC values for the classification of the Deepfake test data as proposed in Sec. 4.1.

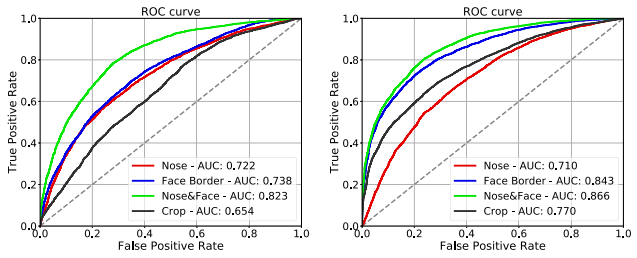
Classifier	Eyes	Teeth	Eyes&Teeth	Crop
MLP - AUC	0.820	0.625	**0.851**	0.568
LogReg - AUC	0.775	0.625	0.784	0.402

We also evaluate the classifiers on the dataset by Afchar *et al*. [1]. Re-training is necessary, as there might be an overlap between the training and testing scenes used in the datasets. 588 fakes and 910 unaltered test samples remain after discarding samples where neither mouth nor eyes are classified as opened or segmentation is missing. The AUC values for classifying the test data are shown in Tab. 4. The results for this data are comparable to the proposed dataset. However, the specific segmentation seems to be less important. Even classifying the texture features for the whole crops leads to an AUC of up to 0.815.

Face2Face. The FaceForensics dataset [33] is used to further evaluate the applicability of the proposed texture features. The resulting ROC curves of the classifiers are shown in Fig. 14. All AUC values are shown in Tab. 5. The classifier based on the logistic regression model performs best,

Table 4. ROC curve AUC values for the classification of the test data proposed in [1].

Classifier	Eyes	Teeth	Eyes&Teeth	Crop
MLP - AUC	**0.838**	0.727	0.830	0.815
LogReg - AUC	0.832	0.727	0.779	0.692



(a) MLP (b) Logistic Regression

Figure 14. Classification of the FaceForensics test data with different features and classifiers.

Table 5. ROC curve AUC values for the classification of FaceForensics test data.

Classifier	Nose	Face Border	Both	Crop
MLP - AUC	0.722	0.738	0.823	0.654
LogReg - AUC	0.710	0.843	**0.866**	0.770

with values of up to 0.866 for the combined features. For the Face2Face samples, the less specific features extracted from the whole face crop lead to an AUC of 0.770. This indicates that the difference in texture can be observed on the whole face, and is not as depended on the segmentation of particular hot spots as the Deepfake data.

5. Conclusion

The proposed algorithms follow a simple recipe. Images emerging from new manipulation techniques are searched for visual artifacts arising from known to be hard problems to solve, like geometry or illumination estimation. Based on this insight, we can build simple handcrafted features to characterize such artifacts which can subsequently be classified by well established methods. Since the features describe specific artifacts, classification models, such as logistic regression or small neural networks with a small parameter space can suffice for the task. This can be a great advantage compared to methods leveraging large deep-learning models, as there are much lower requirements for training data and time. Subsequently, this enables fast prototyping and a certain agility in reacting to new or quickly changing manipulation methods such as Deepfakes. Since the methods rely on scene content and work with fixed resolutions they are robust to varying compression and input sizes, which we find in data as described in Sec. 4. An additional advantage of detecting and classifying visual artifacts is the easy interpretation and comprehensibility of such cues, which helps in communicating results but also in better understanding the added value of other methods by establishing a visual baseline.

The presented classification results are somewhat worse than results reported by Afchar *et al*. [1] or Rössler *et al*. [33] that leverage complex CNN models and large training datasets. Nevertheless, the proposed pipelines achieve surprisingly good results, despite their simplicity, with AUC values of up to 0.866. For the classification of fully generated faces as discussed in Sec. 4.1, the proposed method even achieves an AUC performance of up to 0.814 without the need for any training data at all.

It is important to note that the presented methods are only applicable to images meeting certain prerequisites (e.g. open eyes, visible teeth). Additionally, the results are dependent on the specific test data. For example, mismatches in the eye color can probably be corrected by a manipulator in post-processing. Data such as the Deepfakes collected "in-the-wild" have a high amount of ambiguity, as the underlying methods and models are uncertain. This is shown in the comparison between the proposed data and the data provided by Afchar *et al*. [1]. The Deepfakes data collected for this work is best classified by the highly specific eye and mouth region, whereas our classifier can distinguish the DeepFakes provided by Afchar *et al*. with similar results when the whole region around the face is used.

Generally, we consider the proposed approach as a bridge technology helping to understand to what extend current face manipulations can be exposed visually. With further advances in manipulation methods, as for example shown by Kim *et al*. [20], we expect visual cues to become weaker indicators of manipulations. To prepare for this development, there is clearly a need in further developing statistical methods to take over the detection of visually even more consistent manipulations.

Acknowledgments

This material is based on research sponsored by the Air Force Research Laboratory and the Defense Advanced Research Projects Agency under agreement number FA8750-16-2-0204. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and the Defense Advanced Research Projects Agency or the U.S. Government.

References

[1] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. MesoNet: a Compact Facial Video Forgery Detection Network. In *2018 IEEE Workshop on Information Forensics and Security (WIFS)*, 2018.

[2] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh. Recycle-GAN: Unsupervised Video Retargeting. In *ECCV*, 2018.

[3] M. Barni, E. Nowroozi, and B. Tondi. Detection of Adaptive Histogram Equalization Robust Against JPEG Compression. In *Biometrics and Forensics, 2018 IEEE International Workshop on (IWBF)*, pages 1–8, 2018.

[4] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[5] V. Conotter, E. Bodnari, G. Boato, and H. Farid. Physiologically-based detection of computer generated faces in video. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 248–252. IEEE, 2014.

[6] D. Cozzolino, G. Poggi, and L. Verdoliva. Splicebuster: A new blind image splicing detector. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Nov. 2015.

[7] D. Cozzolino and L. Verdoliva. Noiseprint: a CNN-based camera model fingerprint. arxiv preprint, University of Naples, 2018. arXiv:1808.08396.

[8] L. Dinh, D. Krueger, and Y. Bengio. NICE: Nonlinear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[9] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803*, 2016.

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

[11] J. Fridrich and J. Kodovský. Rich Models for Steganalysis of Digital Images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2012.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[13] D. Güera and E. J. Delp. Deepfake Video Detection Using Recurrent Neural Networks. In *IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2018.

[14] M. Huh, A. Liu, A. Owens, and A. A. Efros. Fighting Fake News: Image Splice Detection via Learned Self-Consistency. *European Conference on Computer Vision (ECCV)*, 2018.

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *CVPR*, 2017.

[16] J. F. Boylan, The New York Times. Will DeepFake Technology Destroy Democracy? `https://www.nytimes.com/2018/10/17/opinion/deep-fake-technology-democracy.html`, Oct. 2018.

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

[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations*, 2018.

[19] E. Kee and H. Farid. Exposing digital forgeries from 3-D lighting environments. In *2010 IEEE International Workshop on Information Forensics and Security*. Institute of Electrical and Electronics Engineers (IEEE), Dec. 2010.

[20] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Nießner, P. Pérez, C. Richardt, M. Zollöfer, and C. Theobalt. Deep Video Portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018.

[21] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

[22] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] K. I. Laws. Textured image segmentation. Technical report, University of Southern California Los Angeles Image Processing INST, 1980.

[24] H. Li, B. Li, S. Tan, and J. Huang. Detection of deep network generated images using disparities in color components. *arXiv preprint arXiv:1808.07276*, 2018.

[25] Y. Li, M.-C. Chang, and S. Lyu. In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking. *ArXiv e-prints*, June 2018.

[26] Y. Li and S. Lyu. Exposing DeepFake Videos By Detecting Face Warping Artifacts. *arXiv preprint arXiv:1811.00656*, 2018.

[27] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, Dec. 2015.

[28] S. Mandelli, N. Bonettini, P. Bestagini, V. Lipari, and S. Tubaro. Multiple JPEG Compression Detection Through Task-Driven Non-Negative Matrix Factorization. In *Acoustics, Speech and Signal Processing, 2018 IEEE International Conference on (ICASSP)*, pages 2106–2110, 2018.

[29] J. F. O'Brien and H. Farid. Exposing Photo Manipulation with Inconsistent Reflections. *ACM Trans. Graph.*, 31(1):4:1–4:11, Feb. 2012.

[30] A. Pumarola, A. Agudo, A. Martinez, A. Sanfeliu, and F. Moreno-Noguer. GANimation: Anatomically-aware Facial Animation from a Single Image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[31] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen. Distinguishing computer graphics from natural images using convolution neural networks. In *Information Forensics and Security (WIFS), 2017 IEEE Workshop on*, pages 1–6. IEEE, 2017.

[32] J. Redi, W. Taktak, and J.-L. Dugelay. Digital Image Forensics: A Booklet for Beginners. *Multimedia Tools and Applications*, 51(1):133–162, Jan. 2011.

[33] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces. *arXiv*, 2018.

[34] O. Stelzer. Iris heterochromia: variations in form, age changes, sex dimorphism. *Anthropologischer Anzeiger; Bericht uber die biologisch-anthropologische Literatur*, 37(2):107–116, 1979.

[35] S. Tariq, S. Lee, H. Kim, Y. Shin, and S. S. Woo. Detecting Both Machine and Human Created Fake Face Images In the Wild. In *Proceedings of the 2Nd International Workshop on Multimedia Privacy and Security*, MPS '18, pages 81–87, New York, NY, USA, 2018. ACM.

[36] The Guardian. You thought fake news was bad? Deep fakes are where truth goes to die. `https://www.theguardian.com/technology/2018/nov/12/deep-fakes-fake-news-truth`, Nov. 2018.

[37] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

[38] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro. Video-to-Video Synthesis. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[39] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[40] Y. Zhang, L. Zheng, and V. L. L. Thing. Automated Face Swapping and Its Detection. In *Signal and Image Processing, IEEE 2nd International Conference on (ICSIP)*, pages 16–19. IEEE, 2017.

[41] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Two-Stream Neural Networks for Tampered Face Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1831–1839, July 2017.

[42] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt. State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications. In *Computer Graphics Forum*, volume 37, pages 523–550. Wiley Online Library, 2018.