DeepFake Detection Based on Discrepancies Between Faces and Their Context

Yuval Nirkin, Lior Wolf, Yosi Keller, and Tal Hassner

Abstract—We propose a method for detecting face swapping and other identity manipulations in single images. Face swapping methods, such as DeepFake, manipulate the face region, aiming to adjust the face to the appearance of its context, while leaving the context unchanged. We show that this modus operandi produces discrepancies between the two regions (e.g., Fig. 1). These discrepancies offer exploitable telltale signs of manipulation. Our approach involves two networks: (i) a face identification network that considers the face region bounded by a tight semantic segmentation, and (ii) a context recognition network that considers the face context (e.g., hair, ears, neck). We describe a method which uses the recognition signals from our two networks to detect such discrepancies, providing a complementary detection signal that improves conventional real versus fake classifiers commonly used for detecting fake images. Our method achieves state of the art results on the FaceForensics++ and Celeb-DF-v2 benchmarks for face manipulation detection, and even generalizes to detect fakes produced by unseen methods.

Index Terms—Image forensics, deep learning, deep fake, face swapping, fake image detection

1 Introduction

Photography is widely perceived as offering authentic evidence of actual events, including, in particular, the presence and actions of human subjects in images and videos. Although this perception is slowly shifting, contemporary technology allows far easier and more accessible manipulation of images than many realize. This gap represents a societal threat whenever manipulated media is released over social networks and consumed by a public that is ill-equipped to question its authenticity.

For instance, existing technology makes it easier for an actor to speak a given text, and then change her facial appearance and voice to imitate those of someone else. Alternatively, the face of a person captured in a crime-scene can be manipulated and replaced by another. Both of these examples are referred to as *face swapping*. A third scenario involves the reenactment of a person's face to change expression or lip motion (aka *face reenactment*). We note, however, that the third scenario differs from the first two, as it does not involve a change in identity.

Most contemporary approaches for detecting such manipulations relate to these three scenarios similarly: by training a classifier to distinguish between real and fake images or videos [8], [9], [10], [11], [12]. Recently, detection methods have been proposed that focus on liveliness and other specific authenticity signals such as heartbeat [13], [14] and specular highlights [15].

The Face X-ray method [16] focuses on the blending step, which is a common post-processing step for methods that manipulate faces in videos. This model detects the boundaries of the blending mask, which is then classified as real or fake. Focusing on a generic step in the manipulation pipeline, makes the approach better suited for unseen manipulation methods. Similar to Face X-ray we also focus on a common trait shared by most face swapping methods. While Face X-ray focuses on the seam between real and face content, we focus on the discrepancy in identities between the two.

Application-wise, swapping is of particular interest, as many of the existing face manipulation methods are designed for such identity modifying use cases. To this end we make two assumptions: (A1) Facial manipulation methods only manipulate the internal part of the face. (A2) The context of the face, which includes the head, neck, and hair regions outside the internal part of the face, provides a significant identity signal for the subject.

We verify assumption A2 in Section 3.2. Our findings are consistent with previous reports, showing that context alone indeed provides strong identity cues [17], [18]. To support assumption A1, Fig. 2 shows the affected regions of six different state of the art facial manipulation methods. Figs. 2a and 2b present two reenactment methods by Thies *et al.* [2], [3]. Both methods manipulate the regions corresponding to a 3D morphable model (3DMM) [19], [20], covering a facial region that contains part of the forehead at the top and most of the jaw on the bottom. Figs. 2c and 2d shows two deepfakes variants samples from the FaceForensics++ [5] and DFD [1] datasets, both affecting a square region in the middle of the face. Fig. 2e is another 3DMM-based face swapping method, affecting similar regions as the reenactment methods, excluding the internal part of the mouth (sample

- *Yuval Nirkin and Yosi Keller are with the Faculty of Engineering, Bar Ilan University, Ramat Gan 5290002, Israel. E-mail: {yuval.nirkin, yosi.keller}@gmail.com.*
- *Lior Wolf is with Tel Aviv University, Tel Aviv 6997801, Israel. E-mail: liorwolf@gmail.com.*
- *Tal Hassner is with Facebook AI, Menlo Park, CA 94025 USA. E-mail: talhassner@gmail.com.*

Manuscript received 20 Aug. 2020; revised 14 Apr. 2021; accepted 14 June 2021. Date of publication 29 June 2021; date of current version 9 Sept. 2022. (Corresponding author: Yuval Nirkin.) Recommended for acceptance by K. Sunkavalli. Digital Object Identifier no. 10.1109/TPAMI.2021.3093446

Fig. 1. *Detecting swapped faces by comparing faces and their context.* Two example fake (swapped) faces from DFD [1]. Left: The arm of the eyeglasses does not extend from face to context. Right: An apparent identity mismatch between face and context. We show how these and similar discrepancies can be used as powerful signals for automatic detection of swapped faces.

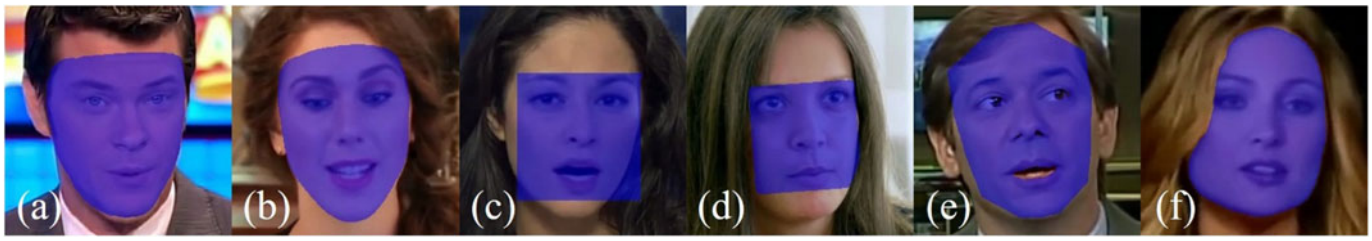

Fig. 2. *Affected regions of different manipulation methods.* (a) + (b) Face2Face [2] and NeuralTextures [3]; (c) + (d) Deepfake [4] variants of FaceForensics++ [5] and DFD [1]; (e) FaceSwap [6]; (f) FSGAN [7]. In all cases, faces are manipulated but their context is left unchanged.

obtained from previous work [5]). Fig. 2f is the output of FSGAN [7] which uses face segmentation to manipulate entire face regions.

We claim that it is no coincidence that all face manipulation methods we know of do not affect the entire head: While human faces have simple, easily modeled geometries, their context (neck, ears, hair, etc.) are highly irregular and therefore difficult to consistently reconstruct and manipulate, especially when considering the temporal constraints in video.

We present a novel signal for identifying fake images based on comparing the inner face region – the one that is directly manipulated – with its outer context, which is left unaltered by all face manipulation methods we are aware of. We do this by representing these two regions, faces and their context, with two separate identity vectors. The two vectors are obtained by training two separate face recognition networks: one trained for identifying a person based on the face region and the other trained to identify the person based on face context. We compare these two vectors, seeking identity-to-identify discrepancies.

Importantly, we *do not* assume prior knowledge of the identity of the person appearing in the image (source or target subject identities). Instead, given an image, we compare the representations for the one or two (unknown) identities, obtained from the face and its context using our two, specially trained networks.

The cue we derive using these two networks differs from those obtained by methods that search for artifacts caused by particular face manipulation techniques. Compared to other methods, our cue has three distinct advantages: First, our cue is based on the inherent design of face swap schemes and so is expected to hold even if future approaches produce photo-realistic, artifact-free results. Second, this cue generalizes well to different manipulation methods, whereas artifact detecting methods rely on algorithm-specific flaws. Finally, since the proposed cue is largely unrelated to artifact detection methods, it is complementary, and can thus be readily combined with such approaches to improve accuracy.

To summarize, we make the following contributions: (1) We propose a novel approach to identifying the results of face swapping methods. (2) Our method is based on a novel fake detection cue that compares two image-derived identity embeddings. (3) The proposed approach is shown to outperform existing state-of-the-art schemes when applied to FaceForensics++ [5], Celeb-DF-v2 [21], and DFDC [22]. (4) We show further results on two additional face swapping benchmarks, created using the FaceForensics++ data and additional swapping techniques, not included in FaceForensics++.

2 Related Work

Face Swapping Techniques. Semi- and fully-automatic face swapping methods were introduced nearly two decades ago [23], [24]. These early methods were proposed as a means for preserving privacy [24], [25], [26], recreation [27], and entertainment (e.g., [28], [29]); a far cry from some of their less appealing applications today in misinformation and fake news. Nearly all pre-deep learning approaches relied to some extent on 3D face representations, notably 3DMM [19], [20]. Some of the more recent examples of such methods are the Face2Face approach for expression transfer [2], face reenactement [30], expression manipulation [31], and face swapping methods [18].

Public awareness of face manipulation methods began following the introduction of deep learning–based swapping

and reenactment, particularly through the use of generative adversarial networks (GAN). A few notable examples of such techniques are GANimation [32], GANnotation [33], and others [34], [35], [36], [37]. Unlike earlier, 3D-based methods, GAN-based approaches are able to produce near photo-realistic results, not only in still photos, but also in videos. The quality of these results, along with the availability of public software, led to the use of what is now collectively known as *DeepFakes*, for undesirable applications, including porn and fake news.

More recently, FSGAN [7] showed convincing swapping results without requiring a dedicated training procedure for each source or target person, i.e., it is trained to replace any face with any other face. The FaceShifter state of the art swapping method [38] first merges the source identity with the features from the target face using multi-scale attention blocks, and then refines the result, handling occlusions in an unsupervised manner.

2.1 Detecting Manipulated Faces

Over the years, many proposed methods for detecting generic, copy-move and splicing manipulations in images and videos [39], [40], [41], [42]. Faces, however, received far less attention, likely because until recently, it was far harder to produce photo-realistic face manipulations.

The elevated threat posed by recent face manipulation methods is now being answered by increased efforts to develop automatic fake detection methods. Early methods for detecting manipulated visual media relied on hand-crafted features [11]. A more modern, deep learning–based implementation of this approach was recently described by Cozzolino *et al*. [10], followed by other deep learning–based methods, [8], [9], [12], [43], [44], [45], [46], [47], [48], as well as approaches utilizing multiple cues [42], [49], [50], [51], [52], [53], [54].

Sabir *et al*. [48] recently proposed a recurrent neural network which uses temporal cues to detect Deepfake manipulations in videos. Stehouwer *et al*. [55] applied an attention mechanism to intermediate feature maps of different backbone classifiers, to improve manipulated region detection accuracy. Songsri *et al*. [56] showed that using additional facial landmarks improves both detection and localization of Deepfakes. Finally, Nguyen *et al*. [51] suggested a fake detection architecture based on the capsule networks. Their work achieves results equivalent to previous methods, while utilizing significantly fewer parameters.

2.2 Benchmarking Face Manipulation

A number of recent efforts try to provide the research community with standard, high quality, fake detection benchmarks. These efforts include FaceForensics [47], DeepFake-TIMIT [57], Celeb-DF [21], VTD dataset [58], FaceForensics++ challenge [5], and the DFD dataset [1]. Several industry research labs have also recently contributed to these efforts, leading to the announcement of the DeepFake Detection Challenge (DFDC) [22].

These benchmarks represent multiple manipulation techniques – not just face swapping. By using a single (or few) synthesis methods, biases can be inadvertently introduced into these challenges: artifacts that are unique to a particular fake generation method, or to the use of particular training data. These sets, therefore, include media generated with a variety of synthesis methods. Our approach is designed to be invariant to such incidental biases: Rather than seeking particular artifacts, we consider a perceptual effect shared by swapping techniques in general and show that our method can detect fakes produced by previously unseen face manipulation techniques.

3 Recognition of Faces and Their Context

We describe the two complementary face recognition networks used to obtain identity cues for the face and its context. We further explain how we use these two networks in our proposed fake detection method. Deep neural networks are extensively used for face identification, and we focus on the contributions of two very specific facial regions, dictated by the desired application: the segmented face and its surrounding context.

3.1 Detecting and Segmenting Faces

We begin by applying the dual shot face detector (DSFD) [59]. We then increase detected bounding box sizes by 20 percent, relative to their height, to expose more of the context around the face, as DSFD is trained to return tight facial bounding boxes. Face crops are then resized to 299×299 pixels; the input resolution of the Xception architecture [60] which we use for our face/context cues (Section 3.2).

To determine which parts of the crop are processed by the face network and which by the context network, we segment the crop into foreground (face) and background (context) using a face segmentation network. The exact architecture and training details for the segmentation network are provided in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3093446. Given the cropped face I and its corresponding face segmentation mask S, we generate image I_f and its complementary image I_c, representing the face and its context, respectively.

3.2 Recognition Networks

Recognition Network Architecture. Our networks are based on the Xception architecture [60] following its success in detecting other DeepFake cues [5]. We train the network using a vanilla cross entropy loss, although other loss functions could presumably also be used. Xception is based on the Inception architecture [61] but with Inception modules replaced with depth-wise separable convolutions. As far as we know, it was never used for face recognition.

In our implementation, the Xception network consists of a strided convolution block, followed by twelve depth-wise separable convolutions blocks with residual connections, except for the last one. The network is terminated by two depth-wise separable convolutions, a pooling operation and a fully connected layer.

We train two identification networks: E_f which maps an image of size 299×299 containing pixels from the face region to a vector of pseudo-probabilities associated with the dataset faces, and, similarly, network E_c maps the remaining pixels from the detection bounding box (the context) to a vector of pseudo-probabilities of the same classes.

TABLE 1
Face Recognition Accuracy on VGGFace2

Method	Train set	Validation set
Context	99.90	87.06
Face	99.89	95.10
Entire region	**99.98**	**96.98**

Results reported for three face identification Xception networks, each applied to a different part of the face. As expected, the entire region, containing both face and context, is the most accurate. Even context alone, however, provides a strong cue for identification, as previously observed by others [17], [18].

We train both E_f and E_c on images from the standard, publicly available VGGFace2 dataset [62]. VGGFace2 contains 9,131 subjects from which we filtered images with a resolution lower than 128×128, resulting in 8,631 identities. The output of these two networks is, therefore, in $\mathbb{R}^{8,631}$.

Validating Recognition Capabilities. To validate and compare the recognition accuracy of these networks, we test their performance on both the VGGFace2 [62] test set and the test set of the Labeled Faces in the Wild (LFW) [63] benchmark (no additional training or fine tuning was applied to the networks before being tested on LFW images).

Unsurprisingly, addressing the internal appearance of the face, network E_f outperforms E_c in term of accuracy, though both accuracies are high. These results are evident from Table 1 for VGGFace2 and Fig. 3 for LFW. We note that the accuracy demonstrated by E_c – its ability to recognize faces despite only seeing the context – is unsurprising: similar results were reported by others, showing that faces can be recognized, even when only their context is visible [17], [18].

Importantly, Fig. 3b shows that the representations typically used for face recognition – the activations of the penultimate layer of the face recognition network, do not match well for the same person, since the two networks were trained independently. When combining the responses from these two networks, we, therefore, use their final output: the per-subject pseudo-probabilities (Section 4.1).

4 Fake Detection Using Faces versus Context

We illustrate our proposed fake detection approach in Fig. 4. Our method combines multiple Xception networks: The recognition networks, E_f and E_c, described in Section 3, a binary Xception net, E_s, trained to distinguish between real and manipulated images by face swapping methods, and another, *optional*, binary Xception net, E_r (not shown in Fig. 4), which we train to differentiate real images from those manipulated by face reenactment methods. We next describe these components in detail.

4.1 Face Discrepancy Component

We train the face discrepancy network to predict whether a face and its context share the same identity. It uses the output of the two recognition networks, E_f and E_c, described in Section 3. We pre-train these two networks and do not change their weights after they are combined, in order to ensure that the identity cues remain the dominant ones. In Section 5.3 we show that training with the recognition network's weights unfrozen leads to a reduced accuracy when generalizing to unseen methods. We process the face and context images, I_f and I_c, with two separate identity classifiers, E_f and E_c, respectively, to compute a discrepancy feature vector v_d

$$v_d = E_f(I_f) - E_c(I_c) = v_f - v_c. \tag{1}$$

4.2 Manipulation Specific Networks

Previous approaches trained classifiers to distinguish between real and fake faces, without considering the particular manipulation applied to the faces – swapping or reenactment. These two manipulations types differ significantly: Swapping manipulates the identity of the face, whereas reenactment manipulates facial pose and expression. While the latter is not the focus of our work, it is required by the

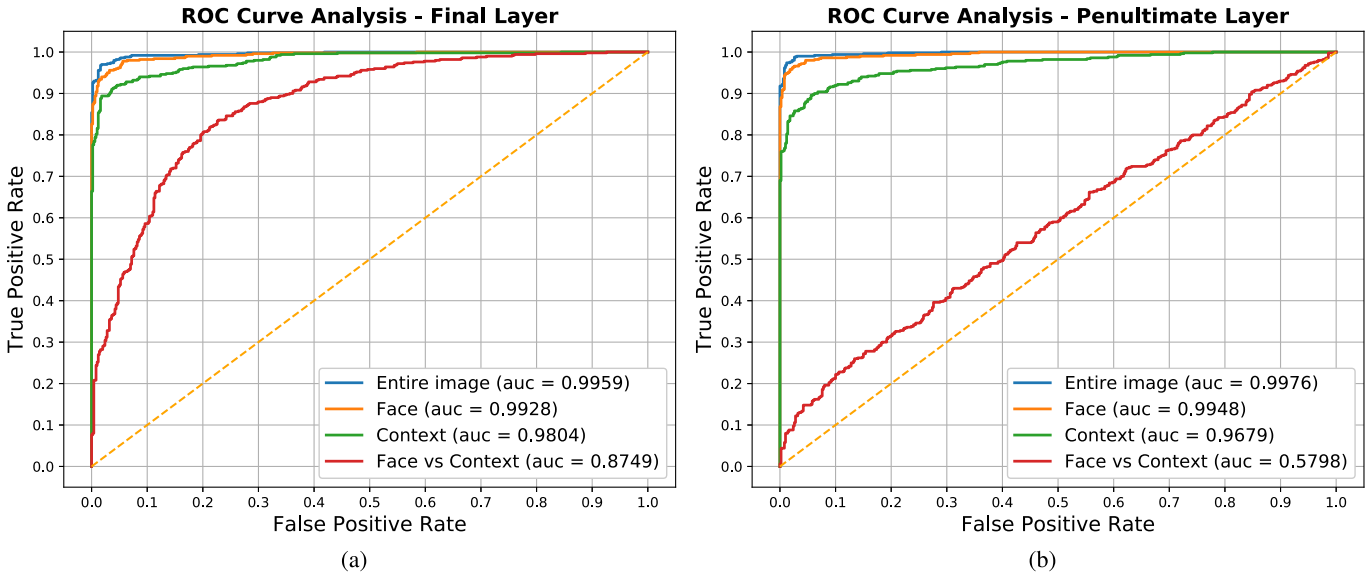

Fig. 3. *LFW verification accuracy for identification networks trained on different face regions.* (a) Results obtained by representing faces with the final layers of the Xception architectures. (b) Faces represented using the activations of the penultimate layers of Xception. In the latter case, face versus context do not match well for the same person, since the two networks were trained independently. Our approach, therefore, uses the final layers of the networks, representing subject pseudo-probabilities, when comparing the two (top).

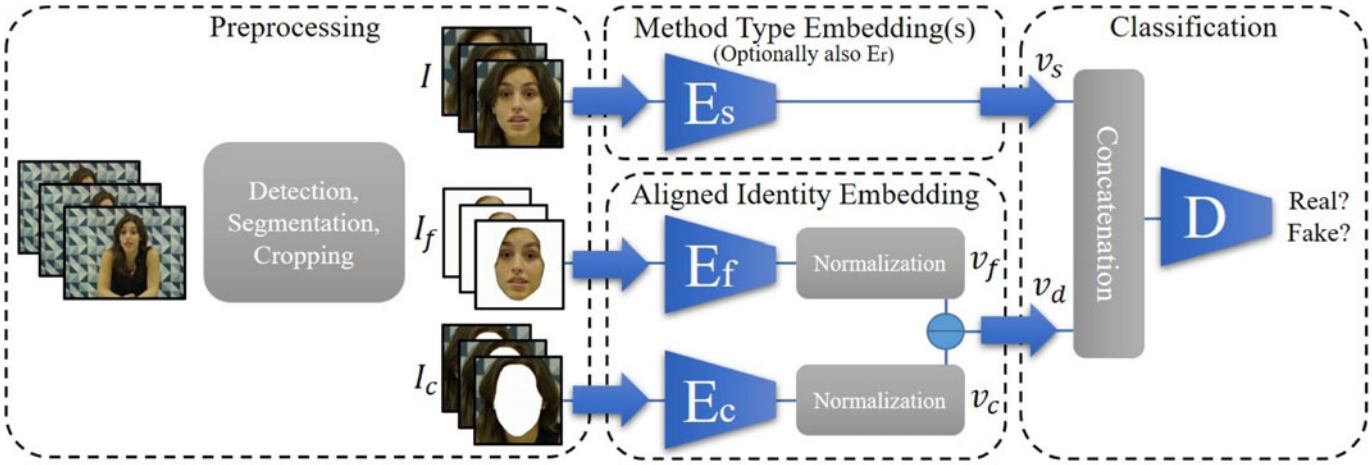

Fig. 4. *Method overview.* Following initial preprocessing, we obtain regions for the face, I_f, and its context, I_c. The two are processed by the face identification networks, E_f and E_c, respectively. A separate network, E_s, considers the input image, I, seeking apparent swapping artifacts to decide if it is a face swapping result. The pseudo-probability vectors of the two face identification networks are subtracted and, jointly with the representations obtained from the method type network, E_s, are passed to the final classifier, D.

FaceForensics++ benchmark used in our tests (Section 5.2). Our approach, therefore, includes also a component for detecting face reenactment.

Specifically, we decouple swapping and reenactment by training a separate, dedicated classifier for each: Network E_s is trained to detect swapping artifacts and network E_r (not shown in Fig. 4) is trained to detect reenactment. We use Xception networks, similar to those described in Section 3.2 for recognition, and train these networks to classify genuine versus manipulated. Our training process first pre-trains both networks on examples of their particular manipulation versus pristine images. Our reenactment network, E_r, is used in cases where the task is to detect both face swapping and face reenactment methods. Otherwise, we use a three network solution, where E_r is omitted.

4.3 Combining All Detection Cues

We chose the simplest method for combining the various signals: concatenating the three vectors v_d, v_s and v_r, where $v_d \in \mathbb{R}^{8,631}$ is defined in Eq. (1), and $v_s = E_s^p(I)$ and $v_r = E_r^p(I)$, both in $\mathbb{R}^{2,048}$, denote the activations of the penultimate layers of the binary E_s and E_r, respectively.

The concatenated vector is passed to classifier D, which outputs a real versus fake binary signal, trained using a logistic loss function. The classifier D consists of an initial linear layer, followed by batch normalization, ReLU, and a final linear layer.

4.4 Training

We first pre-train the four classifiers, E_s, E_r, E_f, and E_c, each on its own task. We train network E_s on the subset of videos in FaceForensics++ [5] consisting of pristine videos and videos manipulated by the face swapping methods: FaceSwap and Deepfakes. Network E_r is trained on the face reenactment methods: Face2Face and NeuralTextures. Note that we only use the compressed versions of these videos for training, with C23 (HQ) and C40 (LQ) compressions. We chose not to use the raw videos for training because there is little difference between them and the C23 compressed videos. The FaceForensics++ benchmark used to test our method does contain all three versions. The training process applied to E_f and E_c is detailed in Section 3.

Once the four networks are trained, we freeze the weights of E_f and E_c, and train the final classification network, D, using the three output vectors (v_s, v_r, v_d), while only fine-tuning E_r and E_s. The final training is done on the same split of the FaceForensics++ videos. For more technical details, please see supplemental, available online.

4.5 Inference on Full Images

During inference, we often process images containing multiple faces. In such cases, we only classify detected faces having a height larger than 64 pixels, and discard the rest as background faces. The only exceptions are images where the largest face does not comply with this criterion, in which case we process the largest detected face.

We further remove false detections by applying a threshold on the number of face pixels in the face segmentation mask, S, for each detection. We start with a threshold of 15 percent of the face pixels, relative to the number of pixels in the cropped region. If this step filters-out all our detections, we reduce the threshold by half. If none of the images pass the 7.5 percent threshold, we simply consider the one face patch with the maximal number of detected pixels.

Finally, we apply the compound network, including E_m, E_f, E_c, and D, to the remaining face patches (one or more) and obtain one score per face patch as the output of D. We take the minimal output of these scores – the face patch predicted as most likely to be fake – in cases where only a single face is manipulated.

5 Experimental Results

We evaluated our proposed scheme using three recent, challenging benchmarks: FaceForensics++ [5], DFDC [22], and Celeb-DF-v2 [21]. In order to evaluate our method using additional face swapping techniques and test its generalization abilities, we further create our own test set, using two more swapping methods.

TABLE 2
Face Swap Detection Results

Methods	FF-DF	Celeb-DF-v2
Two-stream [54]	70.1	53.8
Meso4 [8]	84.7	54.8
MesoInception4 [8]	83.0	53.6
HeadPose [53]	47.3	54.6
FWA [45]	80.1	56.9
DSP-FWA [45]	93.0	64.0
VA-MLP [49]	66.4	55.0
VA-LogReg [49]	78.0	55.1
XceptionNet-raw [5]	**99.7**	48.2
XceptionNet-c23 [5]	**99.7**	65.3
XceptionNet-c40 [5]	95.5	65.5
Multi-task [64]	76.3	54.3
Capsule [51]	96.6	57.5
Ours	**99.7**	**66.0**

Comparison of our approach and leading state of the art methods on two benchmarks using frame-level AUC (%).

TABLE 3
FaceForensics++ Image Benchmark Results

Methods	DF	F2F	FS	NT	Pristine	Total
Steg. Features [11]	73.6	73.7	68.9	63.3	34.0	51.8
Cozzolino *et al.* [10]	85.4	67.8	73.7	78.0	34.4	55.2
Rahmouni *et al.* [12]	85.4	64.2	56.3	60.0	50.0	58.1
Bayar and Stamm [9]	84.5	73.7	82.5	70.6	46.2	61.6
MesoNet [8]	87.2	56.2	61.1	40.6	72.6	66.0
Xception [5]	96.3	86.8	90.3	80.7	52.4	71.0
Ours	94.5	80.3	84.5	74.0	67.6	**75.0**

Columns are: DeepFakes *(DF),* Face2Face *(F2F),* FaceSwap *(FS),* NeuralTextures *(NT), and* Pristine *categories. It is hard to compare specific columns, since there is a threshold-based trade-off between real and fake. These columns are therefore provided only for completeness. Our method leads in the Total score, which is the meaningful metric for this benchmark.*

5.1 Face Swapping Detection Experiments

We use the following three datasets containing only face swapping examples:

FF-DF. FF-DF [21] is a subset of the FaceForensics++ benchmark [5], which includes only faces swapped using the Deepfakes method [4]. These tests therefore include 1,000 videos from the *pristine* subset and 1,000 videos from the *Deepfakes* subset (the full FaceForensics++ is described in Section 5.2).

DFDC. The recently announced, industry-backed, preview of the DFDC benchmark [22] offers a total of 5,244 videos of 66 actors: 4,464 training videos and 780 test videos, 1,131 of them are real videos and 4,113 are fakes generated by two different, unknown, face swapping methods.

Celeb-DF-v2. Another recent dataset containing 590 real videos and 5,639 DeepFake videos of 59 celebrities [21]. This set is especially challenging as most state of the art methods tested on this set report near-chance accuracies.

Training and Evaluation. In these tests, we do not use our reenactment network, E_r. We train on FaceForensics++, as described in Section 3. Results for all baseline methods were previously reported [21]. These methods were trained mainly on FaceForensics++, sometimes with additional self collected data. None of these methods was trained on DFDC or Celeb-DF-v2 and so these experiments also compare the generalization of the different methods.

All methods were compared using the area under the curve (AUC), at the frame level, on all frames in which faces were detected. This metric is very convenient for comparing methods that output per-frame classification as there is no need to set thresholds.

Face Swap Detection Results. We report our results in Table 2. Our method achieves the best AUC scores on all the benchmarks. On FaceForensics's DeepFakes subset [5] our method achieves similar results as the current state of the art, this is due to the accuracy being saturated. On the more challenging Celeb-DF-v2 benchmark, small improvements on the AUC scores are significant. Note also that the results reported for our method on Celeb-DF-v2 testify to its improved generalization abilities compared to the baseline methods.

5.2 Experiments on FaceForensics++

The full FaceForensics++ dataset [5] contains 1,000 videos obtained from the web, from which 1,000 video pairs were randomly selected and used to generate additional 1,000 manipulated videos representing four face manipulation schemes. Two of these methods perform face swapping: a 3D-based face swapping method [6] using a traditional graphics pipeline and blending, and a GAN-based method [4], trained using the images of pairs of subjects to compute a mapping between them. Two additional methods perform face reenactment: Face2Face [2], a 3DMM-based method that manipulates facial expressions by changing the expression-coefficients estimated for the face, and NeuralTextures [3] which learns a face neural texture from a video and uses it to realistically render a 3D reconstructed face model.

Results on FaceForensics++ Image Benchmark. In this benchmark the results are calculated on a private server by uploading binary predictions. It is therefore required to select a threshold for the model's prediction scores, which we selected by optimizing on the validation set. Table 3 shows that our total accuracy outperforms all previous methods by a large margin. Importantly, the accuracy in each of the different categories, on its own, is not a direct indication of detection performance, since there is a threshold-dependent trade-off between the accuracy on real and fake images. These results hint at the relative detection difficulty of each class and are provided for completeness.

5.3 Ablation Study and Generalization Experiment

Face manipulation methods sometimes leave behind artifacts, possibly imperceptible, that can be leveraged for detection. Different manipulation methods, however, can produce different artifacts, as shown in Fig. 5. There is, therefore, no guarantee that a fake detection method would perform well when presented with fakes generated by unseen schemes which do not leave such known, recognizable artifacts. We next verify the accuracy of our proposed scheme in detecting fakes produced by methods that were not part of its training set.

Fig. 5. *Extending FaceForensics++ with unseen methods.* Examples shown for the same source / target face pair, using the 3D-based methods, FaceSwap [6] and Nirkin *et al.* [18], and the GAN-based methods, Deepfakes [4] and FSGAN [7]. Despite using the same image pairs in all four examples, the results are different, each exhibiting its own artifacts.

We conduct these tests by extending the FaceForensics++ set, applying two additional face swapping methods to its videos: (1) FSGAN [7] and (2) Nirkin *et al.* [18], a 3D-based face swapping method that uses single image 3D face reconstruction and segmentation, both have publicly available implementations. Examples of the four face swapping methods, using the same source and target, can be seen in Fig. 5. Each method generates face swaps with distinct artifacts, with the exception of FSGAN, which produces images with fewer apparent artifacts.

The extended version of the benchmark follows the pair selections prescribed by the original FaceForensics++ dataset. Because Nirkin *et al.* [18] was designed for image-to-image face swapping, for each frame in the target video we select its closest frame in the source video, in terms of estimated head pose.

In all our generalization experiments, we train the variants of our method and its XceptionNet baseline on the pristine and face swapping manipulations, using the official training and validation subsets of FaceForensics++. In these experiments, we do not use the reenactment detection network E_r.

5.3.1 Generalization and Ablation Results

We studied the effect of our face versus context discrepancy approach by comparing it to a naive classifier. Three additional variants of our method were also considered: (i) a version where all classifiers are frozen in the training process, (ii) an end-to-end version of our method, where all the classifiers are unfrozen in the training process, and finally, (iii) a variant where instead of subtracting v_f and v_c, we concatenate them.

We report our generalization results in Table 4 (ROC curves provided in Fig. 6). For results appearing at the top of Table 4, we fix the thresholds for XceptionNet and our method at zero. In the bottom of Table 4 we optimize both thresholds on the test set. The threshold of the face identity difference in the first experiment is optimized using the VGGFace2 test set.

Our results show that our method significantly outperforms the baseline on both unseen methods. The performance gap is greater on FSGAN generated faces, where artifacts are more rare. Artifacts produced by the 3DMM-based method are more similar to the ones we encounter in other methods, and so the gap is smaller.

As evident from the ROC curves in Fig. 6, the frozen version of our method, in which the method specific classifier is not given the option to adjust to the identity signal, is the worst performing variant. The end-to-end version of our method is also less able to generalize. This result is due to the end-to-end training process sullying the face and context classifiers roles for extracting aligned identity representations. The concatenation variant performed slightly worse than our method. This could be a result of the increase in the capacity of D.

Finally, note that the face discrepancy signal by itself is not competitive with networks trained to detect fakes. However, it is indicative of fake videos and its contribution to the overall method is seen by comparing our method with the baseline XceptionNet.

5.3.2 Image Laundering Ablation

We demonstrate our method's generalization performance under different image laundering attacks, on three face swapping methods, from older to newer: 3D-based swap [18], FSGAN [7], and FaceShifter [38]. The image laundering operations include JPEG compressions of 25, 50, and 75 percent, where higher percentage means stronger

TABLE 4
Generalization Ablation

	3D-based swap			FSGAN		
Methods	Fake	Real	Total	Fake	Real	Total
Face identity difference	47.33	77.66	62.50	34.66	80.50	57.58
Binary XceptionNet [10]	55.38	**97.72**	76.55	24.80	94.68	59.74
Ours (frozen)	52.79	96.44	74.62	34.76	92.46	63.61
Ours (end-to-end)	54.74	97.70	76.22	31.66	**95.38**	63.52
Ours (concat)	55.42	96.54	75.98	41.64	93.30	67.47
Ours	**68.20**	95.10	**81.65**	**47.14**	90.56	**68.85**
Face identity difference	60.20	66.12	63.16	38.96	77.50	58.23
Binary XceptionNet [10]	89.03	81.36	85.20	73.92	64.04	68.98
Ours (frozen)	85.52	86.92	86.22	67.66	76.20	71.93
Ours (end-to-end)	90.77	83.54	87.16	**79.58**	71.40	**75.49**
Ours (concat)	**91.41**	84.34	87.87	71.92	**78.00**	74.96
Ours	90.52	**88.20**	**89.36**	78.72	71.66	75.19

Generalization results of variants of our method on our extended version of FaceForensics++ [5] test set. Top: Results with a fixed threshold at zero. Bottom: Upper bound results, obtained with a fixed threshold maximizing total accuracy on the test set. See Section 5.3 for more details.

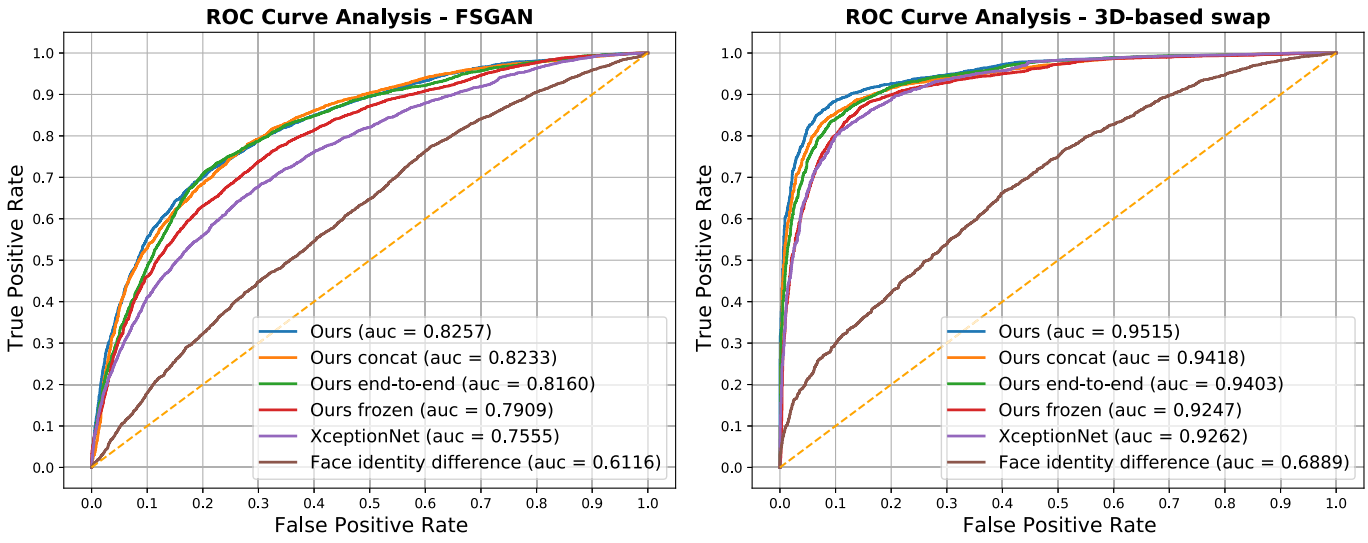

Fig. 6. *Results on our two variations of FaceForensics++ videos.* (a) Generalization results with FSGAN generated swaps [7]. (b) Generalization results with swaps generated by Nirkin *et al.* [18]. See Section 5.3 for more details.

TABLE 5
Image Laundry Ablation

Methods	3D-based swap							FSGAN							FaceShifter						
	RAW	C25	C50	C75	S25	S50	S75	RAW	C25	C50	C75	S25	S50	S75	RAW	C25	C50	C75	S25	S50	S75
Face identity difference	63.16	62.41	63.19	62.73	61.67	62.58	63.02	58.23	57.79	57.32	56.25	56.25	57.72	58.13	55.89	55.92	56.62	55.65	54.19	55.33	55.85
Binary XceptionNet [10]	85.20	84.00	83.07	80.24	75.87	82.09	83.92	68.98	67.68	66.66	63.68	64.77	69.84	70.06	63.62	63.79	62.76	61.90	59.35	63.51	64.43
Ours	**89.36**	**87.49**	**85.79**	**82.21**	**77.51**	**85.46**	**88.06**	**75.19**	**73.60**	**71.96**	**68.61**	**68.07**	**75.03**	**75.85**	**67.64**	**66.96**	**65.85**	**64.28**	**61.47**	**66.66**	**67.52**

Generalization results on three face swapping methods using the videos from FaceForensics++ [5] test set: 3D-based swap [18], FSGAN [7], and FaceShifter [38], where the images are subject to different resizing and compressions.'RAW', the image is unaltered, 'C##', JPEG compression operation (higher percentage means stronger compression), and 'S##' is a scaling operation (percentage relative to original resolution).

compression, and scaling relative to the original resolution, also 25, 50, and 75 percent.

The results are detailed in Table 5. As expected, when applying more than 25 percent compression and more aggressive scaling than 75 percent, the laundry attacks reduce the accuracy of all the detection methods and the larger the compression or the scaling, the larger the drop in accuracy. Our method consistently outperforms XceptionNet [10] and the face identity difference baseline, by a margin, under all the different laundering attacks.

Finally, the results indicate that the face identity difference becomes less effective on the more recent methods. Recent face swapping methods improve the estimated pose and expression of the target face. These methods therefore allow for a more of the target face's identity signal to remain, and hence reduce the effectiveness of the face identity difference.

5.4 Qualitative Results

Fig. 7 presents qualitative examples of detected and missed fake faces from the DFDC collection. Fig. 7a shows example fakes detected by our method but undetected by the state of the art XceptionNet fake detector [60]. Fig. 7b offers example fakes which were detected by XceptionNet, but were missed by our method. Finally, Fig. 7c shows fakes missed by both approaches.

Clearly, our method excels in cases where swapping artifacts are hard to detect (Fig. 7a). Examining Fig. 7b shows that fake images detected by XceptionNet often exhibit visible artifacts, which that method was optimized to detect. Our method includes a face swapping component, E_s (Section 4.2), trained to detect similar method-specific artifacts, but does not provide the same detection accuracy as the baseline when such artifacts are present. Our overall approach still outperforms the baseline by a wide margin, as reported in Sections 5.1 and 5.2. Finally, the fakes missed by both methods are typically challenging images with low contrast or blurry features as in Fig. 7c.

6 Discussion and Limitations

Some of the most recent methods perform face manipulation by generating the entire head [65], [66]. Those methods usually employ a pretrained StyleGAN2 [67] network, or employ its architecture. The generation is controlled by manipulating StyleGAN2's latent code to maintain the source identity and preserve the attributes of the target face.

While the methods are successful in maintaining the appearance of the source face and incorporating the attributes of the target face, the pose and expression are currently less accurate. As a result, the methods lack temporal coherence when applied to videos.

In the future, those methods might overcome the current limitation, and will be able to perform full head face swapping in videos. This would create a new class of methods for which the assumptions underlying our method will not hold.

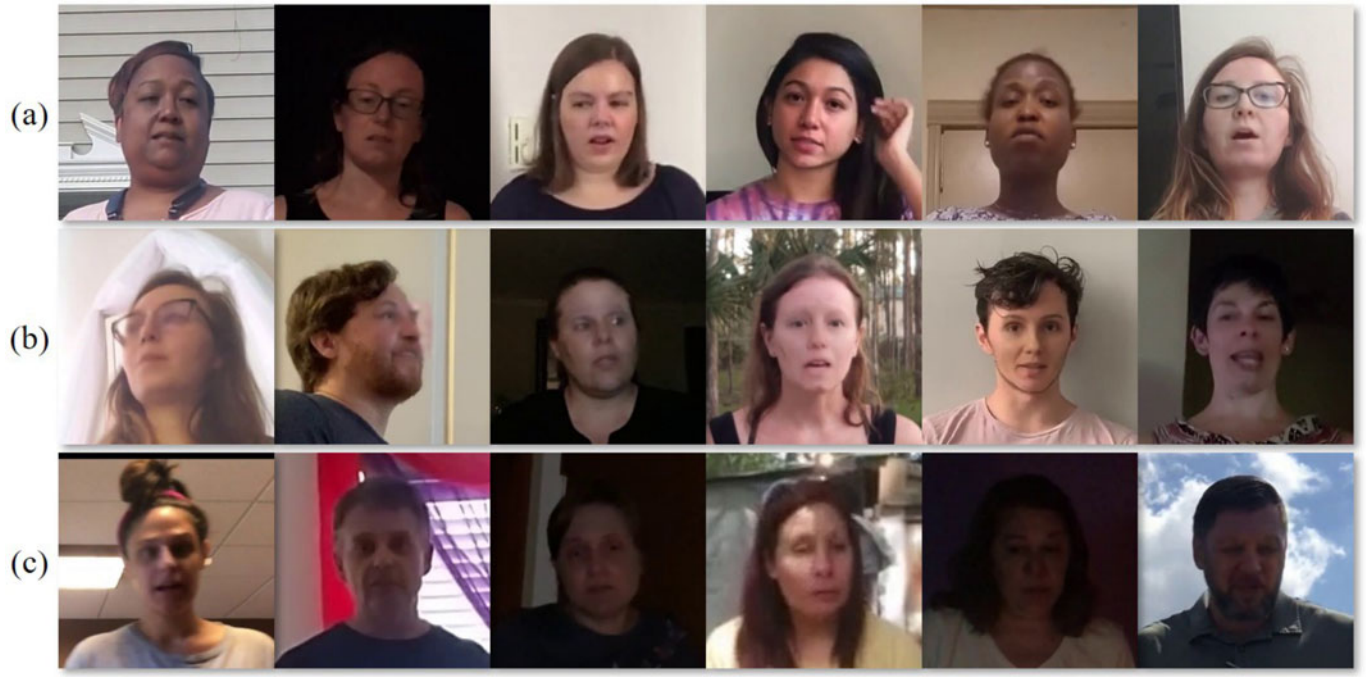

Fig. 7. *Qualitative detection results.* Examples taken from the DFDC collection. (a) Fakes detected by our method, but undetected by a leading baseline, XceptionNet fake detector [60]. (b) Fakes detected by XceptionNet but missed by our approach. (c) Fakes missed by both methods. See Section 5.4 for more details.

6.1 Face Reenactment Detection Cues

Face reenactments are detected by the E_r network (see Section 4.2) that is *specifically trained* to differentiate real images from those manipulated by face reenactment methods. Moreover, considering Figs. 2a and 2b, it seems that the manipulated regions of face reenactment methods resemble those created by face swapping schemes. Thus, E_f and E_c might utilize cues, other than the subject identity, such as those due to the sensor and lenses. Those marks might be overridden in the synthesis process, and might improve the detection of face swapping manipulations as well.

7 Conclusion

While the ability to manipulate faces in images and video has increased dramatically in the last few years, most recent methods follow similar patterns. In this work, we propose a novel detection cue which utilizes the commonalities of all recent face identity manipulation methods. It is complementary to conventional real/fake classifiers and can be used alongside them. Overcoming this approach would require a much broader integration of the new identity into the image, making our contribution hard to circumvent without additional technological breakthroughs. This is in contrast to artifact detection methods, which are susceptible to the constant progress in the visual quality of generated images. It is our hope that by further analyzing the design principles of face swapping techniques, additional methods of identifying fake images and videos would be discovered, leading to effective mitigation of the societal risks of such media.

Acknowledgments

This work was supported by the European Research Council (ERC) through the European Unions Horizon 2020 research and innovation programme under Grant ERC CoG 725974. Lior Wolf, Yosi Keller, and Tal Hassner have equally contributed.

References

[1] Google AI, "Contributing data to deepfake detection research." [Online]. Available: https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html
[2] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of RGB videos," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2387–2395.
[3] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," 2019, *arXiv:1904.12356*.
[4] Deepfakes, "Deepfakes." Accessed: Nov. 15, 2019. [Online]. Available: https://github.com/deepfakes/faceswap
[5] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," 2019, *arXiv:1901.08971*.
[6] FaceSwap, "FaceSwap." Accessed: Nov. 15, 2019. [Online]. Available: https://github.com/MarekKowalski/FaceSwap/
[7] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 7184–7193.
[8] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *Proc. Int. Workshop Inf. Forensics Secur.*, 2018, pp. 1–7.
[9] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proc. Int. Workshop Inf. Hiding Multimedia Secur.*, 2016, pp. 5–10.
[10] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection," in *Proc. Int. Workshop Inf. Hiding Multimedia Secur.*, 2017, pp. 159–164.
[11] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inform. Forensics Secur.*, vol. 7, no. 3, pp. 868–882, Jun. 2012.
[12] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen, "Distinguishing computer graphics from natural images using convolution neural networks," in *Proc. Int. Workshop Inf. Forensics Secur.*, 2017, pp. 1–6.

[13] U. A. Ciftci, I. Demir, and L. Yin, "How do the hearts of deep fakes beat? deep fake source detection via interpreting residuals with biological signals," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, 2020, pp. 1–10.

[14] H. Qi *et al.*, "Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 4318–4327.

[15] S. Hu, Y. Li, and S. Lyu, "Exposing GAN-generated faces using inconsistent corneal specular highlights," 2020, *arXiv:2009.11924*.

[16] L. Li *et al.*, "Face X-ray for more general face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5001–5010.

[17] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 365–372.

[18] Y. Nirkin, I. Masi, A. T. Tuan, T. Hassner, and G. Medioni, "On face segmentation, face swapping, and face perception," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2018, pp. 98–105.

[19] V. Blanz, S. Romdhani, and T. Vetter, "Face identification across different poses and illuminations with a 3D morphable model," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2002, pp. 192–197.

[20] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1063–1074, Sep. 2003.

[21] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A new dataset for deepfake forensics," 2019, *arXiv:1909.12962*.

[22] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (DFDC) preview dataset," 2019, *arXiv:1910.08854*.

[23] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar, "Face swapping: Automatically replacing faces in photographs," *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. no. 39.

[24] V. Blanz, K. Scherbaum, T. Vetter, and H.-P. Seidel, "Exchanging faces in images," *Comput. Graph. Forum*, vol. 23, no. 3, pp. 669–676, 2004.

[25] Y. Lin, S. Wang, Q. Lin, and F. Tang, "Face swapping under large pose variations: A 3D model based approach," in *Proc. Int. Conf. Multimedia Expo*, 2012, pp. 333–338.

[26] S. Mosaddegh, L. Simon, and F. Jurie, "Photorealistic face de-identification by aggregating donors' face components," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 159–174.

[27] I. Kemelmacher-Shlizerman, "Transfiguring portraits," *ACM Trans. Graph.*, vol. 35, no. 4, 2016, Art. no. 94.

[28] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec, "Creating a photoreal digital actor: The digital emily project," in *Proc. Conf. Vis. Media Prod.*, 2009, pp. 176–187.

[29] L. Wolf, Z. Freund, and S. Avidan, "An eye for an eye: A single camera gaze-replacement method," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 817–824.

[30] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman , "Synthesizing Obama: Learning lip sync from audio," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 95.

[31] H. Averbuch-Elor , D. Cohen-Or, J. Kopf, and M. F. Cohen, "Bringing portraits to life," *ACM Trans. Graph.*, vol. 36, no. 6, 2017, Art. no. 196.

[32] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer , "Ganimation: Anatomically-aware facial animation from a single image," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 818–833.

[33] E. Sanchez and M. Valstar, "Triple consistency loss for pairing distributions in GAN-based face synthesis," 2018, *arXiv:1811.03492*.

[34] H. Kim *et al.*, "Deep video portraits," *ACM Trans. Graph.*, vol. 37, no. 4, 2018, Art. no.163.

[35] R. Natsume, T. Yatagawa, and S. Morishima, "FSNet: An identity-aware generative model for image-based face swapping," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 117–132.

[36] R. Natsume, T. Yatagawa, and S. Morishima, "RsGAN: Face swapping and editing using face and hair representation in latent spaces," 2018, *arXiv:1804.03447*.

[37] K. Nagano *et al.*, "paGAN: Real-time avatars using dynamic textures," *ACM Trans. Graph. (TOG)*, vol. 37, no. 6, pp. 1–12, 2018.

[38] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Faceshifter: Towards high fidelity and occlusion aware face swapping," 2019, *arXiv:1912.13457*.

[39] S. Jia, Z. Xu, H. Wang, C. Feng, and T. Wang, "Coarse-to-fine copy-move forgery detection for video forensics," *IEEE Access*, vol. 6, pp. 25323–25335, 2018.

[40] Y. Wu, W. Abd-Almageed, and P. Natarajan, "Busternet: Detecting copy-move image forgery with source/target localization," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 168–184.

[41] Y. Wu, W. Abd-Almageed , and P. Natarajan, "Image copy-move forgery detection via an end-to-end deep neural network," in *Proc. Winter Conf. Appl. Comput. Vis.*, 2018, pp. 1907–1915.

[42] Y. Wu, W. AbdAlmageed, and P. Natarajan, "ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9543–9552.

[43] P. Korshunov and S. Marcel, "Speaker inconsistency detection in tampered video," in *Proc. Eur. Signal Process. Conf.*, 2018, pp. 2375–2379.

[44] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing AI generated fake face videos by detecting eye blinking," 2018, *arXiv:1806.02877*.

[45] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," 2018, *arXiv:1811.00656*.

[46] W. Quan, K. Wang, D.-M. Yan, and X. Zhang, "Distinguishing between natural and computer-generated images using convolutional neural networks," *Trans. Inform. Forensics Secur.*, vol. 13, no. 11, pp. 2772–2787, 2018.

[47] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics: A large-scale video dataset for forgery detection in human faces," 2018, *arXiv:1803.09179*.

[48] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," *CVPRw*, pp. 80–87, 2019.

[49] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *Proc. Winter Conf. Appl. Comput. Vis. Workshops*, 2019, pp. 83–92.

[50] H. H. Nguyen, T. Tieu, H.-Q. Nguyen-Son , V. Nozick, J. Yamagishi, and I. Echizen, "Modular convolutional neural network for discriminating between computer-generated images and photographic images," in *Proc. Int. Conf. Availability, Rel. Secur.*, 2018, pp. 1–10.

[51] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Use of a capsule network to detect fake images and videos," 2019, *arXiv:1910.12467*.

[52] S.-Y. Wang, O. Wang, A. Owens, R. Zhang, and A. A. Efros, "Detecting photoshopped faces by scripting photoshop," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 10072–10081.

[53] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 8261–8265.

[54] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-stream neural networks for tampered face detection," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1831–1839.

[55] J. Stehouwer, H. Dang, F. Liu, X. Liu, and A. Jain, "On the detection of digital face manipulation," 2019, *arXiv:1910.01717*.

[56] K. Songsri-in and S. Zafeiriou, "Complement face forensic detection and localization with faciallandmarks," 2019, *arXiv:1910.05455*.

[57] P. Korshunov and S. Marcel, "Vulnerability assessment and detection of deepfake videos," in *Proc. Int. Conf. Biometrics*, 2019, pp. 1–6.

[58] O. I. Al-Sanjary , A. A. Ahmed, and G. Sulong, "Development of a video tampering dataset for forensic investigation," *Forensic Sci. Int.*, vol. 266, pp. 565–572, 2016.

[59] J. Li *et al.*"DSFD: Dual shot face detector," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5060–5069.

[60] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.

[61] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[62] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2018, pp. 67–74.

[63] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller , "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," UMass Amherst, Univ. Massachusetts, *Tech. Rep.* 07–49, 2007.

[64] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," 2019, *arXiv:1906.06876*.

[65] Y. Shen and B. Zhou, "Closed-form factorization of latent semantics in GANs," 2020, *arXiv:2007.06600*.

[66] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "GANSpace: Discovering interpretable GAN controls," 2020, *arXiv:2004.02546*.
[67] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of styleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8110–8119.
[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[69] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.
[70] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2018, pp. 59–66.

Yuval Nirkin received the BSc degree in computer engineering from the Technion Israel Institute of Technology, Haifa, in 2011 and the MSc degree in computer science from The Open University of Israel, Ra'anana, Israel, in 2017. He is currently working toward the PhD degree with the Faculty of Electrical Engineering, Bar-Ilan University, Ramat-Gan, Israel. His research interests include deep learning, computer vision, and computer graphics. He was a reviewer of ECCV, ICCV, and CVPR, and was recognized as a high quality reviewer in ECCV'20.

Lior Wolf received the PhD degree from the Hebrew University, under the supervision of Prof. Shashua. He is currently a full professor with the School of Computer Science, Tel-Aviv University, Israel. He was a postdoctoral researcher with prof. Poggio's lab, Massachusetts Institute of Technology. He is an ERC grantee and was the recepient of ICCV 2001 and ICCV 2019 honorable mention, and the best paper awards at ECCV 2000 and ICANN 2016.

Yosi Keller received the BSc degree in electrical engineering from the Technion Israel Institute of Technology, Haifa, in 1994, and the MSc and PhD degrees, summa cum laude, in electrical engineering from Tel Aviv University in 1998 and 2003, respectively. From 2003 to 2006, he was a Gibbs assistant professor with the Department of Mathematics, Yale University, New Haven, CT, USA. He is currently an associate professor with the Faculty of Engineering, Bar Ilan University, Ramat-Gan, Israel. His research interests include computer vision, machine and deep learning, and biometrics.

Tal Hassner received the MSc and PhD degrees in applied mathematics and computer science from the Weizmann Institute of Science in 2002 and 2006, respectively. In 2008 he joined the Department of Mathematics and Computer Science, The Open University of Israel, where he was an associate professor until 2018. From 2015 to 2018, he was a senior computer scientist with the Information Sciences Institute (ISI) and a visiting research associate professor with the Institute for Robotics and Intelligent Systems, USC Viterbi School of Engineering, CA, USA. From 2018 to 2019, he was a principal applied scientist with AWS Rekognition where he designed the latest AWS face recognition pipelines. Since 2019 he has been an applied research lead with Facebook AI, supporting both text (OCR) and people (faces) photo understanding teams. He has been a program chair at WACV'18 and ICCV'21. He was also a workshop chair at CVPER'20, a tutorial chair at ICCV'17 and ECCV'22, and the area chair for CVPR, ECCV, and AAAI. He is an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and the *IEEE Transactions on Biometrics, Behavior, and Identity Science*.