FaceForensics++: Learning to Detect Manipulated Facial Images

Andreas Rössler[1] Davide Cozzolino[2] Luisa Verdoliva[2] Christian Riess[3]
Justus Thies[1] Matthias Nießner[1]
[1]Technical University of Munich [2]University Federico II of Naples [3]University of Erlangen-Nuremberg

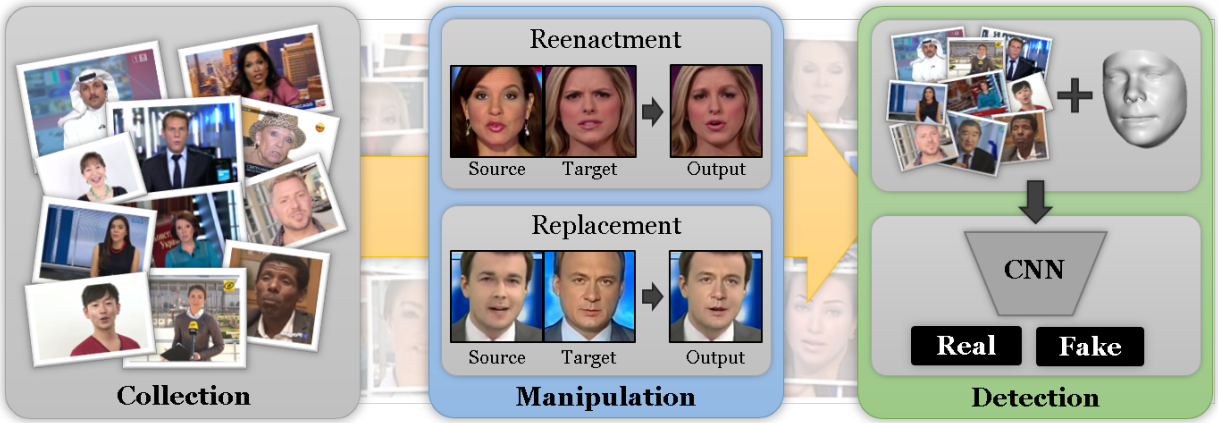


Figure 1: *FaceForensics++* is a dataset of facial forgeries that enables researchers to train deep-learning-based approaches in a supervised fashion. The dataset contains manipulations created with four state-of-the-art methods, namely, *Face2Face*, *FaceSwap*, *DeepFakes*, and *NeuralTextures*.

Abstract

The rapid progress in synthetic image generation and manipulation has now come to a point where it raises significant concerns for the implications towards society. At best, this leads to a loss of trust in digital content, but could potentially cause further harm by spreading false information or fake news. This paper examines the realism of state-of-the-art image manipulations, and how difficult it is to detect them, either automatically or by humans.

To standardize the evaluation of detection methods, we propose an automated benchmark for facial manipulation detection[1]*. In particular, the benchmark is based on DeepFakes [1], Face2Face [59], FaceSwap [2] and NeuralTextures [57] as prominent representatives for facial manipulations at random compression level and size. The benchmark is publicly available*[2] *and contains a hidden test set as well as a database of over* 1.8 *million manipulated images. This dataset is over an order of magnitude larger than comparable, publicly available, forgery datasets. Based on this data, we performed a thorough analysis of data-driven forgery detectors. We show that the use of additional domain-specific knowledge improves forgery detection to unprecedented accuracy, even in the presence of strong compression, and clearly outperforms human observers.*

1. Introduction

Manipulation of visual content has now become ubiquitous, and one of the most critical topics in our digital society. For instance, *DeepFakes* [1] has shown how computer graphics and visualization techniques can be used to defame persons by replacing their face by the face of a different person. Faces are of special interest to current manipulation methods for various reasons: firstly, the reconstruction and tracking of human faces is a well-examined field in computer vision [68], which is the foundation of these editing approaches. Secondly, faces play a central role in human communication, as the face of a person can emphasize a message or it can even convey a message in its own right [28].

Current facial manipulation methods can be separated into two categories: facial expression manipulation and facial identity manipulation (see Fig. 2). One of the most prominent facial expression manipulation techniques is the method of Thies et al. [59] called *Face2Face*. It enables the transfer of facial expressions of one person to another person in real time using only commodity hardware. Follow-up work such as "Synthesizing Obama" [55] is able to animate the face of a person based on an audio input sequence.

1. kaldir.vc.in.tum.de/faceforensics_benchmark
2. github.com/ondyari/FaceForensics

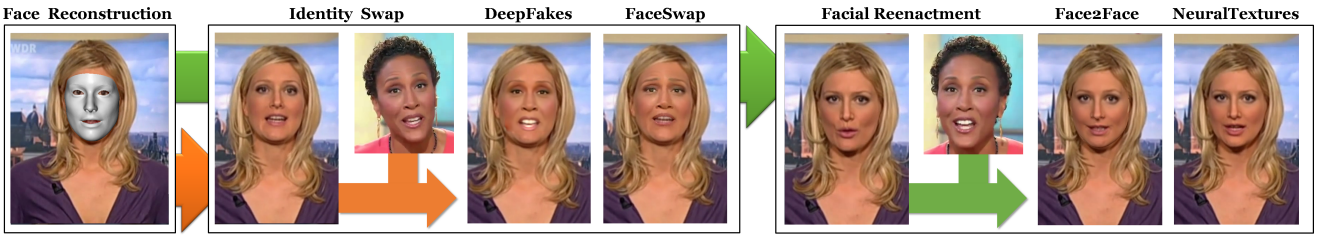


Figure 2: Advances in the digitization of human faces have become the basis for modern facial image editing tools. The editing tools can be split in two main categories: identity modification and expression modification. Aside from manually editing the face using tools such as Photoshop, many automatic approaches have been proposed in the last few years. The most prominent and widespread identity editing technique is face swapping, which has gained significant popularity as lightweight systems are now capable of running on mobile phones. Additionally, facial reenactment techniques are now available, which alter the expressions of a person by transferring the expressions of a source person to the target.

Identity manipulation is the second category of facial forgeries. Instead of changing expressions, these methods replace the face of a person with the face of another person. This category is known as face swapping. It became popular with wide-spread consumer-level applications like Snapchat. *DeepFakes* also performs face swapping, but via deep learning. While face swapping based on simple computer graphics techniques can run in real time, *DeepFakes* need to be trained for each pair of videos, which is a time-consuming task.

In this work, we show that we can automatically and reliably detect such manipulations, and thereby outperform human observers by a significant margin. We leverage recent advances in deep learning, in particular, the ability to learn extremely powerful image features with convolutional neural networks (CNNs). We tackle the detection problem by training a neural network in a supervised fashion. To this end, we generate a large-scale dataset of manipulations based on the classical computer graphics-based methods *Face2Face* [59] and *FaceSwap* [2] as well as the learning-based approaches *DeepFakes* [1] and *NeuralTextures* [57].

As the digital media forensics field lacks a benchmark for forgery detection, we propose an automated benchmark that considers the four manipulation methods in a realistic scenario, i.e., with random compression and random dimensions. Using this benchmark, we evaluate the current state-of-the-art detection methods as well as our forgery detection pipeline that considers the restricted field of facial manipulation methods.

Our paper makes the following contributions:

- an automated benchmark for facial manipulation detection under random compression for a standardized comparison, including a human baseline,
- a novel large-scale dataset of manipulated facial imagery composed of more than 1.8 million images from 1,000 videos with pristine (i.e., real) sources and target ground truth to enable supervised learning,
- an extensive evaluation of state-of-the-art hand-crafted and learned forgery detectors in various scenarios,
- a state-of-the-art forgery detection method tailored to facial manipulations.

2. Related Work

The paper intersects several fields in computer vision and digital multimedia forensics. We cover the most important related papers in the following paragraphs.

Face Manipulation Methods: In the last two decades, interest in virtual face manipulation has rapidly increased. A comprehensive state-of-the-art report has been published by Zollhöfer *et al.* [68]. In particular, Bregler *et al.* [13] presented an image-based approach called Video Rewrite to automatically create a new video of a person with generated mouth movements. With Video Face Replacement [20], Dale *et al.* presented one of the first automatic face swap methods. Using single-camera videos, they reconstruct a 3D model of both faces and exploit the corresponding 3D geometry to warp the source face to the target face. Garrido *et al.* [29] presented a similar system that replaces the face of an actor while preserving the original expressions. VDub [30] uses high-quality 3D face capturing techniques to photo-realistically alter the face of an actor to match the mouth movements of a dubber. Thies *et al.* [58] demonstrated the first real-time expression transfer for facial reenactment. Based on a consumer level RGB-D camera, they reconstruct and track a 3D model of the source and the target actor. The tracked deformations of the source face are applied to the target face model. As a final step, they blend the altered face on top of the original target video. Face2Face, proposed by Thies *et al.* [59], is an advanced

real-time facial reenactment system, capable of altering facial movements in commodity video streams, e.g., videos from the internet. They combine 3D model reconstruction and image-based rendering techniques to generate their output. The same principle can be also applied in Virtual Reality in combination with eye-tracking and reenactment [60] or be extended to the full body [61]. Kim *et al.* [39] learn an image-to-image translation network to convert computer graphic renderings of faces to real images. Instead of a pure image-to-image translation network, NeuralTextures [57] optimizes a neural texture in conjunction with a rendering network to compute the reenactment result. In comparison to Deep Video Portraits [39], it shows sharper results, especially, in the mouth region. Suwajanakorn *et al.* [55] learned the mapping between audio and lip motions, while their compositing approach builds on similar techniques to Face2Face [59]. Averbuch-Elor *et al.* [8] present a reenactment method, Bringing Portraits to Life, which employs 2D warps to deform the image to match the expressions of a source actor. They also compare to the Face2Face technique and achieve similar quality.

Recently, several face image synthesis approaches using deep learning techniques have been proposed. Lu *et al.* [47] provide an overview. Generative adversarial networks (GANs) are used to apply Face Aging [7], to generate new viewpoints [34], or to alter face attributes like skin color [46]. Deep Feature Interpolation [62] shows impressive results on altering face attributes like age, mustache, smiling etc. Similar results of attribute interpolations are achieved by Fader Networks [43]. Most of these deep learning based image synthesis techniques suffer from low image resolutions. Recently, Karras *et al.* [37] have improved the image quality using progressive growing of GANs, producing high-quality synthesis of faces.

Multimedia Forensics: Multimedia forensics aims to ensure authenticity, origin, and provenance of an image or video without the help of an embedded security scheme. Focusing on integrity, early methods are driven by handcrafted features that capture expected statistical or physics-based artifacts that occur during image formation. Surveys on these methods can be found in [26, 53]. More recent literature concentrates on CNN-based solutions, through both supervised and unsupervised learning [10, 17, 12, 9, 35, 67]. For videos, the main body of work focuses on detecting manipulations that can be created with relatively low effort, such as dropped or duplicated frames [63, 31, 45], varying interpolation types [25], copy-move manipulations [11, 21], or chroma-key compositions [48].

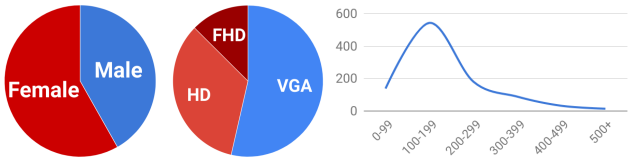
Several other works explicitly refer to detecting manipulations related to faces, such as distinguishing computer generated faces from natural ones [22, 15, 51], morphed faces [50], face splicing [24, 23], face swapping [66, 38] and DeepFakes [5, 44, 33]. For face manipulation detection, some approaches exploit specific artifacts arising from the synthesis process, such as eye blinking [44], or color, texture and shape cues [24, 23]. Other works are more general and propose a deep network trained to capture the subtle inconsistencies arising from low-level and/or high level features [50, 66, 38, 5, 33]. These approaches show impressive results, however robustness issues often remain unaddressed, although they are of paramount importance for practical applications. For example, operations like compression and resizing are known for laundering manipulation traces from the data. In real-world scenarios, these basic operations are standard when images and videos are for example uploaded to social media, which is one of the most important application field for forensic analysis. To this end, our dataset is designed to cover such realistic scenarios, i.e., videos from the wild, manipulated and compressed with different quality levels (see Section 3). The availability of such a large and varied dataset can help researchers to benchmark their approaches and develop better forgery detectors for facial imagery.

Forensic Analysis Datasets: Classical forensics datasets have been created with significant manual effort under very controlled conditions, to isolate specific properties of the data like camera artifacts. While several datasets were proposed that include image manipulations, only a few of them also address the important case of video footage. MICC_F2000, for example, is an image copy-move manipulation dataset consisting of a collection of 700 forged images from various sources [6]. The First IEEE Image Forensics Challenge Dataset comprises a total of 1176 forged images; the Wild Web Dataset [64] with 90 real cases of manipulations coming from the web and the Realistic Tampering dataset [42] including 220 forged images. A database of 2010 FaceSwap- and SwapMe-generated images has been proposed by Zhou *et al.* [66]. Recently, Korshunov and Marcel [41] constructed a dataset of 620 Deepfakes videos created from multiple videos for each of 43 subjects. The National Institute of Standards and Technology (NIST) released the most extensive dataset for generic image manipulation comprising about $50,000$ forged images (both local and global manipulations) and around 500 forged videos [32].

In contrast, we construct a database containing more than 1.8 million images from 4000 fake videos – an order of magnitude more than existing datasets. We evaluate the importance of such a large training corpus in Section 4.

3. Large-Scale Facial Forgery Database

A core contribution of this paper is our *FaceForensics++* dataset extending the preliminary FaceForensics dataset



(a) Gender (b) Resolution (c) Pixel Coverage of Faces

Figure 3: Statistics of our sequences. *VGA* denotes 480p, *HD* denotes 720p, and *FHD* denotes 1080p resolution of our videos. The graph (c) shows the number of sequences (y-axis) with given bounding box pixel height (x-axis).

[52]. This new large-scale dataset enables us to train a state-of-the-art forgery detector for facial image manipulation in a supervised fashion (see Section 4). To this end, we leverage four automated state-of-the-art face manipulation methods, which are applied to 1,000 pristine videos downloaded from the Internet (see Fig. 3 for statistics). To imitate realistic scenarios, we chose to collect videos in the wild, specifically from YouTube. However, early experiments with all manipulation methods showed that the target face had to be nearly front-facing to prevent the manipulation methods from failing or producing strong artifacts. Thus, we perform a manual screening of the resulting clips to ensure a high-quality video selection and to avoid videos with face occlusions. We selected 1,000 video sequences containing $509,914$ images which we use as our pristine data.

To generate a large scale manipulation database, we adapted state-of-the-art video editing methods to work fully automatically. In the following paragraphs, we briefly describe these methods.

For our dataset, we chose two computer graphics-based approaches (*Face2Face* and *FaceSwap*) and two learning-based approaches (*DeepFakes* and *NeuralTextures*). All four methods require source and target actor video pairs as input. The final output of each method is a video composed of generated images. Besides the manipulation output, we also compute ground truth masks that indicate whether a pixel has been modified or not, which can be used to train forgery localization methods. For more information and hyper-parameters we refer to Appendix D.

FaceSwap *FaceSwap* is a graphics-based approach to transfer the face region from a source video to a target video. Based on sparse detected facial landmarks the face region is extracted. Using these landmarks, the method fits a 3D template model using blendshapes. This model is back-projected to the target image by minimizing the difference between the projected shape and the localized landmarks using the textures of the input image. Finally, the rendered model is blended with the image and color correction is applied. We perform these steps for all pairs of source and target frames until one video ends. The implementation is computationally lightweight and can be efficiently run on the CPU.

DeepFakes The term *Deepfakes* has widely become a synonym for face replacement based on deep learning, but it is also the name of a specific manipulation method that was spread via online forums. To distinguish these, we denote said method by *DeepFakes* in the following paper.

There are various public implementations of *DeepFakes* available, most notably *FakeApp* [3] and the *faceswap github* [1]. A face in a target sequence is replaced by a face that has been observed in a source video or image collection. The method is based on two autoencoders with a shared encoder that are trained to reconstruct training images of the source and the target face, respectively. A face detector is used to crop and to align the images. To create a fake image, the trained encoder and decoder of the source face are applied to the target face. The autoencoder output is then blended with the rest of the image using Poisson image editing [49].

For our dataset, we use the *faceswap github* implementation. We slightly modify the implementation by replacing the manual training data selection with a fully automated data loader. We used the default parameters to train the video-pair models. Since the training of these models is very time-consuming, we also publish the models as part of the dataset. This facilitates generation of additional manipulations of these persons with different post-processing.

Face2Face *Face2Face* [59] is a facial reenactment system that transfers the expressions of a source video to a target video while maintaining the identity of the target person. The original implementation is based on two video input streams, with manual keyframe selection. These frames are used to generate a dense reconstruction of the face which can be used to re-synthesize the face under different illumination and expressions. To process our video database, we adapt the Face2Face approach to fully-automatically create reenactment manipulations. We process each video in a preprocessing pass; here, we use the first frames in order to obtain a temporary face identity (i.e., a 3D model), and track the expressions over the remaining frames. In order to select the keyframes required by the approach, we automatically select the frames with the left- and right-most angle of the face. Based on this identity reconstruction, we track the whole video to compute per frame the expression, rigid pose, and lighting parameters as done in the original implementation of *Face2Face*. We generate the reenactment video outputs by transferring the source expression parameters of each frame (i.e., 76 Blendshape coefficients) to the target video. More details of the reenactment process can be found in the original paper [59].

NeuralTextures Thies et al. [57] show facial reenactment as an example for their *NeuralTextures*-based rendering approach. It uses the original video data to learn a neural texture of the target person, including a rendering network. This is trained with a photometric reconstruction loss in combination with an adversarial loss. In our implementation, we apply a patch-based GAN-loss as used in Pix2Pix [36]. The NeuralTextures approach relies on tracked geometry that is used during train and test times. We use the tracking module of *Face2Face* to generate these information. We only modify the facial expressions corresponding to the mouth region, i.e., the eye region stays unchanged (otherwise the rendering network would need conditional input for the eye movement similar to Deep Video Portraits [39]).

Postprocessing - Video Quality To create a realistic setting for manipulated videos, we generate output videos with different quality levels, similar to the video processing of many social networks. Since raw videos are rarely found on the internet, we compress the videos using the H.264 codec, which is widely used by social networks or video-sharing websites. To generate high quality videos, we use a light compression denoted by *HQ* (constant rate quantization parameter equal to 23) which is visually nearly lossless. Low quality videos (*LQ*) are produced using a quantization of 40.

4. Forgery Detection

We cast the forgery detection as a per-frame binary classification problem of the manipulated videos. The following sections show the results of manual and automatic forgery detection. For all experiments, we split the dataset into a fixed training, validation, and test set, consisting of 720, 140, and 140 videos respectively. All evaluations are reported using videos from the test set. For all graphs, we list the exact numbers in Appendix B.

4.1. Forgery Detection of Human Observers

To evaluate the performance of humans in the task of forgery detection, we conducted a user study with 204 participants consisting mostly of computer science university students. This forms the baseline for the automated forgery detection methods.

Layout of the User Study: After a short introduction to the binary task, users are instructed to classify randomly selected images from our test set. The selected images vary in image quality as well as manipulation method; we used a 50:50 split of pristine and fake images. Since the amount time for inspection of an image may be important, and to mimic scenario where a user only spends a limited amount of time per image as is common on social media, we randomly set a time limit of 2, 4 or 6 seconds after which we hide the image. Afterwards, the users were asked whether the displayed image is 'real' or 'fake'. To ensure that the users spend the available time on inspection, the question is asked after the image has been displayed and not during the observation time. We designed the study to only take a few minutes per participant, showing 60 images per attendee, which results in a collection of 12240 human decisions.

Evaluation: In Fig. 4, we show the results of our study on all quality levels, showing a correlation between video quality and the ability to detect fakes. With a lower video quality, the human performance decreases in average from 68.7% to 58.7%. The graph shows the numbers averaged across all time intervals since the different time constraints did not result in significantly different observations.

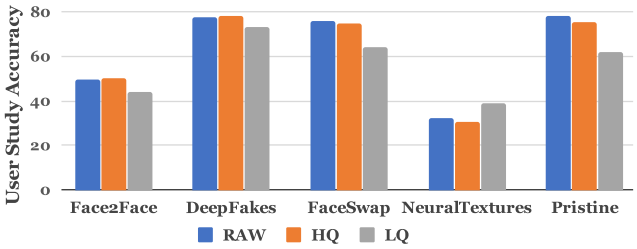


Figure 4: Forgery detection results of our user study with 204 participants. The accuracy is dependent on the video quality and results in a decreasing accuracy rate that is 68.69% in average on raw videos, 66.57% on high quality, and 58.73% on low quality videos.

Note that the user study contained fake images of all four manipulation methods and pristine images. In this setting, Face2Face and NeuralTextures were particularly difficult to detect by human observers, as they do not introduce a strong semantic change, introducing only subtle visual artifacts in contrast to the face replacement methods. NeuralTextures texture seems particularly difficult to detect as human detection accuracy is below random chance and only increases in the challenging low quality task.

4.2. Automatic Forgery Detection Methods

Our forgery detection pipeline is depicted in Fig. 5. Since our goal is to detect forgeries of facial imagery, we use additional domain-specific information that we can extract from input sequences. To this end, we use the state-of-the-art face tracking method by Thies et al. [59] to track the face in the video and to extract the face region of the image. We use a conservative crop (enlarged by a factor of 1.3) around the center of the tracked face, enclosing the reconstructed face. This incorporation of domain knowledge

Figure 5: Our domain-specific forgery detection pipeline for facial manipulations: the input image is processed by a robust face tracking method; we use the information to extract the region of the image covered by the face; this region is fed into a learned classification network that outputs the prediction.

improves the overall performance of a forgery detector in comparison to a naïve approach that uses the whole image as input (see Sec. 4.2.2). We evaluated various variants of our approach by using different state-of-the-art classification methods. We are considering learning-based methods used in the forensic community for generic manipulation detection [10, 17], computer-generated vs natural image detection [51] and face tampering detection [5]. In addition, we show that the classification based on XceptionNet [14] outperforms all other variants in detecting fakes.

4.2.1 Detection based on Steganalysis Features:

We evaluate detection from steganalysis features, following the method by Fridrich et al. [27] which employs hand-crafted features. The features are co-occurrences on 4 pixels patterns along the horizontal and vertical direction on the high-pass images for a total feature length of 162. These features are then used to train a linear Support Vector Machine (SVM) classifier. This technique was the winning approach in the first IEEE Image Forensic Challenge [16]. We provide a 128×128 central crop-out of the face as input to the method. While the hand-crafted method outperforms human accuracy on raw images by a large margin, it struggles to cope with compression, which leads to an accuracy below human performance for low quality videos (see Fig. 6 and Table 1).

4.2.2 Detection based on Learned Features:

For detection from learned features, we evaluate five network architectures known from the literature to solve the classification task:

(1) Cozzolino et al. [17] cast the hand-crafted Steganalysis features from the previous section to a CNN-based network. We fine-tune this network on our large scale dataset.

(2) We use our dataset to train the convolutional neural network proposed by Bayar and Stamm [10] that uses a constrained convolutional layer followed by two convolutional, two max-pooling and three fully-connected layers. The constrained convolutional layer is specifically designed

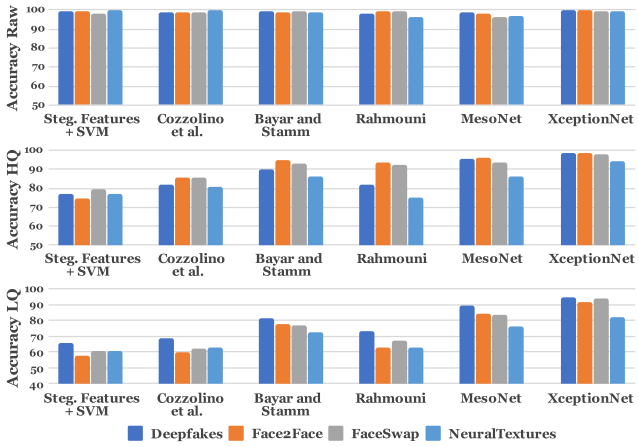


Figure 6: Binary detection accuracy of all evaluated architectures on the different manipulation methods using face tracking when trained on our different manipulation methods separately.

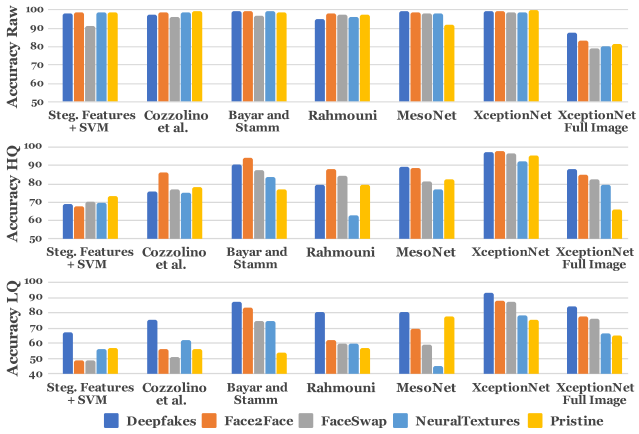


Figure 7: Binary precision values of our baselines when trained on all four manipulation methods simulatenously. See Table 1 for the average accuracy values. Aside from the Full Image XceptionNet, we use the proposed pre-extraction of the face region as input to the approaches.

to suppress the high-level content of the image. Similar to the previous methods, we use a centered 128×128 crop as input.

(3) Rahmouni et al. [51] adopt different CNN architectures with a global pooling layer that computes four statistics (mean, variance, maximum and minimum). We consider the Stats-2L network that had the best performance.

(4) MesoInception-4 [5] is a CNN-based network inspired by InceptionNet [56] to detect face tampering in videos. The network has two inception modules and two classic convolution layers interlaced with max-pooling layers. Afterwards, there are two fully-connected layers. In-

stead of the classic cross-entropy loss, the authors propose the mean squared error between true and predicted labels. We resize the face images to 256×256, the input of the network.

(5) XceptionNet [14] is a traditional CNN trained on ImageNet based on separable convolutions with residual connections. We transfer it to our task by replacing the final fully connected layer with two outputs. The other layers are initialized with the ImageNet weights. To set up the newly inserted fully connected layer, we fix all weights up to the final layers and pre-train the network for 3 epochs. After this step, we train the network for 15 more epochs and choose the best performing model based on validation accuracy.

A detailed description of our training and hyperparameters can be found in Appendix D.

Comparison of our Forgery Detection Variants: Fig. 6 shows the results of a binary forgery detection task using all network architectures evaluated separately on all four manipulation methods and at different video quality levels. All approaches achieve very high performance on raw input data. Performance drops for compressed videos, particularly for hand-crafted features and for shallow CNN architectures [10, 17]. The neural networks are better at handling these situations, with XceptionNet able to achieve compelling results on weak compression while still maintaining reasonable performance on low quality images, as it benefits from its pre-training on ImageNet as well as larger network capacity.

To compare the results of our user study to the performance of our automatic detectors, we also tested the detection variants on a dataset containing images from all manipulation methods. Fig. 7 and Table 1 show the results on the full dataset. Here, our automated detectors outperform human performance by a large margin (cf. Fig. 4). We also evaluate a naïve forgery detector operating on the full image (resized to the XceptionNet input) instead of using face tracking information (see Fig. 7, rightmost column). Due to the lack of domain-specific information, the XceptionNet classifier has a significantly lower accuracy in this scenario. To summarize, domain-specific information in combination with a XceptionNet classifier shows the best performance in each test. We use this network to further understand the influence of the training corpus size and its ability to distinguish between the different manipulation methods.

Forgery Detection of GAN-based methods The experiments show that all detection approaches achieve a lower accuracy on the GAN-based NeuralTextures approach. NeuralTextures is training a unique model for every manipulation which results in a higher variation of possible artifacts. While DeepFakes is also training one model per manipulation, it uses a fixed post-processing pipeline similar to the computer-based manipulation methods and thus has consistent artifacts.

Compression	Raw	HQ	LQ
[14] XceptionNet Full Image	82.01	74.78	70.52
[27] Steg. Features + SVM	97.63	70.97	55.98
[17] Cozzolino *et al.*	98.57	78.45	58.69
[10] Bayar and Stamm	98.74	82.97	66.84
[51] Rahmouni *et al.*	97.03	79.08	61.18
[5] MesoNet	95.23	83.10	70.47
[14] XceptionNet	**99.26**	**95.73**	**81.00**

Table 1: Binary detection accuracy of our baselines when trained on all four manipulation methods. Besides the naïve full image XceptionNet, all methods are trained on a conservative crop (enlarged by a factor of 1.3) around the center of the tracked face.

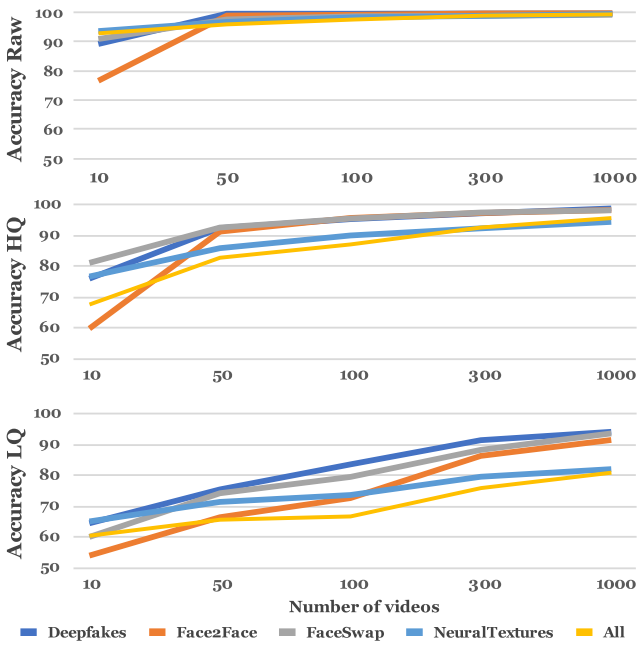


Figure 8: The detection performance of our approach using XceptionNet depends on the training corpus size. Especially, for low quality video data, a large database is needed.

Evaluation of the Training Corpus Size: Fig. 8 shows the importance of the training corpus size. To this end, we trained the XceptionNet classifier with different training corpus sizes on all three video quality level separately. The overall performance increases with the number of training images which is particularly important for low quality video footage, as can be seen in the bottom of the figure.

5. Benchmark

In addition to our large-scale manipulation database, we publish a competitive benchmark for facial forgery detection. To this end, we collected 1000 additional videos and manipulated a subset of those in a similar fashion as in Section 3 for each of our four manipulation methods. As uploaded videos (e.g., to social networks) will be post-processed in various ways, we obscure all selected videos multiple times (e.g., by unknown re-sizing, compression method and bit-rate) to ensure realistic conditions. This processing is directly applied on raw videos. Finally, we manually select a single challenging frame from each video based on visual inspection. Specifically, we collect a set of 1000 images, each image randomly taken from either the manipulation methods or the original footage. Note that we do not necessarily have an equal split of pristine and fake images nor an equal split of the used manipulation methods. The ground truth labels are hidden and are used on our host server to evaluate the classification accuracy of the submitted models. The automated benchmark allows submissions every two weeks from a single submitter to prevent overfitting (similar to existing benchmarks [19]).

As baselines, we evaluate the low quality versions of our previously trained models on the benchmark and report the numbers for each detection method separately (see Table 2). Aside from the Full Image XceptionNet, we use the proposed pre-extraction of the face region as input to the approaches. The relative performance of the classification models is similar to our database test set (see Table 1). However, since the benchmark scenario deviates from the training database, the overall performance of the models is lower, especially for the pristine image detection precision; the major changes being the randomized quality level as well as possible tracking errors during test. Since our proposed method relies on face detections, we predict *fake* as default in case of a tracking failure.

The benchmark is already publicly available to the community and we hope that it leads to a standardized comparison of follow-up work.

Accuracies	DF	F2F	FS	NT	Real	Total
Xcept. Full Image	74.55	75.91	70.87	73.33	51.00	62.40
Steg. Features	73.64	73.72	68.93	63.33	34.00	51.80
Cozzolino *et al.*	85.45	67.88	73.79	78.00	34.40	55.20
Rahmouni *et al.*	85.45	64.23	56.31	60.07	50.00	58.10
Bayar and Stamm	84.55	73.72	82.52	70.67	46.20	61.60
MesoNet	87.27	56.20	61.17	40.67	**72.60**	66.00
XceptionNet	**96.36**	**86.86**	**90.29**	**80.67**	52.40	**70.10**

Table 2: Results of the low quality trained model of each detection method on our benchmark. We report precision results for DeepFakes (DF), Face2Face (F2F), FaceSwap (FS), NeuralTextures (NT), and pristine images (Real) as well as the overall total accuracy.

6. Discussion & Conclusion

While current state-of-the-art facial image manipulation methods exhibit visually stunning results, we demonstrate that they can be detected by trained forgery detectors. It is particularly encouraging that also the challenging case of low-quality video can be tackled by learning-based approaches, where humans and hand-crafted features exhibit difficulties. To train detectors using domain-specific knowledge, we introduce a novel dataset of videos of manipulated faces that exceeds all existing publicly available forensic datasets by an order of magnitude.

In this paper, we focus on the influence of compression to the detectability of state-of-the-art manipulation methods, proposing a standardized benchmark for follow-up work. All image data, trained models, as well as our benchmark are publicly available and are already used by other researchers. In particular, transfer learning is of high interest in the forensic community. As new manipulation methods appear by the day, methods must be developed that are able to detect fakes with little to no training data. Our database is already used for this forensic transfer learning task, where knowledge of one source manipulation domain is transferred to another target domain, as shown by Cozzolino et al [18]. We hope that the dataset and benchmark become a stepping stone for future research in the field of digital media forensics, and in particular with a focus on facial forgeries.

7. Acknowledgement

We gratefully acknowledge the support of this research by the AI Foundation, a TUM-IAS Rudolf Mößbauer Fellowship, the ERC Starting Grant *Scan2CAD* (804724), and a Google Faculty Award. We would also like to thank Google's Chris Bregler for help with the cloud computing. In addition, 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] Deepfakes github. https://github.com/deepfakes/faceswap. Accessed: 2018-10-29. 1, 2, 4, 14

[2] Faceswap. https://github.com/MarekKowalski/FaceSwap/. Accessed: 2018-10-29. 1, 2

[3] Fakeapp. https://www.fakeapp.com/. Accessed: 2018-09-01. 4

[4] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. YouTube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 12

[5] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. *arXiv preprint arXiv:1809.00888*, 2018. 3, 6, 7, 13, 14

[6] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra. A SIFT-based forensic method for copy-move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security*, 6(3):1099–1110, Mar. 2011. 3

[7] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. Face aging with conditional generative adversarial networks. In *IEEE International Conference on Image Processing*, 2017. 3

[8] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F. Cohen. Bringing portraits to life. *ACM Transactions on Graphics (Proceeding of SIGGRAPH Asia 2017)*, 36(4):to appear, 2017. 3

[9] Jawadul H. Bappy, Amit K. Roy-Chowdhury, Jason Bunk, Lakshmanan Nataraj, and B.S. Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *IEEE International Conference on Computer Vision*, pages 4970–4979, 2017. 3

[10] Belhassen Bayar and Matthew C. Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *ACM Workshop on Information Hiding and Multimedia Security*, pages 5–10, 2016. 3, 6, 7, 13, 14

[11] Paolo Bestagini, Simone Milani, Marco Tagliasacchi, and Stefano Tubaro. Local tampering detection in video sequences. In *IEEE International Workshop on Multimedia Signal Processing*, pages 488–493, October 2013. 3

[12] Luca Bondi, Silvia Lameri, David Güera, Paolo Bestagini, Edward J. Delp, and Stefano Tubaro. Tampering Detection and Localization through Clustering of Camera-Based CNN Features. In *IEEE Computer Vision and Pattern Recognition Workshops*, 2017. 3

[13] Christoph Bregler, Michele Covell, and Malcolm Slaney. Video rewrite: Driving visual speech with audio. In *24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 353–360, 1997. 2

[14] Francois Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 6, 7, 13, 14

[15] Valentina Conotter, Ecaterina Bodnari, Giulia Boato, and Hany Farid. Physiologically-based detection of computer generated faces in video. In *IEEE International Conference on Image Processing*, pages 1–5, Oct 2014. 3

[16] Davide Cozzolino, Diego Gragnaniello, and Luisa Verdoliva. Image forgery detection through residual-based local descriptors and block-matching. In *IEEE International Conference on Image Processing*, pages 5297–5301, October 2014. 6

[17] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In *ACM Workshop on Information Hiding and Multimedia Security*, pages 1–6, 2017. 3, 6, 7, 13, 14

[18] 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. 8

[19] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *IEEE Computer Vision and Pattern Recognition*, 2017. 8

[20] Kevin Dale, Kalyan Sunkavalli, Micah K. Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. Video face replacement. *ACM Trans. Graph.*, 30(6):130:1–130:10, Dec. 2011. 2

[21] Luca D'Amiano, Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. A PatchMatch-based Dense-field Algorithm for Video Copy-Move Detection and Localization. *IEEE Transactions on Circuits and Systems for Video Technology, in press*, 2018. 3

[22] Duc-Tien Dang-Nguyen, Giulia Boato, and Francesco De Natale. Identify computer generated characters by analysing facial expressions variation. In *IEEE International Workshop on Information Forensics and Security*, pages 252–257, 2012. 3

[23] Tiago de Carvalho, Fabio A. Faria, Helio Pedrini, Ricardo da S. Torres, and Anderson Rocha. Illuminant-Based Transformed Spaces for Image Forensics. *IEEE Transactions on Information Forensics and Security*, 11(4):720–733, 2016. 3

[24] Tiago de Carvalho, Christian Riess, Elli Angelopoulou, Helio Pedrini, and Anderson Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security*, 8(7):1182–1194, 2013. 3

[25] Xiangling Ding, Gaobo Yang, Ran Li, Lebing Zhang, Yue Li, and Xingming Sun. Identification of Motion-Compensated Frame Rate Up-Conversion Based on Residual Signal. *IEEE Transactions on Circuits and Systems for Video Technology, in press*, 2017. 3

[26] Hany Farid. *Photo Forensics*. The MIT Press, 2016. 3

[27] Jessica Fridrich and Jan Kodovský. Rich Models for Steganalysis of Digital Images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2012. 6, 7, 13

[28] Chris Frith. Role of facial expressions in social interactions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535), Dec. 2009. 1

[29] Pablo Garrido, Levi Valgaerts, Ole Rehmsen, Thorsten Thormaehlen, Patrick Pérez, and Christian Theobalt. Automatic face reenactment. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4217–4224, 2014. 2

[30] Pablo Garrido, Levi Valgaerts, Hamid Sarmadi, Ingmar Steiner, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. *Computer Graphics Forum*, 34(2):193–204, 2015. 2

[31] A. Gironi, Marco Fontani, Tiziano Bianchi, Alessandro Piva, and Mauro Barni. A video forensic technique for detection frame deletion and insertion. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6226–6230, 2014. 3

[32] Haiying Guan, Mark Kozak, Eric Robertson, Yooyoung Lee, Amy N. Yates, Andrew Delgado, Daniel Zhou, Timothee Kheyrkhah, Jeff Smith, and Jonathan Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *IEEE Winter Applications of Computer Vision Workshops*, pages 63–72, Jan 2019. 3

[33] David Güera and Edward J. Delp. Deepfake video detection using recurrent neural networks. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2018. 3

[34] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis. In *IEEE International Conference on Computer Vision*, 2017. 3

[35] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *European Conference on Computer Vision*, 2018. 3

[36] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 5, 14

[37] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations*, 2018. 3

[38] Ali Khodabakhsh, Raghavendra Ramachandra, Kiran Raja, Pankaj Wasnik, and Christoph Busch. Fake face detection methods: Can they be generalized? In *International Conference of the Biometrics Special Interest Group*, 2018. 3

[39] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep Video Portraits. *ACM Transactions on Graphics 2018 (TOG)*, 2018. 3, 5

[40] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 12

[41] Pavel Korshunov and Sebastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018. 3

[42] Pawel Korus and Jiwu Huang. Multi-scale Analysis Strategies in PRNU-based Tampering Localization. *IEEE Transactions on Information Forensics and Security*, 12(4):809–824, Apr. 2017. 3

[43] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, and Marc'Aurelio Ranzato Ludovic Denoyer. Fader networks: Manipulating images by sliding attributes. *CoRR*, abs/1706.00409, 2017. 3

[44] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. In *IEEE WIFS*, 2018. 3

[45] Chengjiang Long, Eric Smith, Arslan Basharat, and Anthony Hoogs. A C3D-based Convolutional Neural Network for Frame Dropping Detection in a Single Video Shot. In *IEEE Computer Vision and Pattern Recognition Workshops*, pages 1898–1906, 2017. 3

[46] Yongyi Lu, Yu-Wing Tai, and Chi-Keung Tang. Conditional cyclegan for attribute guided face image generation. In *European Conference on Computer Vision*, 2018. 3

[47] Zhihe Lu, Zhihang Li, Jie Cao, Ran He, and Zhenan Sun. Recent progress of face image synthesis. In *IAPR Asian Conference on Pattern Recognition*, 2017. 3

[48] Patrick Mullan, Davide Cozzolino, Luisa Verdoliva, and Christian Riess. Residual-based forensic comparison of video sequences. In *IEEE International Conference on Image Processing*, 2017. 3

[49] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on graphics (TOG)*, 22(3):313–318, 2003. 4, 14

[50] Ramachandra Raghavendra, Kiran B. Raja, Sushma Venkatesh, and Christoph Busch. Transferable Deep-CNN features for detecting digital and print-scanned morphed face images. In *IEEE Computer Vision and Pattern Recognition Workshops*, 2017. 3

[51] Nicolas Rahmouni, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Distinguishing computer graphics from natural images using convolution neural networks. In *IEEE Workshop on Information Forensics and Security*, pages 1–6, 2017. 3, 6, 7, 13, 14

[52] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics: A large-scale video dataset for forgery detection in human faces. *arXiv*, 2018. 3

[53] Husrev T. Sencar and Nasir Memon. *Digital Image Forensics — There is More to a Picture than Meets the Eye*. Springer, 2013. 3

[54] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016. 14

[55] Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing Obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4), 2017. 1, 3

[56] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. 2017. 6

[57] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics 2019 (TOG)*, 2019. 1, 2, 3, 4, 14

[58] Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia 2015*, 34(6):Art. No. 183, 2015. 2

[59] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, June 2016. 1, 2, 3, 4, 5, 12

[60] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. FaceVR: Real-Time Gaze-Aware Facial Reenactment in Virtual Reality. *ACM Transactions on Graphics (TOG)*, 2018. 3

[61] Justus Thies, Michael Zollhöfer, Christian Theobalt, Marc Stamminger, and Matthias Nießner. Headon: Real-time reenactment of human portrait videos. *arXiv preprint arXiv:1805.11729*, 2018. 3

[62] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[63] Weihong Wang and Hany Farid. Exposing Digital Forgeries in Interlaced and Deinterlaced Video. *IEEE Transactions on Information Forensics and Security*, 2(3):438–449, 2007. 3

[64] Markos Zampoglou, Symeon Papadopoulos, , and Yiannis Kompatsiaris. Detecting image splicing in the wild (Web). In *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2015. 3

[65] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016. 14

[66] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Two-stream neural networks for tampered face detection. In *IEEE Computer Vision and Pattern Recognition Workshops*, pages 1831–1839, 2017. 3

[67] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection. In *CVPR*, 2018. 3

[68] Michael Zollhöfer, Justus Thies, Darek Bradley, Pablo Garrido, Thabo Beeler, Patrick Péerez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. *Computer Graphics Forum*, 37(2):523–550, 2018. 1, 2

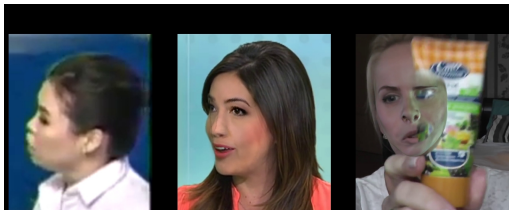


Figure 9: Automatic face editing tools rely on the ability to track the face in the target video. State-of-the-art tracking methods like Thies et al. [59] fail in cases of profile imagery of a face (left). Rotations larger than $45°$ (middle) and occlusions (right) lead to tracking errors.

Methods	Train	Validation	Test
Pristine	366,847	68,511	73,770
DeepFakes	366,835	68,506	73,768
Face2Face	366,843	68,511	73,770
FaceSwap	291,434	54,618	59,640
NeuralTextures	291,834	54,630	59,672

Table 3: Number of images per manipulation method. DeepFakes manipulates every frame of the target sequence, whereas FaceSwap and NeuralTextures only manipulate the minimum number of frames across the source and target video. Face2Face, however, maps all source expressions to the target sequence and rewinds the target video if necessary. Number of manipulated frames can vary due to miss-detection in the respective face tracking modules of our manipulation methods.

Appendix

In *FaceForensics++*, we evaluate the performance of state-of-the-art facial manipulation detection approaches using a large-scale dataset that we generated with four different facial manipulation methods. In addition, we proposed an automated benchmark to compare future detection approaches as well as their robustness against unknown post-processing operations such as compression.

This supplemental document reports details on our pristine data acquisition (Appendix A), ensuring suited input sequences. Appendix B lists the exact numbers of our binary classification experiments presented in the main paper. Besides binary classification, the database is also interesting for evaluating manipulation classification (Appendix C). In Appendix D, we list all chosen hyperparameters of both the manipulation methods as well as the detection techniques.

A. Pristine Data Acquisition

For a realistic scenario, we chose to collect videos in the wild, more specifically from YouTube. Early experiments with all manipulation methods showed that the pristine videos have to fulfill certain criteria. The target face has to be nearly front-facing and without occlusions, to prevent the methods from failing or producing strong artifacts (see Fig. 9). We use the YouTube-8m dataset [4] to collect videos with the tags "face", "newscaster" or "newsprogram" and also included videos which we obtained from the YouTube search interface with the same tags and additional tags like "interview", "blog", or "video blog". To ensure adequate video quality, we only downloaded videos that offer a resolution of 480p or higher. For every video, we save its metadata to sort them by properties later on. In order to match the above requirements, we first process all downloaded videos with the Dlib face detector [40], which is based on Histograms of Oriented Gradients (HOG). During this step, we track the largest detected face by ensuring that the centers of two detections of consecutive frames are pixel-wise close. The histogram-based face tracker was chosen to ensure that the resulting video sequences contain little occlusions and, thus, contain easy-to-manipulate faces.

Except FaceSwap, all methods need a sufficiently large set of image in a target sequence to train on. We select sequences with at least 280 frames. To ensure a high quality video selection and to avoid videos with face occlusions, we perform a manual screening of the clips which resulted in 1,000 video sequences containing $509,914$ images.

All examined manipulation methods need a source and a target video. In case of facial reenactment, the expressions of the source video are transferred to the target video while retaining the identity of the target person. In contrast, face swapping methods replace the face in the target video with the face in the source video. To ensure high quality face swapping, we select video pairs with similar large faces (considering the bounding box sizes detected by DLib), the same gender of the persons and similar video frame rates.

Table 3 lists the final numbers of our dataset for all manipulation methods and the pristine data.

B. Forgery Detection

In this section, we list all numbers from the graphs of the main paper. Table 4 shows the accuracies of the manipulation-specific forgery detectors (i.e., the detectors are trained on the respective manipulation method). In contrast, Table 5 shows the accuracies of the forgery detectors trained on the whole *FaceForensics++* dataset. In Table 6, we show the importance of a large-scale database. The numbers of our user study are listed in Table 7 including the modality which is used to inspect the images.

	Raw				Compressed 23				Compressed 40			
	DF	**F2F**	**FS**	**NT**	**DF**	**F2F**	**FS**	**NT**	**DF**	**F2F**	**FS**	**NT**
Steg. Features + SVM [27]	99.03	99.13	98.27	**99.88**	77.12	74.68	79.51	76.94	65.58	57.55	60.58	60.69
Cozzolino *et al.* [17]	98.83	98.56	98.89	**99.88**	81.78	85.32	85.69	80.60	68.26	59.38	62.08	62.42
Bayar and Stamm [10]	99.28	98.79	98.98	98.78	90.18	94.93	93.14	86.04	80.95	77.30	76.83	72.38
Rahmouni *et al.* [51]	98.03	98.96	98.94	96.06	82.16	93.48	92.51	75.18	73.25	62.33	67.08	62.59
MesoNet [5]	98.41	97.96	96.07	97.05	95.26	95.84	93.43	85.96	89.52	84.44	83.56	75.74
XceptionNet [14]	**99.59**	**99.61**	**99.14**	99.36	**98.85**	**98.36**	**98.23**	**94.5**	**94.28**	**91.56**	**93.7**	**82.11**

Table 4: Accuracy of manipulation-specific forgery detectors. We show the results for raw and the compressed datasets of all four manipulation methods (DF: DeepFakes, F2F: Face2Face, FS: FaceSwap and NT: NeuralTextures).

	Raw					Compressed 23					Compressed 40				
	DF	**F2F**	**FS**	**NT**	**P**	**DF**	**F2F**	**FS**	**NT**	**P**	**DF**	**F2F**	**FS**	**NT**	**P**
Steg. Features + SVM [27]	97.96	98.40	91.35	98.56	98.70	68.80	67.69	70.12	69.21	72.98	67.07	48.55	48.68	55.84	56.94
Cozzolino *et al.* [17]	97.24	98.51	95.93	98.74	99.53	75.51	86.34	76.81	75.34	78.41	75.63	56.01	50.67	62.15	56.27
Bayar and Stamm [10]	99.25	99.04	96.80	**99.11**	98.92	90.25	93.96	87.74	83.69	77.02	86.93	83.66	74.28	74.36	53.87
Rahmouni *et al.* [51]	94.83	98.25	97.59	96.21	97.34	79.66	87.87	84.34	62.65	79.52	80.36	62.04	59.90	59.99	56.79
MesoNet [5]	99.24	98.35	98.15	97.96	92.04	89.55	88.60	81.24	76.62	82.19	80.43	69.06	59.16	44.81	**77.58**
XceptionNet [14]	**99.29**	**99.23**	**98.39**	98.64	**99.64**	**97.49**	**97.69**	**96.79**	**92.19**	**95.41**	**93.36**	**88.09**	**87.42**	**78.06**	75.27
Full Image Xception [14]	87.73	83.22	79.29	79.97	81.46	88.00	84.98	82.23	79.60	65.85	84.06	77.56	76.12	66.03	65.09

Table 5: Detection accuracies when trained on all manipulation methods at once and evaluated on specific manipulation methods or pristine data (DF: DeepFakes, F2F: Face2Face, FS: FaceSwap, NT: NeuralTextures, and P: Pristine). The average accuricies are listed in the main paper.

	Raw					Compressed 23					Compressed 40				
	DF	**F2F**	**FS**	**NT**	**All**	**DF**	**F2F**	**FS**	**NT**	**All**	**DF**	**F2F**	**FS**	**NT**	**All**
10 videos	89.18	76.6	90.89	93.53	92.81	76.06	59.84	81.15	76.73	67.71	64.55	53.99	60.04	65.14	60.55
50 videos	99.52	98.84	97.56	96.67	95.89	92.48	91.33	92.63	85.98	82.89	75.53	66.44	74.25	71.48	65.76
100 videos	99.51	99.09	98.64	98.23	97.54	95.39	95.8	95.56	90.09	87.19	83.68	72.69	79.56	73.72	66.81
300 videos	**99.59**	**99.53**	**98.78**	**98.73**	**98.88**	**97.30**	**97.41**	**97.51**	**92.4**	**92.65**	**91.57**	**86.38**	**88.35**	**79.65**	**76.01**

Table 6: Analysis of the training corpus size. Numbers reflect the accuracies of the XceptionNet detector trained on single and all manipulation methods (DF: DeepFakes, F2F: Face2Face, FS: FaceSwap, NT: NeuralTextures and All: all manipulation methods).

	Raw					Compressed 23					Compressed 40				
	DF	**F2F**	**FS**	**NT**	**P**	**DF**	**F2F**	**FS**	**NT**	**P**	**DF**	**F2F**	**FS**	**NT**	**P**
Average	77.60	49.60	76.12	32.28	78.19	78.17	50.19	74.80	30.75	75.41	73.18	43.86	64.26	39.07	62.06
Desktop PC	**80.41**	**53.73**	75.10	**34.36**	**80.12**	**81.17**	**51.57**	**79.51**	29.50	**78.10**	71.71	**44.09**	62.99	35.71	**64.32**
Mobile Phone	74.80	44.96	**77.11**	30.58	76.40	75.47	48.95	70.08	**31.97**	72.84	**74.62**	43.62	**65.50**	**41.85**	60.00

Table 7: User study result w.r.t. the used device to watch the images (DF: DeepFakes, F2F: Face2Face, FS: FaceSwap, NT: NeuralTextures and P: Pristine). 99 participants used a PC and 105 a mobile phone.

C. Classification of Manipulation Method

To train the XceptionNet classification network to distinguish between all four manipulation methods and the pristine images, we adapted the final output layer to return five class probabilities. The network is trained on the full dataset containing all pristine and manipulated images. On raw data the network is able to achieve a 99.03% accuracy, which slightly decreases for the high quality compression to 95.42% and to 80.49% on low quality images.

D. Hyperparameters

For reproducibility, we detail the hyperparameters used for the methods in the main paper. We structured this section into two parts, one for the manipulation methods and the second part for the classification approaches used for forgery detection.

D.1. Manipulation Methods

DeepFakes and *NeuralTextures* are learning-based, for the other manipulation methods we used the default parameters of the approaches.

DeepFakes: Our *DeepFakes* implementation is based on the *deepfakes faceswap github project* [1]. MTCNN ([65]) is used to extract and align the images for each video. Specifically, the largest face in the first frame of a sequence is detected and tracked throughout the whole video. This tracking information is used to extract the training data for *DeepFakes*. The auto-encoder takes input images of 64 (default). It uses a shared encoder consisting of four convolutional layers which downsizes the image to a bottleneck of 4×4, where we flatten the input, apply a fully connected layer, reshape the dense layer and apply a single upscaling using a convolutional layer as well as a pixel shuffle layer (see [54]). The two decoders use three identical up-scaling layers to attain full input image resolution. All layers use Leaky ReLus as non-linearities. The network is trained using Adam with a learning rate of 10^{-5}, $\beta_1 = 0.5$ and $\beta_2 = 0.999$ as well as a batch size of 64. In our experiments, we run the training for 200000 iterations on a cloud platform. By exchanging the decoder of one person to another, we can generate an identity-swapped face region. To insert the face into the target image, we chose Poisson Image Editing [49] to achieve a seamless blending result.

NeuralTextures: *NeuralTextures* is based on a U-Net architecture. For data generation, we employ the original pipeline and network architecture (for details see [57]). In addition to the photo-metric consistency, we added an adversarial loss. This adversarial loss is based on the patch-based discriminator used in Pix2Pix [36]. During training we weight the photo-metric loss with 1 and the adversarial loss with 0.001. We train three models per manipulation for a fixed 45 epochs using the Adam optimizer (with default settings) and manually choose the best performing model based on visual quality. All manipulations are created at a resolution of 512×512 as in the original paper, with a texture resolution of 512×512 and 16 feature per texel. Instead of using the entire image, we only train and modify the cropped image containing the face bounding box ensuring high resolution outputs even on higher resolution images. To do so, we enlarge the bounding box obtained by the Face2Face tracker by a factor of 1.8.

D.2. Classification Methods

For our forgery detection pipeline proposed in the main paper, we conducted studies with five classification approaches based on convolutional neural networks. The networks are trained using the Adam optimizer with different parameters for learning-rate and batch-size. In particular, for the network proposed in *Cozzolino at al.* [17] the used learning-rate is 10^{-5} with batch-size 16. For the proposal of *Bayar and Stamm* [10], we use a learning-rate equal to 10^{-5} with a batch-size of 64. The network proposed by *Rahmouni* [51] is trained with a learning-rate of 10^{-4} and a batch-size equal to 64. *MesoNet* [5] uses a batch-size of 76 and the learning-rate is set to 10^{-3}. Our *XceptionNet* [14]-based approach is trained with a learning-rate of 0.0002 and a batch-size of 32. All detection methods are trained with the Adam optimizer using the default values for the moments ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$).

We compute validation accuracies ten times per epoch and stop the training process if the validation accuracy does not change for 10 consecutive checks. Validation and test accuracies are computed on 100 images per video, training is evaluated on 270 images per video to account for frame count imbalance in our videos. Finally, we solve the imbalance between real and fake images in the binary task (i.e., the number of fake images being roughly four times as large as the number of pristine images) by weighing the training images correspondingly.