VideoForensicsHQ: Detecting High-quality Manipulated Face Videos

Gereon Fox[1], Wentao Liu[1], Hyeongwoo Kim[1], Hans-Peter Seidel[1], Mohamed Elgharib[1], and Christian Theobalt[1]

Max Planck Institute for Informatics, Saarland Informatics Campus
{gfox,wliu,hyeongwoo.kim,hpseidel,elgharib,theobalt}@mpi-inf.mpg.de

Abstract. New approaches to synthesize and manipulate face videos at very high quality have paved the way for new applications in computer animation, virtual and augmented reality, or face video analysis. However, there are concerns that they may be used in a malicious way, e.g. to manipulate videos of public figures, politicians or reporters, to spread false information. The research community therefore developed techniques for automated detection of modified imagery, and assembled benchmark datasets showing manipulatons by state-of-the-art techniques. In this paper, we contribute to this initiative in two ways: First, we present a new audio-visual benchmark dataset. It shows some of the highest quality visual manipulations available today. Human observers find them significantly harder to identify as forged than videos from other benchmarks. Furthermore we propose new family of deep-learning-based fake detectors, demonstrating that existing detectors are not well-suited for detecting fakes of a quality as high as presented in our dataset. Our detectors examine spatial and temporal features. This allows them to outperform existing approaches both in terms of high detection accuracy and generalization to unseen fake generation methods and unseen identities.

1 Introduction

Accelerated by new combinations of model-based and deep learning-based approaches face video editing and synthesis approaches have reached unprecedented levels of visual realism [54]. On the one hand, methods enable high-quality reenactment [40,24,41], i.e. face expressions in video are manually modified or transferred from another video. Some also empower face swapping, i.e. replacing the face interior with a different face identity [18]; other popular implementations exist on GitHub [1,3]. These capabilities allow new applications in virtual reality, such as head-set removal for telepresence [43], or in visual effects creation, such as visual dubbing [19,38,23], and others. However, concerns arose that they could be misused to modify face videos of public figures or reporters in an unethical way or with the intent of misinformation. The research community has therefore developed techniques to detect such modifications and to verify authenticity of imagery, whether for generic content [7,4,47] or specifically for faces

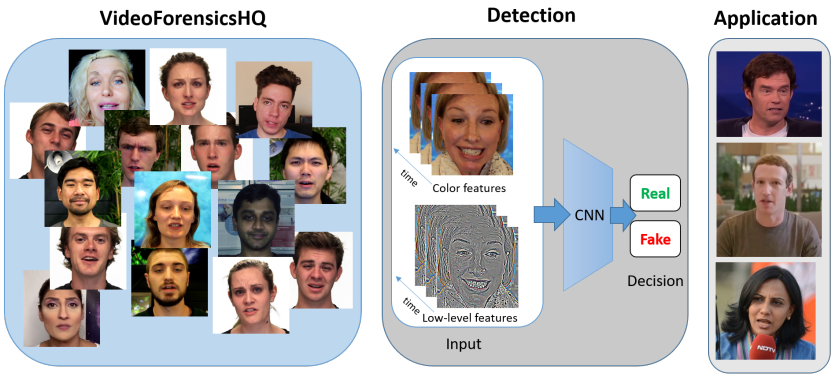

Fig. 1. We present the first large-scale dataset of high-quality face video fakes. Our dataset contains audio and is challenging for human detection. We also present a novel detector that examines a hybrid of features. Our detector outperforms existing techniques and produces good generalization to unseen fake generation methods.

[34,5,35,16]. Also, commendable efforts lead to larger benchmark datasets to develop and compare detection methods were released, both for images [53,22] and videos [26,22,34,15,13]. For instance, FaceForensics++ [34] contains internet videos modified by several face synthesis techniques [1,3,40,41,15].

In this paper, we contribute to the detection of manipulated face videos in two ways. First, although face image/video datasets cover various generation methods, we found even the best manipulations be easily unmasked by humans. We therefore propose VideoForensicsHQ, a benchmark dataset of high quality face video manipulations. It is one of the first face video manipulation benchmark sets that also contains audio and thus complements existing datsets along a new challenging dimension. VideoForensicsHQ shows manipulations at much higher video quality and resolution, and shows manipulations that are provably much harder to detect by humans than videos in existing data sets. To generate these results, we built on Deep Video Portraits [24]. A user-study shows that manipulations in VideoForensicsHQ are rated "real" at least 65.8% of the time, while the highest-quality reenactments in existing datasets [34] are rated "real" only 15% of the time. In total, our dataset contains 1,666,816 frames. It contains 1737 videos covering a wide variety of emotions. Second, we propose a new family of neural network-based detectors that examine low-level noise features, RGB color and temporal correlations. Comparisons see them outperform state-of-the-art detection methods on high quality manipulations. They also generalize well to unseen identities and to data generated by unseen methods.

2 Related Work

In the following we review face video reenactment and editing techniques, benchmark datasets, and manipulation detection techniques.

Face Reenactment and Editing Methods: Face reenactment techniques allow the control of face expressions in a target video [40,24,23,30,48,21,49,51]. Many extract expressions of target faces by fitting a parametric face model [20], and then re-synthesize the face with parameters copied from a source video [40,24,23,30]. Kim *et al.* [24] for the first time showed space-time coherent realistic global face pose and expression editing in videos by means of a GAN. Deferred Neural Rendering [41] can synthesize realistic face imagery using learned feature textures. Popular public implementations of face manipulation techniques [1,3] require large corpora of training data to achieve sufficient quality. We refer to [23] for a more comprehensive overview.

Face Manipulation Datasets: Several datasets of manipulated still images [53,22] or videos [26,22,34,13] exist. Zhou *et al.* [53] provide 2010 images generated by the manipulation approaches of [3,2]. Guan *et al.* [22] presented a dataset containing 50,000 manipulated images and 500 manipulated videos. Korshunov *et al.* [26] presented a dataset of 620 manipulated videos of 43 subjects. Roessler *et al.* [34]'s FaceForensics++ dataset contains 1,000 videos, each manipulated by 4 different techniques: DeepFakes [1], FaceSwap [3], Face2Face [40], and Neural Textures [41]. Their results show that many manipulation approaches, in particular DeepFakes [1] and FaceSwap [3], produce very noticeable artefacts. Also, none of these datasets provides audio with the videos. Google released the Deep Fake Detection Dataset [15]. It contains over 3000 manipulated videos (with unknown techniques) from 28 actors in various scenes. Many sequences exhibit notable visual artefacts; audio is not provided. Facebook recently released a large dataset of manipulated face videos [13]. The quality of the generated videos varies and faces are of low resolution (often much less than 299×299). The techniques used for manipulating the videos have not been disclosed.

Detection of Manipulated Visual Content: There is a lot of previous work on detecting computer-generated visual content. We summarize the most related methods and refer the reader to [44,45] for more details. Detection techniques can be coarsely segmented into approaches for faces [53,4,32,5,35,34,28,16,31] and approaches for generic content [17,11,12,8,33,27,52,7,50,10,47]. Generic techniques usually examine low-level features such as high-frequency components and noise. Fridrich *et al.* [17] introduced convolutional kernels designed for steganalysis that several works in the domain of fake detection built on. Cozzolino *et al.* [11] combined some of these kernels with an SVM-based classifier. In [12] the residual-based descriptors of [17] are formulated as a constrained convolutional neural network (CNN). Bayar *et al.* [8,7] suppress image content to focus on low-level patterns. Zhou *et al.* [52] use a two-stream network to detect edited content in images. One stream examines the image content while the other examines noise features. The work of [10] localizes edited regions of an image by examining the so-called "noiseprints" of camera models. Wang *et al.* [47] showed that a standard image classifier trained on one CNN generator can generalize well to data produced by unseen architectures. The classifier is trained on a large volume of data with careful pre- and post-processing plus data augmentation.

Results show high classification accuracy on a wide variety of unseen network architectures, including architectures for face generation.

Face-specific detection techniques can be classified into single-image-based [53,4,32,5,34,46,28,16,47] and multiple-image based [35,14,31] approaches. Zhou *et al.* [53] detect face swaps using a two stream network: One stream is trained to detect facial tampering artifacts while the other studies steganalysis features. Raghavendra *et al.* [32] detect whether a face is morphed from two different faces. Afchar *et al.* [4] proposed *MesoInc-4*, an inception-inspired [39] convolutional neural network with a small number of layers. *MesoInc-4* stacks the output of several convolutional layers with different kernel shapes, to learn at which level of granularity the input should be investigated. Rössler *et al.* [34] examined a variety of manipulation detection techniques [17,11,12,8,33,4] on the large FaceForensics++ dataset. Across different levels of compression, XceptionNet [9] turned out to be the most robust detector in this study. Durall *et al.* [16] classified the Discrete Fourirer Transform (DFT) of images using support vector machines, logistic regression and k-means. They reported very good results for high-resolution images. Other works have studied temporal correlations to achieve higher accuracy [6,35,6,5]. Agarwal *et al.* [5] learn a person-specific signature by extracting so-called "action units", that capture characteristic movements of known identities. An SVM is used to distinguish individuals. Sabir *et al.* [35] proposed a detector based on a recurrent neural network.

3 VideoForensicsHQ Dataset

Existing datasets for fake detection, such as FaceForensics++ [34], cover many state-of-the-art face synthesis techniques [1,3,40,41]. This breadth is valuable to benchmark and develop detection techniques, given a range of edit types (reenactment, face swap etc.). We provide VideoForensicsHQ, a new large scale and high-quality dataset of manipulated face video. VideoForensicsHQ complements existing datasets in several ways: First, most existing datasets only provide video, leaving out the speech audio channel. We argue that multi-modal analysis can open interesting questions for forgery detection in the near future. Second, community videos modified with state-of-the-art approaches frequently exhibit artefacts clearly visible to the human eye (see Fig. 2). As a result they are easily spotted by humans, which we could confirm in a user study (see Tab. 1).

One of the most discussed malevolent use cases is the modification of videos of public figures or politicians to disseminate wrong information. This threat, however, rests on the assumption that modifications cannot be detected by human observers. Most existing benchmark video sets do not reach this level of visual quality.

Providing benchmark data of high visual quality is key to model a more realistic threat scenario, anticipating improved capabilities of future face video synthesis algorithms. VideoForensicsHQ therefore contributes a large number of manipulated face videos with speech of previously unseen video resolution and visual modification quality (section 3.1). Our dataset contains self-reenactment

Fig. 2. Samples from existing face manipulation datasets. Most datasets contain noticeable visual artefacts. Neural textures [34] is of the highest visual quality in current datasets, and hence we focused our user-study on studying it. For more detail on the user-study please see the supplemental document.

manipulations, where the source and the target videos are the same. This design choice closely resembles the aforementioned scenario, where the speech of a world leader is modified, while maintaining his/her face identity, audio integrity and the remaining scene components, e.g., illumination, background and so on.

For the creation of our dataset, we rely on an extended version of Deep Video Portraits [24], allowing us to produce high visual quality for a sufficiently large number of videos (section 3.2). A user study shows that our modified videos are significantly harder to detect by human observers than videos in FaceForensics++ (see Tab. 1).

VideoForensicsHQ subsets

Subset	Fake frames	Real frames
group#1	60,058	119,992
group#2	74,765	190,259
group#3	192,150	1,029,592
total	32,6973	1,339,843

User study results

Source	Rated "fake"	Rated "real"
"Neural Textures" fakes [34]	85.7%	14.3%
VideoForensicsHQ fakes	34.2%	65.8%
Real videos	15.0%	85.0%

Tab 1. VideoForensicsHQ contains 326,973 frames of manipulated content produced by an enhanced implementation of Kim *et al.* [24]. Videos from VideoForensicsHQ are rated "real" 65.8% of the time, which compares favourably to FaceForensics++'s Neural Textures (the highest-quality method in that dataset). Our study also shows a baseline error of 15% where unmodified videos were incorrectly detected as "fake".

3.1 VideoForensicsHQ At-A-Glance

VideoForensicsHQ contains 1737 videos of speaking faces (44% male, 56% female), with 8 different emotions (Fig. 3), most of them of "HD" resolution. The

videos amount to 1,666,816 frames. Tab. 1 (left) shows the number of frames for the different groups of our data. Here, group#1 is mined from [23], group#2 from RAVDSS [29], and group#3 from YouTube. In group#1 each emotion is performed twice, and therefore we use one for training and other for testing. For the remaining groups we approximately use a $67\% - 33\%$ training-test split. In total, our dataset contains 326973 frames of fake content synthesized with the approach in section 3.2. This is comparable in size to the 306350 frames produced by the high-quality rendering approach of Thies *et al.* [41] in FaceForensics++.

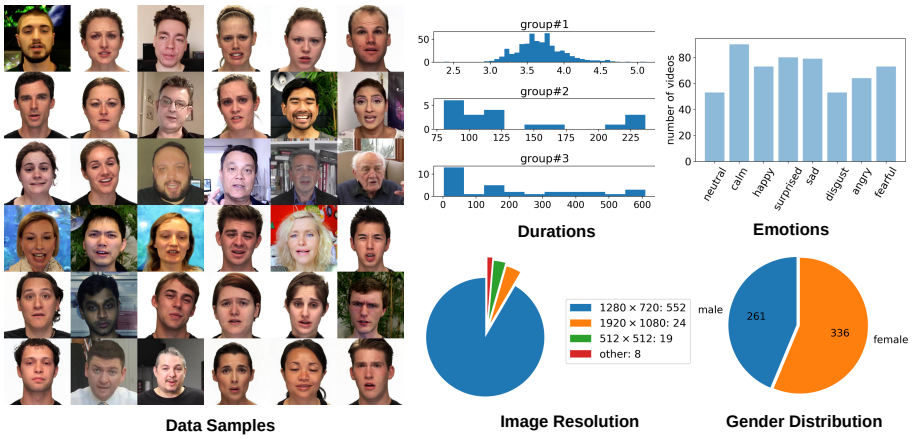

Fig. 3. Samples from our VideoForensicsHQ dataset and corresponding statistics. Our dataset offers a previously unseen level of visual quality, contains audio, and covers a variety of expressions, age, illuminations and backgrounds. Synthesized content of VideoForensicsHQ is much harder to detect by human observers than previous datasets.

3.2 Creation of VideoForensicsHQ

Mining Training and Test Segments Our face reenactments are produced on data mined from three input sources: 1) sequences from the work of Kim *et al.* [23], 2) the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [29] and 3) YouTube. We now describe how to identify suitable subsequences in this candidate set, and filter out unwanted scene or jump cuts.

Facial reenactment is more likely to exhibit artefacts on large, out-of-plane face poses. Hence, to ensure high visual quality of manipulated faces, we train our face reenactment approach on long video sub-segments of the original source videos, in the range of 5 - 10 minutes, showing near-frontal faces. To automatically find these, we run a facial landmark detector [37] on all frames of all source videos. This produces 66 landmark positions for every frame f_i, and one

confidence value in the range $[0, 1]$ for every landmark position. Based on the landmarks, we compute three metrics:

1. c_i: average confidence of the 66 landmarks for frame f_i
2. d_i: average difference between the landmark positions in f_i and the landmark positions in f_{i-1}, divided by the size of the face. Face size is taken as the average side length of the bounding box of the face landmarks.
3. mean and standard deviations of the c_i's and of the d_i's.

A frame is regarded as unsuitable for training if any of the following four cases are satisfied: 1) $c_i < 0.2$, 2) $d_i > 0.1$, 3) c_i is less than 0.6 and deviates from the confidence mean by more than 110% of the standard deviation (in negative direction), 4) d_i is greater than 0.025 and deviates from the displacement mean by more than 110% of the standard deviation (in positive direction). If none of these conditions apply, the frame is considered suitable for training and added to the current segment of suitable frames. Here, a segment is defined as a continuous set of suitable frames, with no scene cuts. We add the longest good segments to the self-reenactment training set of each identity until 5000 to 6000 frames are reached. From the remaining segments we create a test set for each identity. We mask out the background pixels that are not inside the convex hull of the face landmarks previously estimated. Finally, we crop and scale all frames to a resolution of 256×256, centered around the face.

Face Reenactment Algorithm To generate our results, we extend the Deep Video Portraits algorithm [24]. This method uses a rendering-to-real (R2R) translation network that maps a synthetic computer graphics rendering of a face to its photoreal equivalent.

The original Deep Video Portraits algorithm does not handle dynamic scene backgrounds. We therefore modify it to only process the facial region, as defined by the 66 facial landmarks of Saragih *et al.* [36].

Instead of a separate conditioning image for the eye-gaze we use only one input image to the translation network, with the eye-gaze overlayed on the synthetic rendering (see Fig. 4). We use the approach of Garrido *et al.* [20] to reconstruct parameters of a parametric 3D face model (PCA coefficients for identity geometry, PCA coefficient for face albedo, blendshape coefficients for expression, spherical harmonics coefficients for lighting, 3D head pose) for every frame of training video. Identity parameters are estimated based on the frame with highest landmark detection confidence c_i, before the remaining parameters of all frames are computed. The parameters obtained in this way are then rendered, to produce the conditioning input to the R2R network. We train this network per sequence, to make it learn the synthetic-to-real mapping. We train for 200 epochs and estimate the mean squared photometric error against ground truth on the validation set. The model with the smallest error ends up being used for synthesis.

Fig. 4 shows an overview of our synthetic-to-real translation. During inference we apply the learned translation model to the synthesized images with

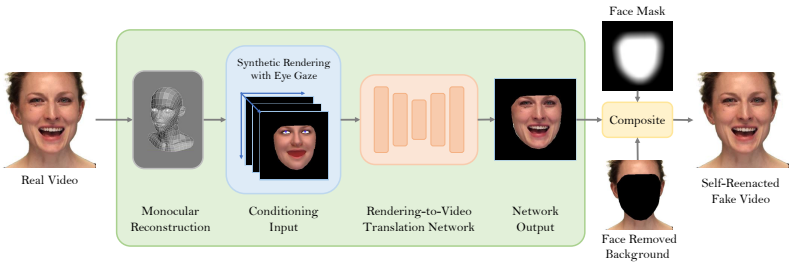

Fig. 4. We reenact the real input video using a translation network. The network takes a synthetic rendering of the face only, with the eye-gaze overlayed and with no background. It produces a photorealistic rendering which is then alpha blended into the background. This method is a special case of the Deep Video Protraits approach [24], where source and target sequence coincide.

eye-gaze overlay. This produces photo-realistic renderings that lack plausible background. We therefore composite them over the input frames, with a smooth alpha gradient around the background and the edges of the face (based on the landmarks). The results obtained in this way for group#3 are filtered manually, to ensure hight visual quality. The results of group#1 are filtered automatically, based thresholding the mean Euclidean distance between the real and reenacted videos. No filtering was necessary for group#2.

4 Detecting High-quality Manipulated Face Videos

We propose a family of detectors that examine multiple image features: low-level noise features, the original RGB color values and temporal correlations (see Fig. 6). We use XceptionNet [9] as the basis of our detectors, since it was reported as the most successful detector in [34]. XceptionNet consists of an entry flow $\mathrm{In}_{d\alpha\beta\gamma\delta\epsilon}$, a middle flow M, and an exit flow Out. The parameters α, β,γ, δ, and ϵ specify the number of features per convolutional layer (see supplemental material), while d specifies the number of input channels. To obtain the original XceptionNet, one can instantiate these building blocks as the sequence $C := \mathrm{In}_{3,32,64,128,256,728} \circ \mathrm{M}^8 \circ \mathrm{Out}$. C receives RGB images over the range $[0, 1]$ with mean $\frac{1}{2}$ as input. Its output are two scores, one for class "real" and one for "fake", for which we minimize cross-entropy loss. In the following, leading or trailing zeros in the indices of $\mathrm{In}_{d,\alpha,\beta,\gamma,\delta,\epsilon}$ disable the respective layers.

Since the fakes in our proposed dataset very rarely contain strong visual artifacts that an image classifier could easily pick up, we have derived a variant that classifies not frames themselves, but their spatially high-pass-filtered versions:

$$\mathrm{S} := \mathrm{In}_{3,32,64,128,256,728} \circ \mathrm{M}^0 \circ \mathrm{Out}$$

receives $\frac{1}{2} \cdot (F - g * F) + \frac{1}{2}$ as input, where $F \in \mathbb{R}^{299 \times 299 \times 3}$ is a video frame and g is a Gaussian kernel of size 5, with standard deviation $\sigma = 1.1$. Our combination

of C and S,

$$\mathrm{CS} := (\mathrm{In}_{3,32,64,128,256,364}, \mathrm{In}_{3,32,64,128,256,364}) \circ \mathrm{M}^2 \circ \mathrm{Out}$$

receives the same inputs as C and S and fuses the color and noise features just before entering M^2, where the combined receptive field of the convolutional kernels has size 17×17. We extend CS to

$$\mathrm{CST} := ((\mathrm{In}_{3,32,64,0,0,0}, \mathrm{In}_{3,8,8,0,0,0}) \circ \mathrm{In}_{0,0,72,128,256,512}, \mathrm{In}_{3,16,32,64,128,256}) \circ \mathrm{M}^1 \circ \mathrm{Out}$$

which in addition to color and spatial noise receives temporal noise as a third input (see Fig. 6).

Temporal noise is extracted as follows:

1. Spatial low-pass filtering with a Gaussian kernel of size 49 and deviation $\sigma = 7.7$. This suppresses high spatial frequencies that motion would turn into high temporal frequences.
2. Temporal high-pass filtering of the form
$$A_i := -\frac{1}{4}F_{i-1} + \frac{1}{2}F_i + -\frac{1}{4}F_{i+1}$$
extracts high temporal frequencies for each pixel, where F_i is the the frame to be classified.
3. Batch normalization to the range $[0; 1]$ with mean $\frac{1}{2}$.
4. Amplitudes below threshold t are dampened by computing
$$A'_i := \mathrm{thr}_t(A_i) - \mathrm{thr}_t(-A_i)$$
where
$$\mathrm{thr}_t(F) := \frac{1}{f} \cdot \left(\ln\left(1 + \exp(x)\right) + \frac{10}{1 + \exp(-x)} \right)$$
for $f := \frac{10}{t}$ and $x := f \cdot (F - t)$. We initialize t with $\frac{1}{40}$ and let the training process update it. See Fig. 6 (bottom-right) for the graph of thr_t.
5. $G_i := |A_i - A_{i-1}|$.
6. Temporal lowpass filtering ensures stability of the preprocessed signal over time:
$$T_i := \frac{1}{32}G_{i-2} + \frac{1}{8}G_{i-1} + \frac{3}{16}G_i + \frac{1}{8}G_{i+1} + \frac{1}{32}G_{i+2}$$

Fig. 5. A result of CST preprocessing, for a real (left) and a fake (right) frame [34]: Normalized color, spatial noise and temporal noise (from top to bottom). Training learned threshold $t = 0.0129$. Since dlib tracks the videos slightly differently, even unedited regions show marginal differences.

This preprocessing is supposed to emphasize rapid flickering of parts of the image, as is often observed in deep fake results, for example in Fig. 5.

C, S, CS and CST are all trained with stochastic gradient descent, with momentum 0.9 and a weight decay of 10^{-5}. We multiply the initial learning rate of 0.03 with a decay factor of $0.97^{0.1}$ after every epoch.

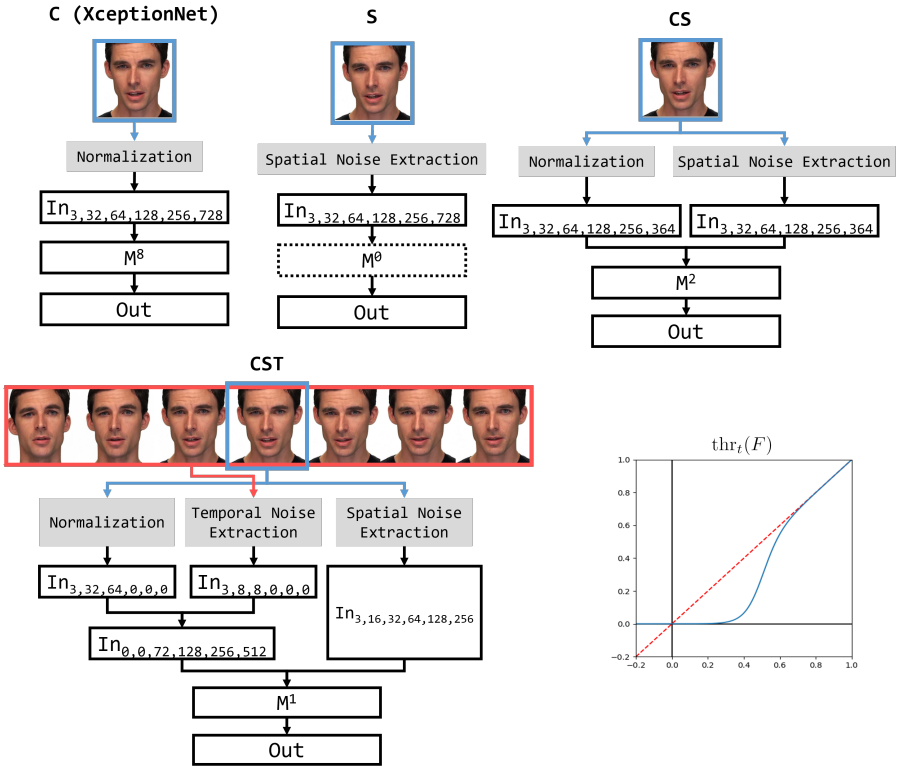

Fig. 6. Our detectors consist of building blocks extract from XceptionNet [9]. They examine spatial noise, color information and temporal noise. The latter involves learning a threshold r that determines the shape of the activation function thr_t.

5 Results

We perform a number of experiments to illustrate the high quality of our renderings in VideoForensicsHQ. Here, we provide evidence that state-of-the-art detectors struggle with the high level of photorealism of our dataset. Results show that by focusing on the low-level features of the input, our detectors manage to outperform existing methods. Furthermore, we demonstrate the generalization capabilities of our detectors. We examine the generalization to unseen rendering methods and also to unseen test identities in a few-shot learning setup. Results show that our detectors consistently outperform existing techniques.

State of the Art Techniques We compare our detectors to a number of related techniques. The detector C, published as "XceptionNet" [9], performs best in the FaceForensics++ benchmark [34]. It is the basis of our detectors. We evaluate *MesoInc-4* [4], *Bayar et al.* [7] and the approach of *Durall et al.* [16] as they show good results in analysing low-level features. For every training batch of *Durall et al.* [16] we optimize a new SVM model on the Fourier features. At

validation and test time, we average the predictions of all SVM models obtained in this way. To evaluate the generalization capabilities of our approach we compare to *Wang et al.* [47], a very recent classifier that generalizes well to unseen rendering methods. Due to this claim we do not update its weights during training, but only optimize a threshold on its singular output value, based on the ROC curve over the samples that were seen within one epoch of training. We perform this optimization for 5 epochs and average the 5 resulting thresholds.

Preprocessings and training All input data was preprocessed by computing face bounding boxes using *dlib* [25], smoothing their coordinates temporally and extracting constant-size square bounding boxes, scaled to resolution 299×299. We resample videos by linear interpolation to a framerate of 25fps. Frames for which no face bounding box could be found are omitted. For *MesoInc-4* we scale the resulting frames to 256×256, whereas for *Durall et al.*we compute a 209-dimensional feature vector as specified in [16]. All detectors were trained with batch size 24, except for *MesoInc-4* (512) , *Durall et al.* (512) and *Bayar et al.* (256). Except for *Wang et al.*, all methods are trained with a hard limit of 100 epochs. We stop training earlier if 5 epochs with a validation accuracy of more than 99% have been seen (not necessarily consecutively). The model with maximal validation accuracy is used at test time. To account for imbalances in the datasets, we randomly sample 10% of the training frames and 20% of the validation frames in every epoch. Sampling here means to first uniformly select a class ("real"/"fake"), then a subset (which is relevant for VideoForensicsHQ because it consists of three different groups), then a subject and then one of the sequences for this subject. Frames are sampled uniformly from sequences. Since *Durall et al.* is not designed for the amounts of data resulting from the aforementioned sampling rates we lower them to 0.5% training and 1% validation samples for this method. At test time, we evaluate *all* frames of the test set, but weigh per-frame predictions by the probability of a frame being sampled according to above sampling process.

5.1 Detecting highly photorealistic manipulations

To assess the difficulty of detecting the high quality fakes of our dataset, we trained our three detectors and the existing techniques on it. As can be seen in Tab. 2, all our detectors outperform the state of the art on VideoForensicsHQ. In addition to higher test accuracies, our detectors achieve peak validation accuracies much earlier than C/XceptionNet, from which they were derived. Furthermore, evaluation on FaceForensics++ confirms that our detectors perform well not only on our data. Overall this provides evidence for the claim that the fakes in VideoForensicsHQ are indeed more challenging to detect. Note that on VideoForensicsHQ, CS and CST exhibit lower test accuracies than the simpler S. We believe that this is due the design choice of dedicating considerably fewer channels to spatial noise at the end of the In module in CS and CST. This was necessary to limit the memory footprint during training.

Tab. 2 reports slightly lower accuracies for *MesoInc-4* and *Bayar et al.* on FaceForensics++ than [34], whereas the detectors based on XceptionNet, includ-

FaceForensics++

Arch.	Val.max.	Acc.	Acc. in [34]
C	99.47%@9	99.23%	99.26%
S	99.51%@6	99.38%	-
CS	99.53%@9	99.25%	-
CST	99.50%@6	99.35%	-
MesoInc-4	94.03%@50	92.44%	95.23%
Wang et al.	76.01%@2	75.55%	-
Durall et al.	57.77%@82	56.68%	-
Bayar et al.	96.97%@93	95.82%	98.74%

VideoForensicsHQ

Arch.	Val.max.	Acc.
C	91.18%@87	88.59%
S	99.40%@36	99.45%
CS	99.54%@45	97.12%
CST	98.15%@20	97.78%
MesoInc-4	68.29%@4	76.73%
Wang et al.	65.76%@5	56.44%
Durall et al.	64.27%@52	61.98%
Bayar et al.	81.51%@5	74.65%

Tab 2. Detection accuracies and validation accuracy maxima (including the epochs after which they occurred) of our detectors (S, CS, CST) and the existing approaches of *C*, *MesoInc-4*, *Wang et al.* and *Durall et al.*. All numbers are averages of two independent training runs. The accuracies we report for *MesoInc-4* and *Bayar et al.* on FaceForensics++ are slightly lower than in [34], because our training procedure has each detector see only 10% of the training data in each epoch, and because training is stopped after at most 100 epochs (see section 5).

ing the method by *Chollet et al.*, are clearly able to cope with our training setup (sampling only 10% of the training data per epoch, for at most 100 epochs). We interpret this as a strength of XceptionNet-based architectures, which require less training to achieve near-perfect accuracy on FaceForensics++.

5.2 Generalization across manipulation techniques

To assess the ability of our detectors to generalize to unseen manipulation techniques, we chose to train them on the FaceForensics++ subsets $FS \cup NT$ (FaceSwap + Neural Textures [41]) and $F2F \cup DF$ (Face2Face [40] and Deep Fakes) and then test them on the subset they were *not* trained on. See *Rössler et al* [34] for more information on the subsets. We chose these unions of subsets because each union thus contains a neural-network-based technique and a more traditional CG-based one. We consider VideoForensicsHQ not suitable for this experiment because it contains only one manipulation technique and because its characteristics (much higher resolution of face regions in most videos and a much smaller number of identities) are very different from those in FaceForensics++, which would spoil results when examining generalization to unseen *techniques*.

Tab. 3 shows that training our detectors on $FS \cup NT$ makes them generalize well to $F2F \cup DF$, where they outperform existing methods. We also observe that CS outperforms S and C by a large margin, suggesting that *combining* colour and spatial noise can help detection. The inverse direction, training on $F2F \cup DF$ and testing on FS sees low performance for all detectors, suggesting that this set contains artefacts not seen in $F2F \cup DF$. The subset NT seems to be easier to generalize to, with our detectors outperforming most existing techniques by a large margin. For this setting S outperforms CS, which suggests that color features learnt during training might not generalize well to NT. However, adding

temporal information seems to remedy this, as CST tops the accuracy of S and all other techniques.

A closer look at temporal analysis: Since the advantage of CST over CS when generalizing to NT could be due to the fact that CST dedicates less capacity to colour and noise information (in order to free GPU memory for temporal information) and thus is not able to overfit to the training data as much as CS, we also trained and tested the variant CST \T, which results from CST by replacing temporal noise extraction with a layer that produces a constant zero image. This variant has the same "advantage" of having lower capacities for colour and spatial noise than CS but does not receive temporal information. As Tab. 3 shows, this ablation leads to a significant drop in accuracy, making CST \T slightly worse than CS. The lack of capacity is thus not an advantage, which confirms that temporal information helps generalization.

Arch.	Val.max.	Acc.F2F	Acc.DF
C	99.35%@33	83.41%	93.16%
S	99.52%@10	98.54%	92.92%
CS	99.53%@8	99.53%	98.58%
CST	99.54%@9	99.09%	99.09%
CST \T	99.52%@13	99.36%	99.23%
MesoInc-4	98.96%@56	96.60%	62.45%
Wang et al.	71.17%@2	77.91%	80.03%
Durall et al.	58.62%@44	55.75%	55.45%
Bayar et al.	96.36%@98	62.62%	94.84%

Arch.	Val.max.	Acc.NT	Acc.FS
C	99.33%@11	58.55%	50.09%
S	99.56%@4	90.60%	54.82%
CS	99.64%@6	86.04%	52.17%
CST	99.62%@6	92.84%	55.61%
CST \T	99.59%@5	82.17%	53.16%
MesoInc-4	98.78%@57	71.85%	50.15%
Wang et al.	80.93%@4	84.40%	58.89%
Durall et al.	60.44%@69	53.63%	54.16%
Bayar et al.	98.78%@80	62.45%	50.05%

Tab 3. Validation maxima for the FaceForensics++ subsets $FS \cup NT$ and $F2F \cup DF$, together with the test accuracies for the subsets they were *not* trained on. All numbers are averages of three independent training runs.

5.3 Impact of training corpus size

To evaluate the importance of the number of identities in the training set, we have randomly sampled small training and validation sets from VideoForensicsHQ, with different numbers of identities. The models we trained on these subsets were tested on randomly selected VideoForensicsHQ subsets consisting of 15 identities each (different from the training and validation identities, of course). For each number of identities we have sampled 3 to 5 subsets and report the averages of the predictions from the models trained on these subsets in Fig. 7. We observe that our best models achieve close to 100% test accuracy already for training corpora of only 15 identities (training + validation), which is much fewer than the 45 identites in our dataset in total, providing evidence that our dataset is sufficient to generalize to unseen identities, and that our detectors do *not* overfit on the training identities.

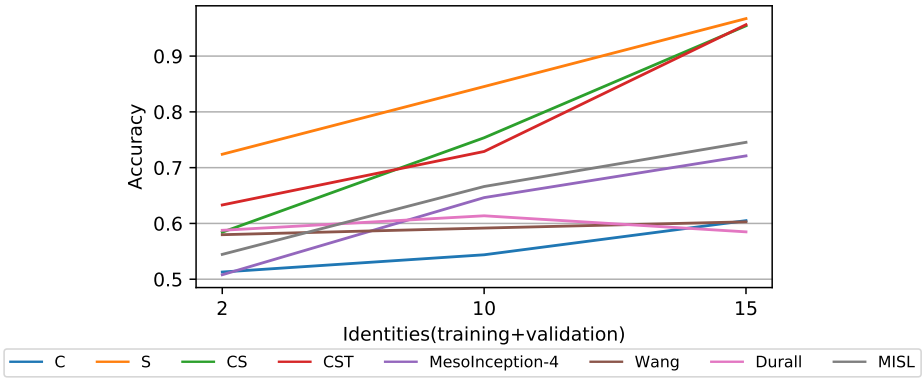

Fig. 7. Average test accuracies achieved by models that were trained on VideoForensicsHQ subsets containing different numbers of identities. Models were tested on a randomly selected subset of 15 VideoForensicsHQ identities. Training was stopped if 10 consecutive epochs without a new maximum for validation accuracy had been seen. The figure reports average of 3 - 5 runs for each training set size.

6 Conclusion

We presented a new approach to the important problem of detecting manipulated face videos. Our work is driven by the belief that the most dangerous form of facial manipulations are those which are challenging for *humans* to detect. Foreseeing that face reenactment techniques will be able to deliver a very high quality of photorealism in the near future, we make two main contributions to the important problem of forgery detection: First, we contribute a large-scale dataset of reals and fakes the high visual quality of which is confirmed by a user study. Second, we present a set of novel neural-network-based detectors that use spatial and temporal information to outperform the state of the art on our high-quality manipulated videos. In addition, our detectors generalize well to unseen rendering methods and unseen test identities. We plan to release our dataset and models to encourage research on this important problem. Future work can examine fusing audio and visual cues for better detection. We believe our dataset to be helpful for this endeavour.

References

1. Deepfakes. GitHub. `https://github.com/deepfakes/faceswap`
2. Faceswap. `https://itunes.apple.com/us/app/swapme-by-faciometrics`, this company has been acquired by Facebook and no longer available in AppStore
3. Faceswap. GitHub. `https://github.com/MarekKowalski/FaceSwap/`
4. Afchar, D., Nozick, V., Yamagishi, J., Echizen, I.: Mesonet: a compact facial video forgery detection network. International Workshop on Information Forensics and Security (WIFS) pp. 1–7 (2018)

5. Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., Li, H.: Protecting world leaders against deep fakes. In: Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (June 2019)
6. Amerini, I., Galteri, L., Caldelli, R., Del Bimbo, A.: Deepfake video detection through optical flow based cnn. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (Oct 2019)
7. Bayar, B., Stamm, M.C.: Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. IEEE Transactions on Information Forensics and Security **13**(11), 2691–2706 (Nov 2018). https://doi.org/10.1109/TIFS.2018.2825953
8. Bayar, B., Stamm, M.C.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: ACM Workshop on Information Hiding and Multimedia Security. pp. 5–10 (2016)
9. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1800–1807 (July 2017)
10. Cozzolino, D., Verdoliva, L.: Noiseprint: A cnn-based camera model fingerprint. IEEE Transactions on Information Forensics and Security **15**, 144–159 (2020). https://doi.org/10.1109/TIFS.2019.2916364
11. Cozzolino, D., Gragnaniello, D., Verdoliva, L.: Image forgery detection through residual-based local descriptors and block-matching. In: International Conference on Image Processing (ICIP). pp. 5297–5301 (Oct 2014)
12. 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. pp. 159–164 (2017)
13. Dolhansky, B., Howes, R., Pflaum, B., Baram, N., Ferrer, C.C.: The deepfake detection challenge (dfdc) preview dataset (2019)
14. Du, M., Pentyala, S., Li, Y., Hu, X.: Towards generalizable forgery detection with locality-aware autoencoder (2019)
15. Dufour, N., Gully, A.: Contributing data to deepfake detection research. Google Blog. `https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html`
16. Durall, R., Keuper, M., Pfreundt, F.J., Keuper, J.: Unmasking deepfakes with simple features (2019)
17. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security **7**(3), 868–882 (Jun 2012)
18. Garrido, P., Valgaerts, L., Rehmsen, O., Thormaehlen, T., Perez, P., Theobalt, C.: Automatic face reenactment. In: Computer Vision and Pattern Recognition (CVPR) (2014)
19. 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 (Proceedings of Eurographics) (2015)
20. Garrido, P., Zollhöfer, M., Casas, D., Valgaerts, L., Varanasi, K., Pérez, P., Theobalt, C.: Reconstruction of personalized 3d face rigs from monocular video. ACM Transactions on Graphics (TOG) **35**(3), 28 (2016)
21. Geng, J., Shao, T., Zheng, Y., Weng, Y., Zhou, K.: Warp-guided GANs for single-photo facial animation. ACM Transactions on Graphics (TOG) **37**(6), 231:1–231:12 (2018). https://doi.org/10.1145/3272127.3275043, `http://doi.acm.org/10.1145/3272127.3275043`

22. Guan, H., Kozak, M., Robertson, E., Lee, Y., Yates, A.N., Delgado, A., Zhou, D., Kheyrkhah, T., Smith, J., Fiscus, J.: Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In: Winter Applications of Computer Vision Workshops (WACVW). pp. 63–72 (Jan 2019)
23. Kim, H., Elgharib, M., Zollhöfer, M., Seidel, H.P., Beeler, T., Richardt, C., Theobalt, C.: Neural style-preserving visual dubbing. ACM Transactions on Graphics (TOG) (2019)
24. Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Niessner, M., Pérez, P., Richardt, C., Zollhöfer, M., Theobalt, C.: Deep Video Portraits. ACM Transactions on Graphics (TOG) (2018)
25. King, D.E.: Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research **10**, 1755–1758 (2009)
26. Korshunov, P., Marcel, S.: Deepfakes: a new threat to face recognition? Assessment and detection. arXiv **abs/1812.08685** (2018), http://arxiv.org/abs/1812.08685
27. Li, H., Huang, J.: Localization of deep inpainting using high-pass fully convolutional network. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8300–8309 (Oct 2019). https://doi.org/10.1109/ICCV.2019.00839
28. Li, Y., Lyu, S.: Exposing deepfake videos by detecting face warping artifacts. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2019)
29. Livingstone, S.R., Russo, F.A.: The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. In: PloS one (2015)
30. Nagano, K., Seo, J., Xing, J., Wei, L., Li, Z., Saito, S., Agarwal, A., Fursund, J., Li, H.: paGAN: Real-time avatars using dynamic textures. ACM Transactions on Graphics (TOG) **37**(6), 258:1–258:12 (2018). https://doi.org/10.1145/3272127.3275075, http://doi.acm.org/10.1145/3272127.3275075
31. Nguyen, H.H., Yamagishi, J., Echizen, I.: Capsule-forensics: Using capsule networks to detect forged images and videos. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2307–2311 (May 2019). https://doi.org/10.1109/ICASSP.2019.8682602
32. Raghavendra, R., Raja, K.B., Venkatesh, S., Busch, C.: Transferable deep-cnn features for detecting digital and print-scanned morphed face images. In: Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1822–1830 (July 2017)
33. Rahmouni, N., Nozick, V., Yamagishi, J., Echizen, I.: Distinguishing computer graphics from natural images using convolution neural networks. In: Workshop on Information Forensics and Security (WIFS). pp. 1–6 (Dec 2017)
34. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: Faceforensics++: Learning to detect manipulated facial images. arXiv (2019)
35. Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., Natarajan, P.: Recurrent convolutional strategies for face manipulation detection in videos. In: Workshop on Applications of Computer Vision and Pattern Recognition to Media Forensics with CVPR (2019)
36. Saragih, J.M., Lucey, S., Cohn, J.F.: Face alignment through subspace constrained mean-shifts. In: International Conference on Computer Vision (ICCV). pp. 1034–1041 (2009)
37. Saragih, J.M., Lucey, S., Cohn, J.F.: Deformable model fitting by regularized landmark mean-shift. IJCV **91**(2) (2011)

38. Suwajanakorn, S., Seitz, S.M., Kemelmacher-Shlizerman, I.: Synthesizing obama: Learning lip sync from audio. ACM Transactions on Graphics (TOG) **36**(4) (Jul 2017)
39. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI Conference on Artificial Intelligence (AAAI). pp. 4278–4284 (2017)
40. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In: Computer Vision and Pattern Recognition (CVPR) (2016)
41. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. ACM Transactions on Graphics (TOG) (2019)
42. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. ACM Transactions on Graphics 2019 (TOG) (2019)
43. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Niessner, M.: Facevr: Real-time gaze-aware facial reenactment in virtual reality. ACM Transactions on Graphics (TOG) **37**(2), 25:1–25:15 (Jun 2018). https://doi.org/10.1145/3182644, http://doi.acm.org/10.1145/3182644
44. Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., Ortega-Garcia, J.: Deepfakes and beyond: A survey of face manipulation and fake detection (2020)
45. Verdoliva, L.: Media forensics and deepfakes: an overview (2020)
46. Wang, S.Y., Wang, O., Owens, A., Zhang, R., Efros, A.A.: Detecting photoshopped faces by scripting photoshop. arXiv preprint arXiv:1906.05856 (2019)
47. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot...for now. In: CVPR (2020)
48. Wiles, O., Sophia Koepke, A., Zisserman, A.: X2face: A network for controlling face generation by using images, audio, and pose codes. In: European Conference on Computer Vision (ECCV). pp. 690–706 (2018)
49. Wu, W., Zhang, Y., Li, C., Qian, C., Loy, C.C.: Reenactgan: Learning to reenact faces via boundary transfer. In: European Conference on Computer Vision (ECCV). pp. 622–638 (2018)
50. Wu, Y., AbdAlmageed, W., Natarajan, P.: Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9535–9544 (June 2019). https://doi.org/10.1109/CVPR.2019.00977
51. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models (2019)
52. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Learning rich features for image manipulation detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1053–1061 (June 2018). https://doi.org/10.1109/CVPR.2018.00116
53. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Two-stream neural networks for tampered face detection. In: Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1831–1839 (July 2017)
54. Zollhöfer, M., Thies, J., Garrido, P., Bradley, D., Beeler, T., Pérez, P., Stamminger, M., Nießner, M., Theobalt, C.: State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications. Computer Graphics Forum (Eurographics State of the Art Reports) **37**(2) (2018)

A User-Study

We conducted a user study to asses how difficult it is for people to spot manipulations in VideoForensicsHQ compared to FaceForensics++ [34]. We randomly selected 13 manipulated videos VideoForensicsHQ and 13 manipulated videos from the "Neural Textures" subset of FaceForensics++ [34], created with the reenactment technique by Thies *et al.* [41]. Other approaches in FaceForensics++ produce manipulations with much more visible artefacts (see Figure 2 in the main manuscript). In addition, we randomly selected 6 unmodified videos from VideoForensicsHQ and 7 from FaceForensics++.

In total our study contains 39 videos, randomly shuffled for each user. For each video, we recorded the answer to the question "Does the video look real or fake?". Most participants were computer scientists, with little-to-no knowledge of face manipulation techniques. 61 subjects participated in the study. On average, modified videos from VideoForensicsHQ were rated real 65.8% of the time, and modified videos from FaceForensics++ were rated real only 14.3% of the time. It is important to note that unmodified videos were also rated as manipulated 15% of the time, which reflects a baseline error level in human detection performance. We also asked participants what made them flag a video as modified. Some of the most common responses were:

1. Various forms of visual artefacts, especially in the mouth interior
2. Non-natural eye movement
3. Body movements and hand gestures do not match the speech
4. Non-natural mouth-related movements e.g. lips are tight when they should not be, deforming/dislodging jaw and so on.
5. Incorrect audio-lip synchronization
6. A single glitch occurring over 2-3 seconds
7. The spoken language did not match the language of the on-screen text

B Impact of compression

We have evaluated the impact of compression on the test accuracy of all detectors on our dataset.

Our first experiments showed that applying the same compression to our data as was applied to FaceForensics++ by *Rössler et al.* [34] does not lead to the expected drop in detection accuracy. The reason is that the face regions in our dataset (average size 488×488 pixels) are much larger than in FaceForensics++(average size 258×258). Since compression happens *before* we detect square face crops and resize them to the detector input resolution of 299×299, compression artifacts hurt the quality of our higher-resolution video much less than that of *Rössler et al.* [34].

This is why we have resized all videos in our dataset to contain the same average face crop resolution as in FaceForensics++ and *then* compressed the result, before we applied face crop extraction. See Fig. 8 for example results. Tab. 4 reports the expected decrease in detector performance: Light compression

already leads to significant deterioration, with our detectors still outperforming previous methods. While S, though based exclusively on high-frequency spatial information suffers least under light compression, the picture changes as the compression level is increased: S is now performing worst among our detectors, which still outperform the baselines. However, they do so by a much smaller margin and with the exception of *MesoInc-4*, that is now on par with the simpler ones.

Light compression (CRF 23)

Arch.	Val.max.	Acc.
C	63.34%@23	71.16%
S	84.68%@13	83.58%
CS	84.34%@5	76.33%
CST	83.93%@3	78.65%
MesoInc-4	62.34%@3	68.27%
Wang et al.	57.61%@3	52.40%
Durall et al.	60.91%@15	58.51%
Bayar et al.	76.25%@6	69.99%

Strong compression (CRF 40)

Arch.	Val.max.	Acc.
C	60.41%@12	64.84%
S	65.48%@4	60.19%
CS	63.06%@5	67.53%
CST	61.10%@8	65.49%
MesoInc-4	56.85%@3	63.89%
Wang et al.	51.20%@3	47.88%
Durall et al.	57.70%@14	53.16%
Bayar et al.	64.31%@5	57.39%

Tab 4. Validation maxima and test accuracies of all evaluated detectors, trained on the same H.264 compression levels as in *Rössler et al.* [34], i.e. "light compression" (constant rate factor 23) and "strong compression" (constant rate factor 40). Training was stopped if validation accuracy did not see a new maximum for 10 consecutive epochs. All numbers are averages of three independent training runs. Since the videos in our dataset are of much higher resolution than in FaceForensics++, and since in addition the face regions are much larger relative to the video resolution, we have resized all our videos such that the average face crop resolution in each video over time is equal to the one found in FaceForensics++. Only after that we applied compression and then preprocessed the result as input for the evaluated detectors.

C Parametrization of XceptionNet

Our detectors are composed of the modules $\text{In}_{d\alpha\beta\gamma\delta\epsilon}$, M and Out, illustrated in Fig. 9.

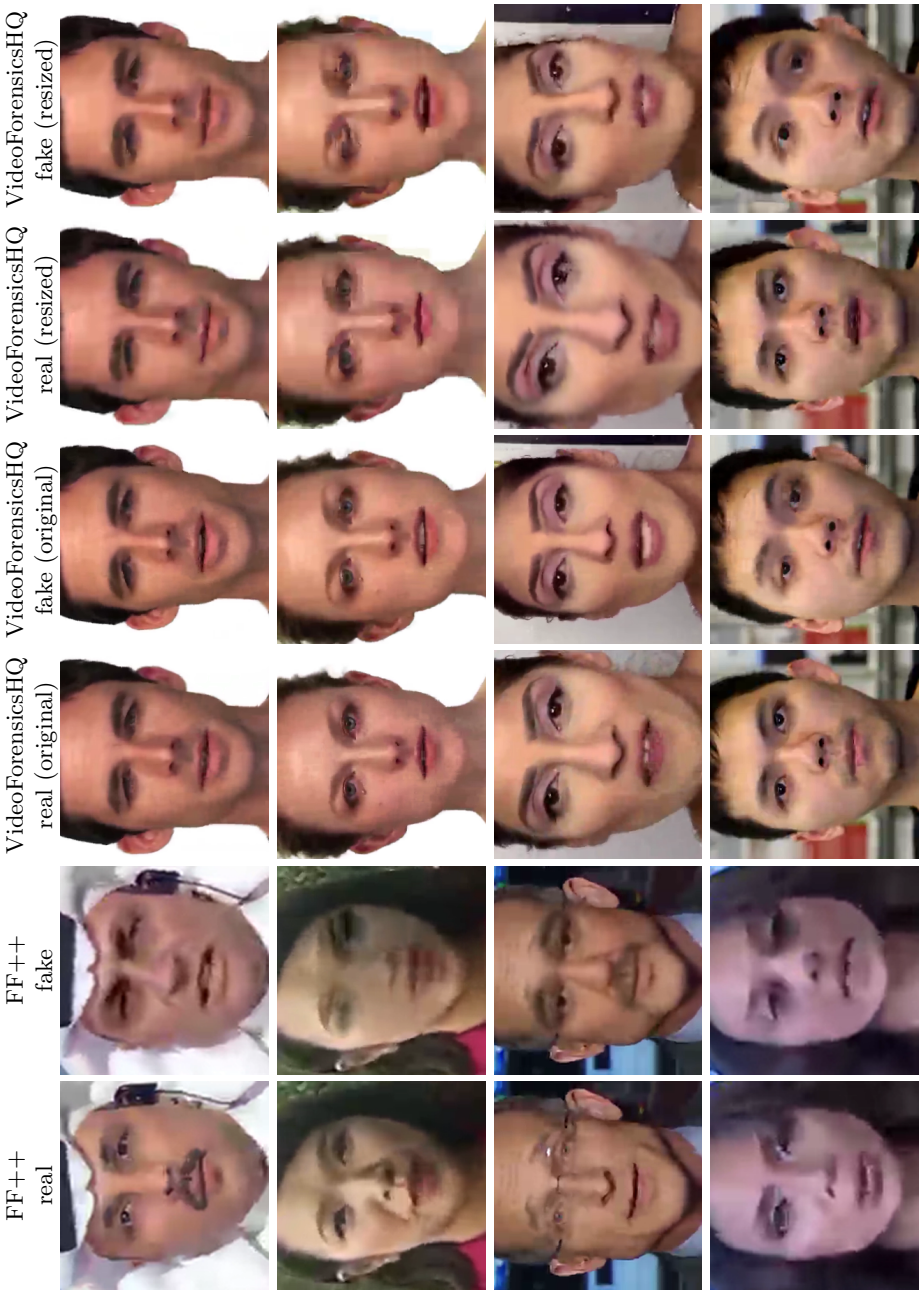

Fig. 8. A comparison of reals and fakes, all resulting from H.264 compression with constant rate factor 40, and then preprocessed as input to the detectors. If the VideoForensicsHQ videos are not resized before they are compressed, compression has much less impact on video quality and thus detection accuracy. These examples are better viewed on screen than on paper.

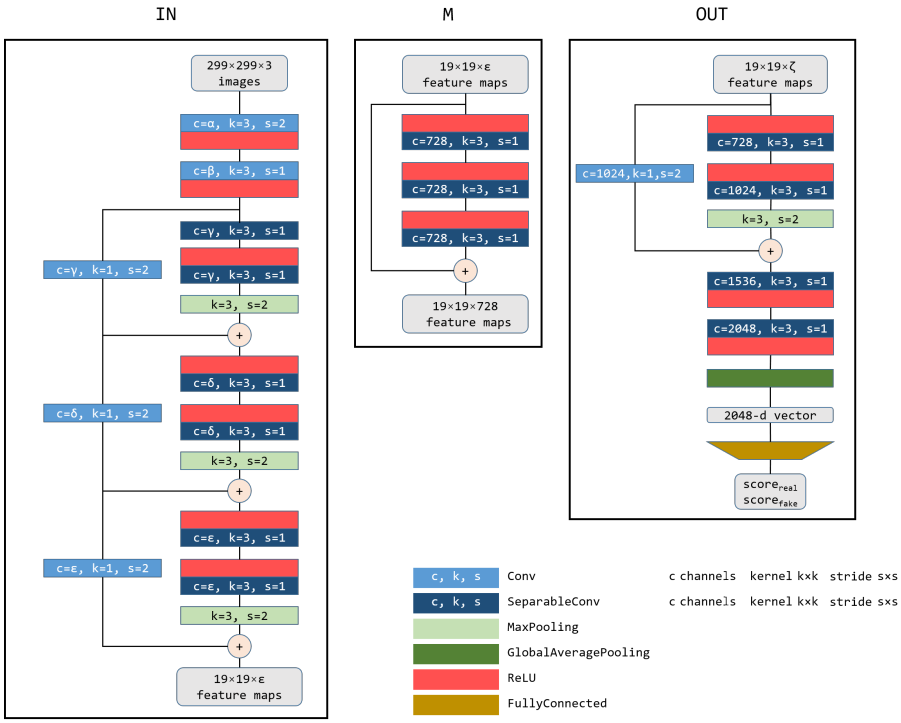

Fig. 9. The blocks out of which we composed our detectors are identical to the blocks of XceptionNet [9], except for the numbers of features treated in each layer. ϵ in M and ζ in Out are determined by the number of output feature in the preceding block.