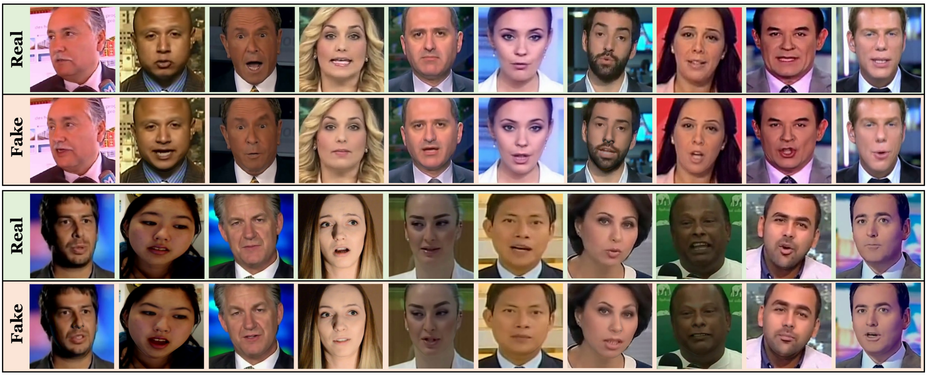
FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces

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



Abstract. With recent advances in computer vision and graphics, it is now possible to generate videos with extremely realistic synthetic faces, even in real time. Countless applications are possible, some of which raise a legitimate alarm, calling for reliable detectors of fake videos. In fact, distinguishing between original and manipulated video can be a challenge for humans and computers alike, especially when the videos are compressed or have low resolution, as it often happens on social networks. Research on the detection of face manipulations has been seriously hampered by the lack of adequate datasets. To this end, we introduce a novel face manipulation dataset of about half a million edited images (from over 1000 videos). The manipulations have been generated with a state-of-the-art face editing approach. It exceeds all existing video manipulation datasets by at least an order of magnitude. Using our new dataset, we introduce benchmarks for classical image forensic tasks, including classification and segmentation, considering videos compressed at various quality levels. In addition, we introduce a benchmark evaluation for creating indistinguishable forgeries with known ground truth; for instance with generative refinement models.

Keywords: Image Forensics, Video Manipulation, Facial Reenactment

1 Introduction

Faces play a central role in human interaction, as the face of a person can emphasize a message or it can even convey a message in its own right [1]. In particular, for faces, we have seen stunning progress in image and video manipulation methods in recent years. State-of-the-art methods can now generate manipulated videos in real time [2], can synthesize videos based on audio input [3], or can artificially animate static images [4]. At the same time, the ability to edit facial expressions has also gained tremendous attention in the context of fake-news discussions and in the current political climate in many countries. The ability to effortlessly create visually plausible editing of faces in videos has the potential to severely undermine trust in any form of digital communication. For instance, in social networks, filtering out or tagging manipulated images is currently one of the most problematic issues. Furthermore, the authenticity of face pictures also plays a role in completely different applications, such as biometric access control [5,6].

In this context, image forensics research has recently gained momentum in examining the authenticity of images. Here, we believe the recent advances in deep learning offer a unique opportunity due to the ability to learn extremely powerful image features with convolutional neural networks (CNNs). In particular, supervised training has shown to produce extremely impressive results, and we speculate that they could be well-suited to robustly identify manipulations. Unfortunately, these methods rely on large amounts of training data, and most forensic datasets to date are manually created, thus limited in size. This lack of available training data is a severe bottleneck for training deep networks for manipulation detection and makes it hard to evaluate different methods.

In order to alleviate this shortage of training samples, we introduce a comprehensive dataset of facial manipulations composed of over 500,000 frames from 1004 videos using the state-of-the-art Face2Face approach [2]. We consider two types of manipulation: *source-to-target*, where we transfer facial expressions from a source video to a target video using Face2Face, and *self-reenactment*, where we use Face2Face to reenact the facial expressions of a source video. In addition, we provide the reconstructed face masks generated by Face2Face for all videos in the source-to-target dataset.

Thanks to the *source-to-target* dataset, we can carry out a forensic analysis and train data reliant algorithms in a realistic scenario, given that the source and target videos were retrieved from YouTube. In particular, we evaluate the performance of a variety of methods on two main tasks: forgery *classification* (is anything in an image forged?) and *segmentation* (is the current pixel forged?). Performance is analyzed on manipulated videos compressed at various quality levels to account for the typical processing encountered when the video is uploaded on the internet. This is a very challenging situation since low-level manipulation traces can get lost after compression.

In addition to classification and evaluation, the *self-reenactment* dataset allows us to evaluate generative methods. In particular, the generation process can start from an already well-structured fake, which helps us focus on refinement in a possibly supervised environment, a problem resembling synthetic-to-real translations [7]. Furthermore, the performance of our refinement models can be evaluated using forgery detection approaches, without resorting to subjective metrics, such as visual user studies. Here, we introduce an evaluation scheme based on creating indistinguishable images based on generative models with known ground truth.

In summary, we introduce two versions of a novel dataset of manipulated facial expressions composed of more than 500,000 images from 1004 videos with pristine sources and target ground truth in Section 3. In particular, our new dataset focuses on the following problem statements:

- How well do current state-of-the-art approaches perform in a realistic setting both for forgery detection (Section 4) and segmentation (Section 5)?
- Can we use generative networks to improve the quality of forgeries (Section 6)?

2 Related Work

Face Manipulation Methods. In the last two decades interest in virtual face manipulation has rapidly increased. Breglera *et al.* [8] 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 [9], 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.* [10] presented a similar system that replaces the face of an actor while preserving the original expressions. VDub [11] 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.* [12] 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.* [2], 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 [13].

Recently, Suwajanakorn *et al.* [3] learned the mapping between audio and lip motions, while their compositing approach builds on similar techniques to Face2Face [2]. Averbuch-Elor *et al.* [4] 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. Other editing by use 3D proxies for 3D object manipulation in a single photograph using stock 3D models [14], physics-based edits in videos [15,16].

Recently, several face image synthesis approaches using deep learning techniques have been proposed. Lu *et al.* [17] provides an overview. Generative adversarial networks (GANs) are used to apply Face Aging [18], to generate new viewpoints [19], or to alter face attributes like skin color [20]. Deep Feature Interpolation [21] shows impressive results on altering face attributes like age, mustache, smiling etc. Similar results of attribute interpolations are achieved by Fader Networks [22]. Most of these deep learning based image synthesis techniques suffer from low image resolutions. Karras *et al.* [23] improve the image quality using progressively growing of GANs. Their results include 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 [24,25]. Recently, several CNN-based solutions have been proposed in image forensics [26,27,28,29]. 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 [30,31,32], varying interpolation types [33], copy-move manipulations [34,35], or chroma-key compositions [36]. The proposed face benchmark fills this gap in the research landscape by providing a huge video dataset of advanced synthesized faces.

For forensics specifically on faces, some methods have been proposed to distinguish computer generated faces from natural ones [37,38,39], and to detect face retouching [40]. In biometry, Raghavendra *et al.* [41] recently proposed to detect morphed faces with two pre-trained deep CNNs, VGG19 and AlexNet. Finally, Zhou *et al.* [42] proposed detection of two different face swapping manipulations using a two-stream network: one stream detects low-level inconsistencies between image patches while the other stream explicitly detects tampered faces.

However, robustness issues are addressed only in very few works, even though it is of paramount importance for practical applications. For example, operations like compression and resizing are known for laundering manipulation traces from the data. Unfortunately, compression and resizing are routinely carried out when images and videos are uploaded to social networks, which is one of the most typical application fields for forensic analysis. An even greater challenge to a forensic detector are targeted attacks that consist of suitable post-processing steps to hide the traces of manipulation. All these attacks go under the collective name of counter-forensics [43]. Forensic analysis and counter-forensics are in continuous competition. Model-based methods appear to be extremely fragile on laundered data since they focus on specific image features which oftentimes disappear with post-processing. Data-driven methods can be expected to be more robust, especially if they rely on data which have a processing history coherent with the asset of interest [44]. A key benefit of the proposed dataset is that its size lifts video forensics research to a level that allows to create better detectors, but also better counter-forensics methods on a significant amount of data. At the same time, the dataset serves as a unified benchmark.

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. Most notably, the "Dresden image database" consists of 14,000 images from 73 cameras, and is used primarily for camera fingerprinting [45]. The recent VISION dataset also aims at camera fingerprinting, with 34,427 images and 1914 videos that were uploaded and downloaded from social media [46].

While several datasets were proposed that include image manipulations, only a few of them address also the important case of video. For image copy-move manipulations a large dataset is MICC_F2000 consisting of a collection of 700 forged images from various sources [47]. Datasets containing very different and realistic image manipulations are the First IEEE Image Forensics Challenge Dataset[1], which comprises a total of 1176 forged images, the Wild Web Dataset [48] with 90 real cases of manipulations coming from the web and the Realistic Tampering dataset [49] including 220 forged images.

More recently, Al-Sanjary *et al.* presented 33 videos on YouTube that contain different manipulations [50]. The National Institute of Standards and Technology (NIST) presented with the Nimble Challenge 2017 a large benchmark dataset [51]. However, it contains a total of 2520 manipulated images, but only 23 manipulated videos with ground truth. A database of 2010 FaceSwap- and SwapMe-generated images has recently been proposed by Zhou *et al.* [42]. While this dataset is most similar to our proposed benchmark, it is orders of magnitude smaller, and only consists of still images instead of videos.

3 The FaceForensics Dataset

We introduce the *FaceForensics* dataset which is created from 1004 videos (i.e., unique identities). In the following, we describe the data collection and processing used to generate our two datasets. The first dataset (see Section 3.1) contains manipulated videos where the source and target video differs, while the second dataset (see Section 3.2) consists of videos where Face2Face is used to reproduce the input video (i.e., source and target video are the same). This second dataset gives us access to ground truth pairs of synthetic and real images.

Data Collection The data was collected from YouTube. We chose videos with a resolution larger than 480p that were tagged with ”face”, ”newscaster” or ”newsprogram” on the youtube8m dataset [52] as well as other videos that were found on YouTube with these tags. We use the Viola-Jones [53] face detector to extract video sequences that contain a face for more than 300 consecutive frames. In addition to that, we perform a manual screening of the resulting clips to ensure a high quality of video selection and to avoid videos with face occlusions.

Data Processing To process the video data, we use a variant of the state-of-the-art Face2Face approach [2], that is able to fully-automatically create reenactment manipulations. The technique re-renders the face in a target video under possibly different expressions. We process each video in a preprocessing pass; here, we use the first frames in order to obtain a temporary face identity (i.e., 3D model), and track the expressions over the remaining frames. In order to improve the identity fitting and the static texture, we select the frames with the left- and rightmost angle of the face in an automated way; in the original implementation of Face2Face this step has to be done manually. Using these poses, we jointly fit the identity and estimate a static texture. Based on this identity reconstruction, we track the whole video to compute the per frame expression, rigid pose, and lighting parameters.

The generated tracking and reconstruction results allow us to generate any source-target video combinations for the reenactment. We generate the reenactment video by

[1] http://ifc.recod.ic.unicamp.br/fc.website/index.py?sec=0

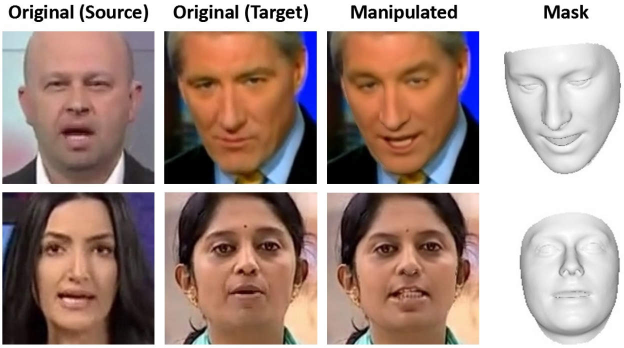


Fig. 1. Source-to-Target Dataset. From left to right: original input image of the source actor, input image of the target actor, reenactment result and face mask that is used during synthesis of the output image.

transferring the source expression parameters (i.e., 76 Blendshape coefficients) to the target video. A detailed explanation of the reenactment process can be found in the original paper [2]. As the result, we store the original source, the target image, and the manipulated output image for each frame. In addition, we generate a per-pixel binary mask of the modified pixels, which serves as ground truth for segmentation tasks.

3.1 Source-to-Target Reenactment Dataset

For the *source-to-target* dataset, we use the original Face2Face reenactment approach between two randomly chosen videos (see Fig. 1). The technique uses a mouth retrieval approach that selects the mouth interiors from a mouth database based on the target expressions. This person specific mouth database is built upon the tracked videos in the preprocessing step (i.e., contains images of the target video). The mouth database is one of the most limiting factors of the Face2Face approach, since the videos may not cover a variety of mouth expressions, leading to distortions of the mouth in the resulting reenactment output. The dataset is split into 704 videos for training (364,256 images), 150 videos for validation (76,309 images), and 150 videos for testing (78,562 images). We use the *source-to-target* reenactment dataset for all testing, as well as for training all classification and segmentation approaches; see Section 4 and Section 5.

3.2 Self-Reenactment Dataset

The second dataset is built upon *self-reenactment* generated by Face2Face (see Fig. 2). Instead of different source and target video combinations, the self reenactment scenario uses the same video as source and target video. Applying this reenactment technique to a video, we obtain video pairs consisting of ground truth data and manipulated (re-rendered) facial imagery. These ground truth pairs are ideally suited for training generative approaches for FaceForensics, which we explore in Section 6. We split the *self-reenactment* dataset into the same 704 videos for training (368,135 images), 150 videos for validation (75,526 images), and 150 videos for testing (77,745 images).

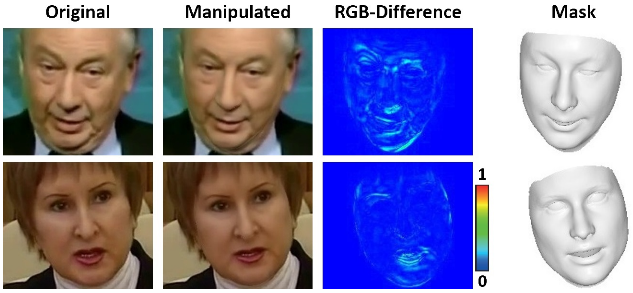


Fig. 2. Fake or Real? Examples of the FaceForensics Self-Reenactment Dataset. From left to right: original input image, self-reenacted output image, color difference plot and face mask that is used during synthesis of the output image.

4 Forgery Classification Task

The forgery classification task has the goal to identify forged images. It is cast as a binary classification problem on a per frame basis of the manipulated videos. Since there are no specific approaches in the current literature to detect Face2Face manipulations, we decided to consider learning-based methods used in the forensic community for generic manipulation detection [26,27], computer-generated vs natural image detection [39] and face tampering detection [41,42]. In addition, we also included a state-of-the-art deep network [54]. Each of these methods is trained on the same source-to-target reenactment dataset comprising 10 frames from each of the 704 forged and 704 pristine videos. Likewise, the validation and test set both consist of 10 frames extracted from each of 150 (pristine) and 150 (fake) videos. For each frame, we crop all images to be centered around the face, where we make use of the face mask provided by Face2Face. The faces have been resized to the input size of the network when requested [42,54], otherwise, a clip of 128x128 pixels centered on the face has been extracted as input [26,27,39].

For all baselines, we evaluate classification accuracy on uncompressed data, on H.264 compressed data with quantization parameter equal to 23 (light compression) and 40 (strong compression), to cover the quality parameters of a range of different distribution channels, including popular social networks. A sample frame extracted from these three settings is shown in Fig. 3. In the following, we briefly describe all the approaches used for comparison.

Steganalysis Features + SVM: it is a handcrafted solution based on the extraction of co-occurrences on 4 pixels patterns along the horizontal and vertical direction on the high-pass images, proposed originally in steganalysis [55], using only one single model (for a total feature length of 162) which was the winning approach in the first IEEE Image forensic Challenge [56]. These features are then used to train a linear SVM classifier.

Cozzolino et al. 2017 [27]: use a CNN-based network implementing the handcrafted features described above. The network is then fine-tuned on our dataset.

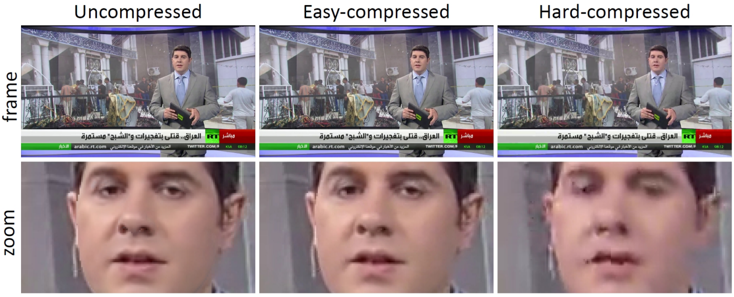


Fig. 3. Uncompressed frame (left), easy-compressed (middle), and hard-compressed one (right).

Bayar and Stamm 2016 [26]: propose a CNN-based network with 8 layers: a constrained convolutional layer, 2 convolutional layers, 2 max-pooling layers and 3 fully-connected layers. The constrained convolutional layer is specifically designed to suppress the high-level content of the image.

Rahmouni et al. 2017 [39]: adopt different CNN architectures with a global pooling layer that computes four statistics (mean, variance, maximum and minimum). We consider the network that had the best performance (Stats-2L).

Raghavendra et al. 2017 [41]: use two pre-trained CNNs VGG19 and AlexNet. The networks are fine-tuned on our dataset, then the feature vectors extracted from the first fully connected layer of the two networks are concatenated and used as input for the Probabilistic Collaborative Representation Classifier.

Zhou et al. 2017 [42]: consider a two-stream network, a pre-trained deep CNN, fine-tuned on our dataset, (GoogLeNet Inception V3 model) and a patch triplet stream trained on 5514D steganalysis features [57]. The final score is then obtained by combining the output scores of the two streams.

In addition to these approaches, we also evaluate a transfer learning model of the state-of-the-art XceptionNet CNN architecture [54]. It is based on depthwise separable convolution layers with residual connections. XceptionNet is pre-trained on ImageNet and fine-tuned on our source-to-target reenactment dataset. During fine-tuning, we freeze the first 36 layers which corresponds to the first 4 blocks of the network. Only the last layer is replaced by a dense layer with two outputs, initialized randomly and trained anew for 10 epochs. After that, we train the resulting network until the validation does not change in 5 consecutive epochs. For optimization, we use the following hyperparameters for our reported scores: ADAM [58] with a learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as well as a batch-size of 64.

In Tab. 1, we show a comparison of these methods applied to uncompressed and compressed videos. In the absence of compression, all methods, including [55] based on handcrafted features, achieve a relatively high performance. For compressed videos, performance drops, particularly for handcrafted features and for shallow CNN architectures [26,27]. Deep neural networks are better at handling these situations, with XceptionNet slightly outperforming the method by Zhou *et al.* [42]. On the other hand, even humans have a hard time detecting manipulations under strong compression as shown in Section 6.1.

Methods	no-c	easy-c	hard-c
[55] Steganalysis Features + SVM	99.40	75.87	58.16
[27] Cozzolino *et al.*	99.60	79.80	55.77
[26] Bayar and Stamm	99.53	86.10	73.63
[39] Rahmouni *et al.*	98.60	88.50	61.50
[41] Raghavendra *et al.*	97.70	93.50	82.13
[42] Zhou *et al.*	99.93	96.00	86.83
[54] XceptionNet	99.93	98.13	87.81

Table 1. Classification accuracy (face-level detection; i.e., is a face manipulated or not) of reference methods with no compression (*no-c*), light compression (*easy-c*), and strong compression (*hard-c*) using our FaceForensics benchmark dataset.

5 Forgery Segmentation Task

Pixel-level segmentation of manipulated images (also referred to as forgery localization in the forensics community) is a very challenging task. The most successful approaches proposed in the image forensics literature rely on camera-based artifacts (e.g. sensor noise, demosaicking). However, their application on the frames extracted from our dataset did not provide satisfactory results, not even for uncompressed data. Hence, we discard them and focus only on deep learning methods, which can take full advantage of our dataset for training. In particular, those proposed in [27] and [39] already perform localization and need no further adaptation.

Additionally, considering its very good performance in classification, we adapt also XceptionNet [54] to the localization task, as described in the following.

At test time, the network runs in sliding-window modality on patches of 128 × 128 pixels, with stride 16. For each patch, W_i, it outputs the estimated manipulation probability, $\hat{p}_i = \mathrm{CNN}(W_i)$, which is assigned to the central 16 × 16 region.

Preliminary to training, a ground truth is computed by labeling as manipulated all pixels that have been modified with respect to the pristine frame. Spurious pixels are removed by morphological filtering, and a spatial filtering is performed to smooth boundaries. Eventually, the ground truth pixels range from 0 (pristine background) to 1 (manipulated face), with intermediate values on the boundaries, and such values are regarded as manipulation probabilities p_i. These will be used to compute the loss function as the cross-entropy between ground-truth and estimated probabilities $\sum_i -(p_i)\log(\hat{p}_i) - (1-p_i)\log(1-\hat{p}_i)$, where the sum goes over all patches of a mini-batch, and p_i is the ground truth probability of the central pixel.

The patch-level training set is formed by taking 10 frames from each training set video, and 3 patches from each frame, one from the face, one from the background, and one over the face-background boundary. Training is performed using ADAM, with mini-batches of 96 patches, formed by taking the 3 (pristine) plus 3 (fake) patches associated with 16 forged frames and with the corresponding 16 pristine frames. For each epoch, the frames are shuffled, preserving the correspondence between pristine and forged patches. We use a learning rate of 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, batch-size 96, and again train the resulting network until the validation does not change in 5 consecutive epochs.

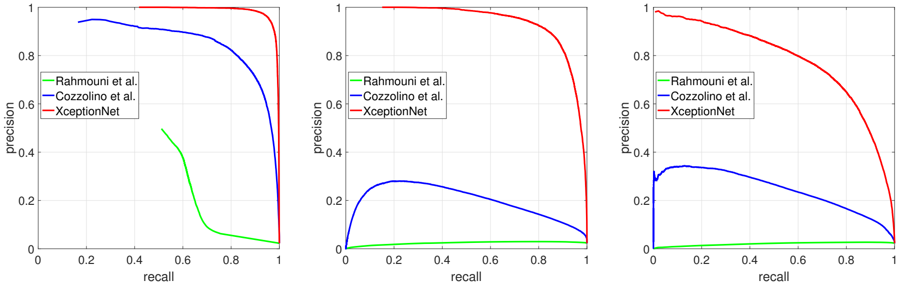


Fig. 4. Precision vs recall on uncompressed videos (left), easy-compressed (middle), and hard-compressed (right). The test set comprises both forged and pristine images.

In Fig. 4, we show a quantitative evaluation of these methods. For all of them, performance degrades with increasing compression rate, as more and more false positives and false negatives occur. Eventually, at the highest compression rate, only the method based on XceptionNet keeps providing good results. In Fig. 5 and Fig. 6, we also show visual results over both uncompressed and compressed data. In this last case, we only show results provided by XceptionNet, since the other two methods output useless heatmaps.

6 Refinement Task

Section 4 shows that Face2Face manipulations can be detected quite easily in an uncompressed setting with a sufficiently large amount of data. This gives rise to the question whether such an amount of data can also be used in the opposite direction, i.e., to improve the quality of the manipulations. To this end, we leverage the self-reenactment dataset which contains 521,406 manipulated frames with target ground truth pairs for supervised training.

As a baseline, we devise an autoencoder CNN architecture with skip connections that takes as input a 128 × 128 pixels image and predicts an image of the same resolution (see Fig. 7 for the detailed architecture). In order to obtain meaningful and strong features for images of human faces, we first pre-train the autoencoder network on a self-reconstruction task using the VGGFace2 dataset [59] in an unsupervised fashion. This dataset contains 3.31 million images of 9131 subjects, which is about an order of magnitude more than our dataset but does not provide annotations. In this pre-training process, where we use all of these images for training, we disable the skip connections, thus forcing the networks to solely rely on the bottleneck layer.

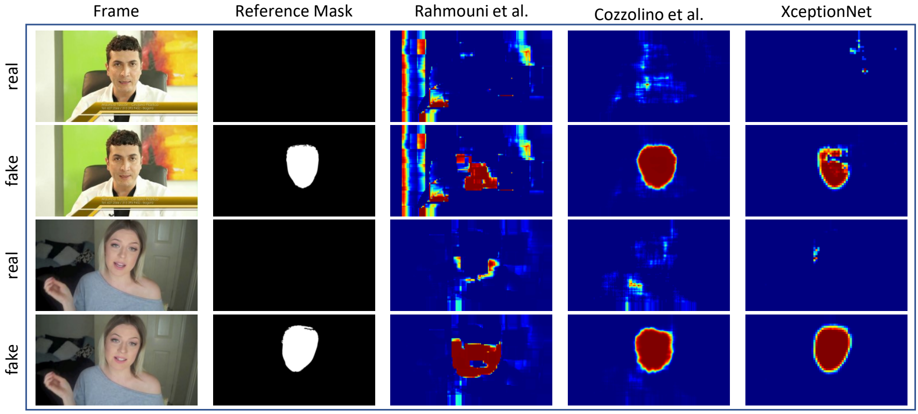


Fig. 5. Forgery segmentation results. For each frame, we show the heatmaps for the original video (first row) and the manipulated one (second row). From left to right: input frame, ground truth mask (only for the fake input), results of Rahmouni *et al.* [39], Cozzolino *et al.* [27], and the XceptionNet-based method. Both Cozzolino *et al.* and XceptionNet reliably localize the manipulations on uncompressed data.

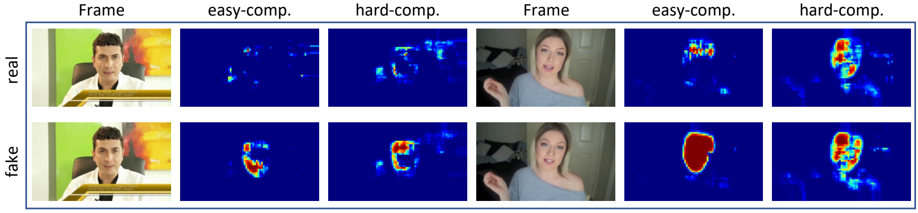


Fig. 6. Forgery segmentation vs. compression. With increasing compression rate, the segmentation results get worse. The XceptionNet-based method is still able to segment the manipulation in most cases, even under hard compression.

We then fine tune the pre-trained autoencoder network on our FaceForensic self-reenactment dataset using the 368,135 training images. Here, we input the manipulated faces and constrain it with the known target ground truth using an ℓ_1 loss in the supervised training process; note that we aim to minimize the difference image, which is a widely-used technique. In addition, we enable the skip connections which allow us to obtain sharper results in the autoencoder output. At test time, we feed in data from the FaceForensic source-to-target test dataset in order to improve the quality of forgeries. We optimize the network with ADAM using a batch size of 32, a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and continue training until convergence on the self-reenactment validation set.

The main advantage of an autoencoder architecture over a network without a bottleneck layer is the ability to leverage the larger (unlabeled) VGGFace2 dataset for unsupervised pre-training.

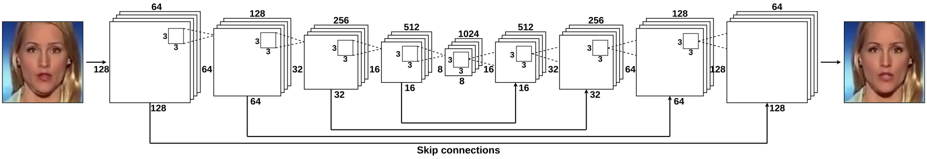


Fig. 7. The autoencoder (AE) architecture with skip connections used for refining the forged images. The AE is first pre-trained on the significantly larger, but unlabeled, VGGFace2 dataset in an unsupervised fashion (w/o enabling the skip connections). We then fine tune on our self-reenactment training set using supervision with Face2Face and target ground truth training pairs.

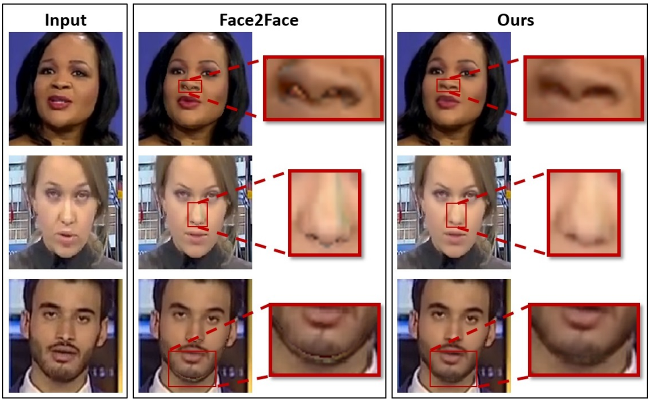


Fig. 8. Refinement of our autoencoder approach: we can see in the close-ups that our refinement significantly improves the visual result of Face2Face [2]. Especially, regions around the nose, the chin, and the cheek, where most of the artifacts of the Face2Face method occur, are corrected. Errors occurring in the transition between the mask region, hosting the reconstructed and modified face, and the unmodified background are removed by our method. The autoencoder also improves regions where a wrong illumination estimate in the Face2Face algorithm leads to artifacts (e.g., see second row).

6.1 Perceptual Evaluation of the Refinement

Fig. 8 shows a qualitative comparison of the Face2Face reenactment approach and our refinement results using our source-to-target test dataset. It shows that most artifacts of Face2Face are visible in the area of the chin, the cheek, and the nose. These are border regions of the face mask that is used to re-synthesize the face and the original image. The autoencoder significantly improves these regions and nicely blends between foreground manipulated face, delimited by the mask region, with the target video stream in the background. Illumination errors in the Face2Face output are also corrected by our method.

In order to compare the visual quality of forged images obtained with Face2Face and our refinement network, we conduct a user study with 14 participants, whose results are shown in Tab. 2. The 14 participants are Master and Ph.D. students in computer science who are not involved in this project. For the study, we randomly choose 50

User Study	no-c	easy-c	hard-c
Face2Face: w/o AE	68.71	62.00	50.00
Face2Face: Refined w/ AE	60.57	51.29	48.93

Table 2. User study results on images generated with the raw Face2Face output (top) and with the proposed refinement approach (bottom). Candidates are shown an image for three seconds and have to classify it into real or fake. We see that it is noticeably harder for humans to identify forgeries after refinement.

images from each of the no-compression and easy-compression, and 20 from the hard-compression categories. All images are taken from the source-to-target test set at a 128^2 resolution, and we select images at a ratio of 50% pristine and 50% forged; i.e., we have 25 fake and 25 pristine images for the no-c and easy-c categories, and 10 fake and 10 pristine images on the hard-c one. For each participant, we randomly shuffle the images within each of the compression categories. Before showing an image from a category, we let the participants know which category is being presented; i.e., explain details regarding compression. We show each image for three seconds to a participant, then the image goes blank, and either real or fake has to be chosen. This process is repeated for each image and for each compression category. We conduct this experiment for the raw Face2Face output [2] and for our refined results obtained with the autoencoder network.

Quantitative results show that that humans are worse at identifying manipulations than the XceptionNet-based approach. For highly-compressed images, this becomes particularly obvious, as human accuracy is about 50%, which is essentially random guessing. For the easier compression setups, participants are able to identify better than random chance; however, accuracy is still relatively low. We can also clearly observe that our autoencoder refinement makes visual differences even harder to spot, thus increasing the quality of forgeries for human observers.

6.2 Quantitative Evaluation

However, we can also evaluate our refiner with the classification methods described in Section 4. As we aim to improve the quality of our fakes, the created data should be more difficult to detect than without refinement under the same circumstances, namely identical classifier architecture and amount of training data. Therefore, we use the same evaluation protocol as in Section 4, i.e., we refine 10 images for every video in our source-to-target training, validation and test set. In addition to that, we resize face images to 128×128 pixels for a fair comparison between the refined and raw images and retrain XceptionNet on the resulting dataset as in Section 4.

In Table 3 we observe that the autoencoder slightly lowers the detection accuracy on compressed data, but it does not affect the overall performance by a large margin. Therefore, even if the visual quality of fakes seems to be high, there are still many shortcomings that make these methods easy to detect for forgery detection algorithms as the classifier is still able to detect refined fakes with high accuracy, which suggests

Datasets on 128x128	no-c	easy-c	hard-c
Face2Face: w/o AE	99.42	96.17	84.56
Face2Face: Refined w/ AE	99.23	96.07	80.97

Table 3. Classification accuracy of a XceptionNet on our source-to-target test dataset using images of 128 × 128 pixels. In the top row, we show results on the fake data directly generated by Face2Face; in the second row, we use our autoencoder refiner that is trained on the self-reenactment training set. The autoencoder succeeds to slightly lower detection performances under strong compression.

that visual results itself seem to be a poor metric. One possibility to circumvent this problem and produce high quality refinements would be generative adversarial networks [60], which have already been successfully applied to unsupervised refinement [7] and were shown to be able to produce high-resolution results [23].

7 Conclusions

In this work, we introduce a novel dataset of manipulated videos that exceeds all existing publicly available forensic datasets by orders of magnitude. We provide a benchmark for general image forensic tasks on this dataset such as identification and segmentation of forged images. We show that handcrafted approaches are highly challenged by realistic amounts of compression, whereas we set a strong baseline of results for detecting a facial manipulation with modern deep learning architectures.

We also introduce a second application of the dataset, by visually improving the quality of the forgery with an autoencoder that is trained in a supervised fashion on our self-reenactment dataset. However, our refiner mainly improves visual quality, but it only slightly encumbers forgery detection for deep learning method trained exactly on the forged output data. This motivates us to further investigate refinement methods in future work, as we believe that this interplay between tampering and detection is not only an extremely exciting avenue for follow-up work but also of utmost importance in order to build robust and generalizable classifiers.

8 Acknowledgement

We gratefully acknowledge the support of this research by the AI Foundation, a TUM-IAS Rudolf Mößbauer Fellowship, and Google Faculty Award. 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.

Appendix

A User Study Interface

Fig. 9 visualizes the interface for conducting our user study. We show participants a random images, and ask them to select fake or real. The images are randomly chosen from the no-, easy-, and hard-compression sets; however, we ensure an equal number of fake an real images for each participant.

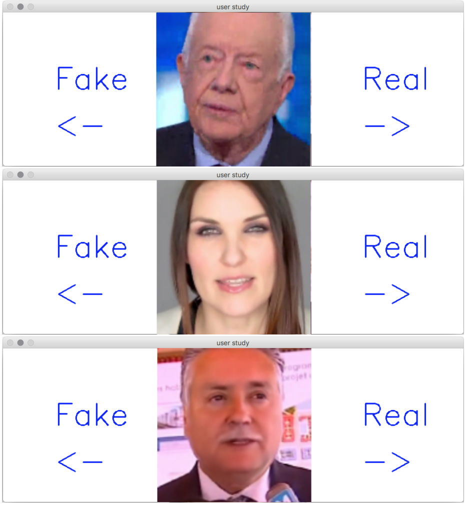


Fig. 9. Three exampes of our user interface; we show each image for three seconds, after the participant needs to select either fake or real. By the way, the top row is real (hard compression), the second row is fake (easy compression), the bottom row is fake with refinement (uncompressed).

B Forgery Segmentation Examples

In this section, we show additional qualitative results on forgery segmentation for compressed and uncompressed videos for three CNN-based architectures: Rahmouni et al. [39], Cozzolino et al. [27] and XceptionNet [54].

Fig. 10 shows results on umcompressed video where we can see that the XceptionNet model provides the best results compared to [39] and [27]. It is able to correctly locate the manipulated area on the fake videos, while on real videos, we can hardly notice any false positives.

On compressed videos, segmentation becomes more difficult. The methods proposed by Rahmouni et al. [39] and Cozzolino et al. [27] produce almost random heatmaps, while the XceptionNet model provides results at acceptable quality. In Fig. 11 we can see that with easy-compressed videos XceptionNet still works pretty well allowing reliable segmentation of altered pixels. If we again increase the compression rate, the task becomes even more challenging, and the resulting segmentation is rather poor, but much better than the compared methods (see Fig. 12).

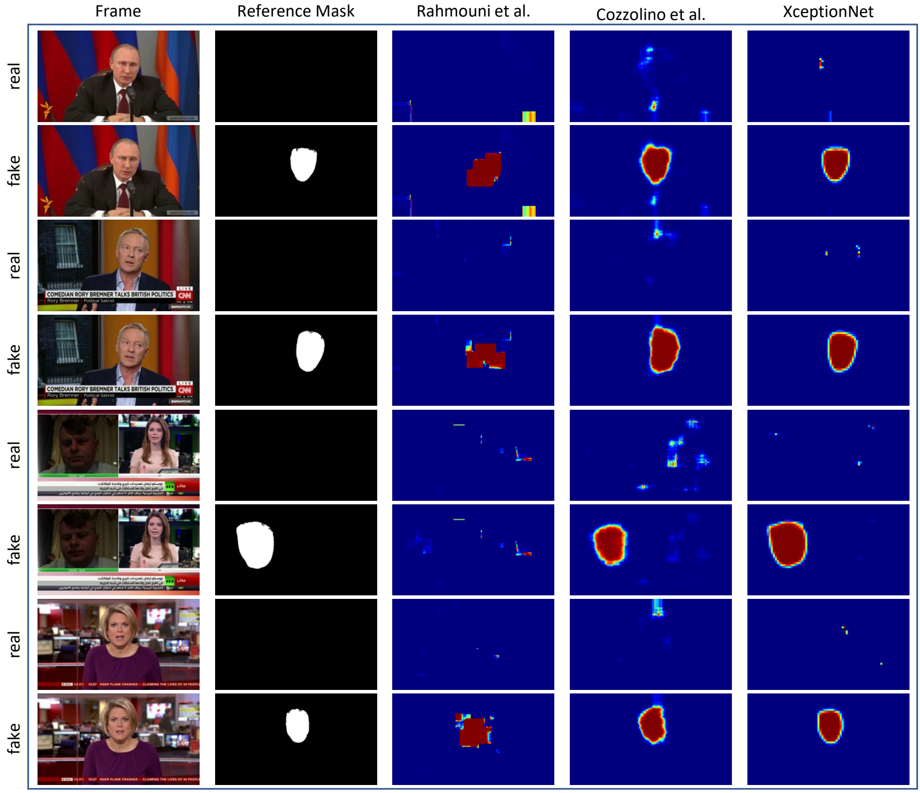


Fig. 10. Additional forgery segmentation examples. For each frame, we show the heatmaps for the original video (first row) and the manipulated one (second row). From left to right: input frame, ground truth mask (only for the fake input), results of Rahmouni *et al.* [39], Cozzolino *et al.* [27], and the XceptionNet-based method.

References

1. Frith, C.: Role of facial expressions in social interactions. Philosophical Transactions of the Royal Society B: Biological Sciences **364**(1535) (December 2009)
2. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In: IEEE Conference on Computer Vision and Pattern Recognition. (June 2016) 2387–2395
3. Suwajanakorn, S., Seitz, S.M., Kemelmacher-Shlizerman, I.: Synthesizing Obama: learning lip sync from audio. ACM Transactions on Graphics (TOG) **36**(4) (2017)

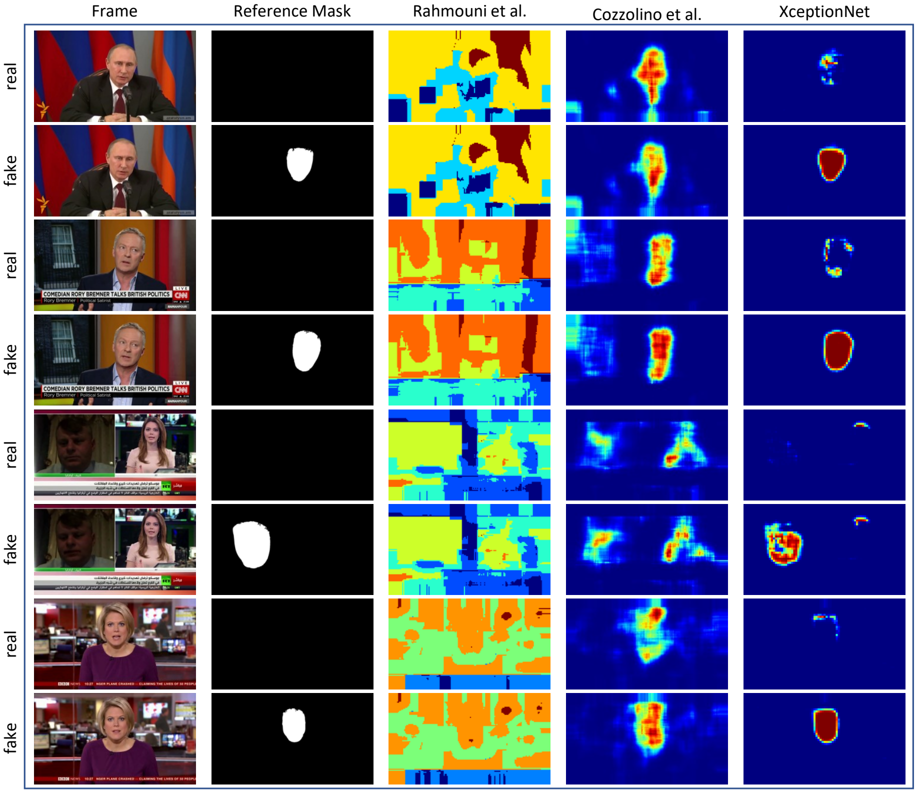


Fig. 11. Forgery segmentation examples on easy-compressed videos. For each frame, we show the heatmaps for the original video (first row) and the manipulated one (second row). From left to right: input frame, ground truth mask (only for the fake input), results of Rahmouni *et al.* [39], Cozzolino *et al.* [27], and the XceptionNet-based method.

4. Averbuch-Elor, H., Cohen-Or, D., Kopf, J., Cohen, M.F.: Bringing portraits to life. ACM Transactions on Graphics (Proceeding of SIGGRAPH Asia 2017) **36**(4) (2017) to appear
5. Jain, A.K., Ross, A., Prabhakar, S.: An introduction to biometric recognition. IEEE Transactions on Circuits and Systems for Video Technology **14**(1) (January 2004) 4–20
6. Ferrara, M., Franco, A., Maltoni, D.: The magic passport. In: IEEE International Joint Conference in Biometrics. (2014) 1–7
7. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 3. (2017) 2107–2116
8. Bregler, C., Covell, M., Slaney, M.: Video rewrite: Driving visual speech with audio. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '97, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co. (1997) 353–360
9. Dale, K., Sunkavalli, K., Johnson, M.K., Vlasic, D., Matusik, W., Pfister, H.: Video face replacement. ACM Trans. Graph. **30**(6) (December 2011) 130:1–130:10

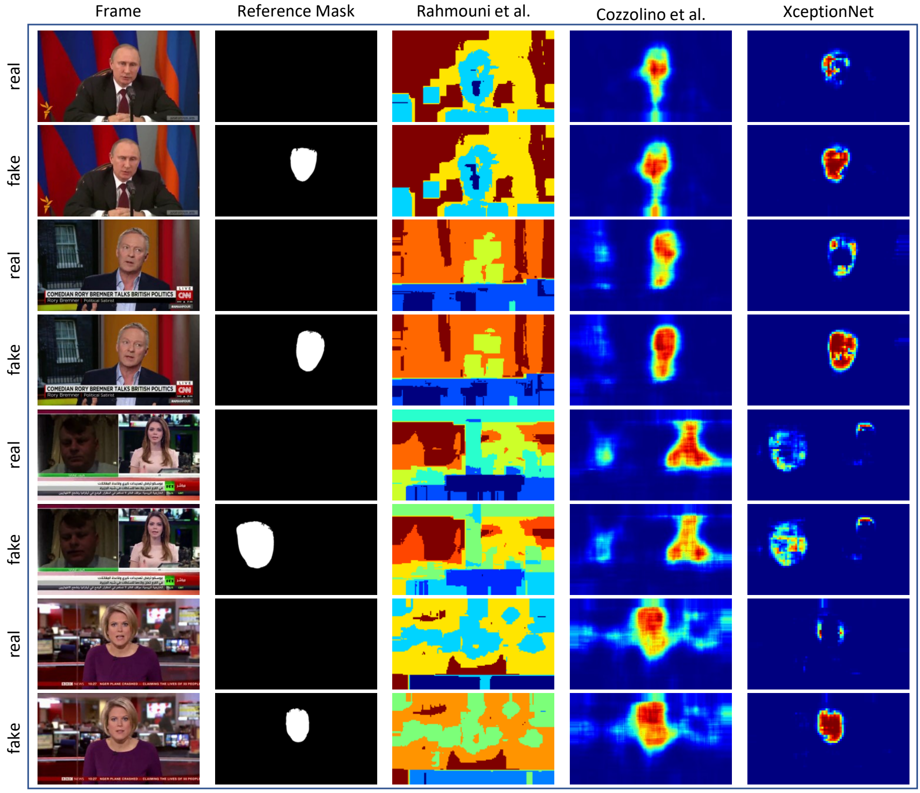


Fig. 12. Forgery segmentation examples on hard-compressed videos. For each frame, we show the heatmaps for the original video (first row) and the manipulated one (second row). From left to right: input frame, ground truth mask (only for the fake input), results of Rahmouni *et al.* [39], Cozzolino *et al.* [27], and the XceptionNet-based method.

10. Garrido, P., Valgaerts, L., Rehmsen, O., Thormaehlen, T., Perez, P., Theobalt, C.: Automatic face reenactment. In: IEEE Conference on Computer Vision and Pattern Recognition. (2014) 4217–4224
11. Garrido, P., Valgaerts, L., Sarmadi, H., Steiner, I., Varanasi, K., Perez, P., Theobalt, C.: Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. Computer Graphics Forum **34**(2) (2015) 193–204
12. Thies, J., Zollhöfer, M., Nießner, M., Valgaerts, L., Stamminger, M., Theobalt, C.: Real-time expression transfer for facial reenactment. ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia 2015 **34**(6) (2015) Art. No. 183
13. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Facevr: Real-time gaze-aware facial reenactment in virtual reality. ACM Transactions on Graphics 2018 (TOG) (2018)
14. Kholgade, N., Simon, T., Efros, A., Sheikh, Y.: 3d object manipulation in a single photograph using stock 3d models. ACM Transactions on Graphics (TOG) **33**(4) (2014) 127
15. Bazin, J.C., Yu, G., Martin, T., Jacobson, A., Gross, M., et al.: Physically based video editing. Computer Graphics Forum **35**(7) (2016) 421–429

16. Haouchine, N., Roy, F., Courtecuisse, H., Nießner, M., Cotin, S.: Calipso: Physics-based image and video editing through cad model proxies. arXiv preprint arXiv:1708.03748 (2017)
17. Lu, Z., Li, Z., Cao, J., He, R., Sun, Z.: Recent progress of face image synthesis. CoRR **abs/1706.04717** (2017)
18. Antipov, G., Baccouche, M., Dugelay, J.: Face aging with conditional generative adversarial networks. CoRR **abs/1702.01983** (2017)
19. Huang, R., Zhang, S., Li, T., He, R.: Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis. CoRR **abs/1704.04086** (2017)
20. Lu, Y., Tai, Y., Tang, C.: Conditional cyclegan for attribute guided face image generation. CoRR **abs/1705.09966** (2017)
21. Upchurch, P., Gardner, J.R., Bala, K., Pless, R., Snavely, N., Weinberger, K.Q.: Deep feature interpolation for image content changes. CoRR **abs/1611.05507** (2016)
22. Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., Ranzato, M.: Fader networks: Manipulating images by sliding attributes. CoRR **abs/1706.00409** (2017)
23. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive Growing of GANs for Improved Quality, Stability, and Variation. CoRR **abs/1710.10196** (2017)
24. Farid, H.: Photo Forensics. The MIT Press (2016)
25. Sencar, H.T., Memon, N.: Digital Image Forensics — There is More to a Picture than Meets the Eye. Springer (2013)
26. Bayar, B., Stamm, M.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: ACM Workshop on Information Hiding and Multimedia Security. (2016) 5–10
27. Cozzolino, D., Poggi, G., Verdoliva, L.: Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In: ACM Workshop on Information Hiding and Multimedia Security. (2017) 1–6
28. Bondi, L., Lameri, S., Güera, D., Bestagini, P., Delp, E., Tubaro, S.: Tampering Detection and Localization through Clustering of Camera-Based CNN Features. In: IEEE Computer Vision and Pattern Recognition Workshops. (2017)
29. Bappy, J., Roy-Chowdhury, A., Bunk, J., Nataraj, L., Manjunath, B.: Exploiting spatial structure for localizing manipulated image regions. In: IEEE International Conference on Computer Vision. (2017) 4970–4979
30. Wang, W., Farid, H.: Exposing Digital Forgeries in Interlaced and Deinterlaced Video. IEEE Transactions on Information Forensics and Security **2**(3) (September 2007) 438–449
31. Gironi, A., Fontani, M., Bianchi, T., Piva, A., Barni, M.: A video forensic technique for detection frame deletion and insertion. In: IEEE International Conference on Acoustics, Speech and Signal Processing. (2014) 6226–6230
32. Long, C., Smith, E., Basharat, A., Hoogs, A.: A C3D-based Convolutional Neural Network for Frame Dropping Detection in a Single Video Shot. In: IEEE Computer Vision and Pattern Recognition Workshops. (2017) 1898–1906
33. Ding, X., Gaobo, Y., Li, R., Zhang, L., Li, Y., Sun, X.: Identification of Motion-Compensated Frame Rate Up-Conversion Based on Residual Signal. IEEE Transactions on Circuits and Systems for Video Technology, in press (2017)
34. Bestagini, P., Milani, S., Tagliasacchi, M., Tubaro, S.: Local tampering detection in video sequences. In: IEEE International Workshop on Multimedia Signal Processing. (October 2013) 488–493
35. D'Amiano, L., Cozzolino, D., Poggi, G., Verdoliva, L.: 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)
36. Mullan, P., Cozzolino, D., Verdoliva, L., Riess, C.: Residual-based forensic comparison of video sequences. In: IEEE International Conference on Image Processing. (2017)

37. Dang-Nguyen, D.T., Boato, G., De Natale, F.: Identify computer generated characters by analysing facial expressions variation. In: IEEE International Workshop on Information Forensics and Security. (2012) 252–257
38. Conotter, V., Bodnari, E., Boato, G., Farid, H.: Physiologically-based detection of computer generated faces in video. In: IEEE International Conference on Image Processing. (Oct 2014) 1–5
39. Rahmouni, N., Nozick, V., Yamagishi, J., Echizeny, I.: Distinguishing computer graphics from natural images using convolution neural networks. In: IEEE Workshop on Information Forensics and Security. (2017) 1–6
40. Bharati, A., Singh, R., Vatsa, M., Bowyer, K.: Detecting facial retouching using supervised deep learning. IEEE Transactions on Information Forensics and Security **11**(9) (Sep 2016) 1903–1913
41. Raghavendra, R., Raja, K., Venkatesh, S., Busch, C.: Transferable Deep-CNN features for detecting digital and print-scanned morphed face images. In: IEEE Computer Vision and Pattern Recognition Workshops. (2017)
42. Zhou, P., Han, X., Morariu, V., Davis, L.: Two-stream neural networks for tampered face detection. In: IEEE Computer Vision and Pattern Recognition Workshops. (2017) 1831–1839
43. Böhme, R., Kirchner, M.: Counter-forensics: attacking image forensics. In Sencar, H., Memon, N., eds.: Digital Image Forensics. Springer (2012)
44. Boroumand, M., Fridrich, J.: Deep learning for detecting processing history of images. In: IS&T Electronic Imaging: Media Watermarking, Security, and Forensics. (2018)
45. Gloe, T., Böhme, R.: The 'Dresden Image Database' for benchmarking digital image forensics. In: Proceedings of the 25th Annual ACM Symposium On Applied Computing (SAC 2010). Volume 2., Sierre, Switzerland (March 2010) 1585–1591
46. Shullani, D., Fontani, M., Iuliani, M., Al Shaya, O., Piva, A.: VISION: a video and image dataset for source identification. EURASIP Journal on Information Security (December 2017) 2017:15
47. Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., Serra, G.: A SIFT-based forensic method for copy-move attack detection and transformation recovery. IEEE Transactions on Information Forensics and Security **6**(3) (March 2011) 1099–1110
48. Zampoglou, M., Papadopoulos, S., , Kompatsiaris, Y.: Detecting image splicing in the wild (Web). In: IEEE International Conference on Multimedia & Expo Workshops (ICMEW). (2015)
49. Korus, P., Huang, J.: Multi-scale Analysis Strategies in PRNU-based Tampering Localization. IEEE Transactions on Information Forensics and Security **12**(4) (April 2017) 809–824
50. Al-Sanjary, O.I., Ahmed, A.A., Sulong, G.: Development of a video tampering dataset for forensic investigation. Forensic Science International **266** (September 2016) 565–572
51. Fiscus, J.G., Guan, H., Lee, Y., Yates, A.N., Delgado, A.P., Zhou, D.F., Kheyrkhah, T.N.: Nimble Challenge 2017 Evaluation. National Institute of Standards and Technology. (2017) `https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation`.
52. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675 (2016)
53. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 1., IEEE (2001) I–I
54. Chollet, F.: Xception: Deep Learning with Depthwise Separable Convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition. (2017)
55. Fridrich, J., Kodovský, J.: Rich Models for Steganalysis of Digital Images. IEEE Transactions on Information Forensics and Security **7**(3) (June 2012) 868–882

56. Cozzolino, D., Gragnaniello, D., L.Verdoliva: Image forgery detection through residual-based local descriptors and block-matching. In: IEEE International Conference on Image Processing. (October 2014) 5297–5301
57. Goljan, M., Fridrich, J.: CFA-aware features for steganalysis of color images. In: SPIE/IS&T Electronic Imaging. (2015)
58. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
59. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: A dataset for recognising faces across pose and age. arXiv preprint arXiv:1710.08092 (2017)
60. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680