

DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection

Liming Jiang¹ Wayne Wu^{1,2} Ren Li² Chen Qian² Chen Change Loy¹

¹Nanyang Technological University ²SenseTime Research

{liming002, ccloy}@ntu.edu.sg {wuwenyan, qianchen}@sensetime.com
tomo.blade.lee@hotmail.com

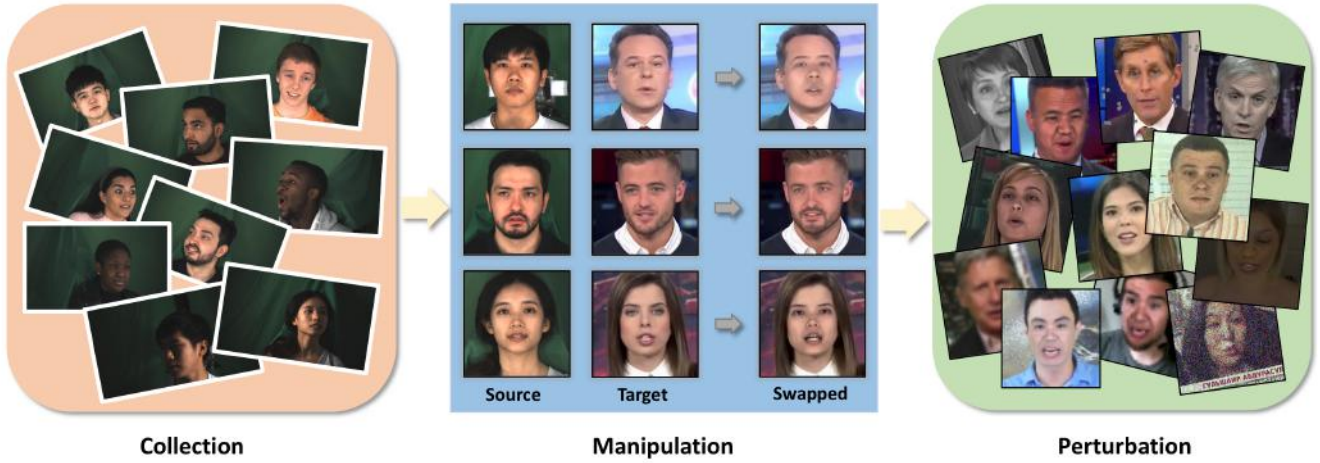


Figure 1: DeepForensics-1.0 dataset is a new large-scale dataset for *real-world* face forgery detection.

Abstract

In this paper, we present our on-going effort of constructing a large-scale benchmark, *DeeperForensics-1.0*, for face forgery detection. Our benchmark represents the largest face forgery detection dataset by far, with 60,000 videos constituted by a total of 17.6 million frames, 10 times larger than existing datasets of the same kind. Extensive real-world perturbations are applied to obtain a more challenging benchmark of larger scale and higher diversity. All source videos in *DeeperForensics-1.0* are carefully collected, and fake videos are generated by a newly proposed end-to-end face swapping framework. The quality of generated videos outperforms those in existing datasets, validated by user studies. The benchmark features a hidden test set, which contains manipulated videos achieving high deceptive scores in human evaluations. We further contribute a comprehensive study that evaluates five representative detection baselines and make a thorough analysis of different settings. We believe this dataset will contribute to real-world face forgery detection research.¹²

¹Code, model and data will be made publicly available.

²Project page: <https://liming-jiang.com/projects/DrF1/DrF1.html>.

1. Introduction

Face swapping has become an emerging topic in computer vision and graphics. Indeed, many works [1, 2, 4] on automatic face swapping have been proposed in recent years. These efforts have circumvented the cumbersome and tedious manual face editing processes, hence expediting the advancement in face editing. At the same time, such enabling technology has sparked legitimate concerns, particularly on its potential for being misused and abused. The popularization of “Deepfakes” on the internet has further set off alarm bells among the general public and authorities, in view of the conceivable perilous implications. Accordingly, there is a dire need for countermeasures to be in place promptly, particularly innovations that can effectively detect videos that have been manipulated.

Working towards forgery detection, various groups have contributed datasets (e.g., FaceForensics++ [37], Deep Fake Detection [9] and DFDC [14]) comprising manipulated video footages. The availability of these datasets has undoubtedly provided essential avenues for research into forgery detection. Nonetheless, the aforementioned datasets suffer several drawbacks. Videos in these datasets are either of a small number, of low quality, or overly arti-

cial. Understandably, these datasets are inadequate to train a good model for effective forgery detection in *real-world* scenarios. This is particularly true when current advances in human face editing are able to produce extremely realistic videos, rendering forgery detection a highly challenging task. On another note, we observe high similarity between training and test videos, in terms of their distribution, in certain works [30, 37]. Their actual efficacy in detecting *real-world* face forgery cases, which are much more variable and unpredictable, remains to be further elucidated.

We believe that forgery detection models can only be enhanced when trained with a dataset that is exhaustive enough to encompass as many potential real-world variations as possible. To this end, we propose a large-scale dataset named DeeperForensics-1.0 consisting of 60,000 videos with a total of 17.6 million frames for real-world face forgery detection. The main steps of our dataset construction are shown in Figure 1. We set forth three yardsticks when constructing this dataset: 1) *Quality*. The dataset shall contain videos more realistic and much closer to the distribution of real-world detection scenarios. (details in Section 3.1 and 3.2) 2) *Scale*. The dataset shall be made up of a large-scale video sets. (details in Section 3.3) 3) *Diversity*. There shall be sufficient variations in the video footages (*e.g.*, compression, blurry, transmission errors) to match those that may be encountered in the real world (details in Section 3.3).

The primary challenge in the preparation of this dataset is the lack of good-quality video footages. Specifically, most publicly available videos are shot under an unconstrained environment resulting in large variations, including but not limited to suboptimal illumination, large occlusion of the target faces, and extreme head poses. Importantly, the lack of official informed consents from the video subjects precludes the use of these videos, even for non-commercial purposes. On the other hand, while some videos of manipulated faces are deceptively real, a larger number remains easily distinguishable by human eyes. The latter is often caused by model negligence towards appearance variations or temporal differences, leading to preposterous and incongruous results.

We approach the aforementioned challenge from two perspectives. 1) Collecting fresh face data from 100 individuals with informed consents (details in Section 3.1) 2) Devising a novel method, DeepFake Variational Auto-Encoder (DF-VAE), to enhance existing videos (details in Section 3.2). In addition, we introduce diversity into the video footages through deliberate addition of distortions and perturbations, simulating real-world scenarios. We collate the newly collected data and the DF-VAE-modified videos into the DeeperForensics-1.0 dataset, with the aim of further expanding it gradually over time. We benchmark five representative open-source forgery detection methods using our DeeperForensics-1.0 dataset as well as a hidden

test set containing manipulated videos that achieve high deceptive ranking in user studies.

We summarize our contributions as follows: 1) We propose a new dataset, DeeperForensics-1.0 that is larger in scale than existing ones, of high quality and rich diversity. To improve its quality, we introduce a carefully designed data collection and a novel framework, DF-VAE, that effectively mitigate the obvious fabricated effects of existing manipulated videos. DeeperForensics-1.0 dataset shall facilitate future research in forgery detection of human faces in real-world scenarios. 2) We benchmark results of existing representative forgery detection methods on our DeeperForensics-1.0 dataset, offering insights into the current status and future improvisation strategy in face forgery detection.

2. Related Work

This paper includes two main aspects of face forgery detection related to other works: dataset and benchmark. We will cover some important works in this section.

Face forgery detection datasets. Building a dataset for forgery detection requires a huge amount of effort on data collection and manipulation. Early forgery detection datasets are based on images under highly restrictive conditions, *e.g.*, MICC_F2000 [7], Wild Web dataset [50], Realistic Tampering dataset [28].

Owing to the urgency in video-based face forgery detection, some prominent groups have devoted their efforts to create face forensics video datasets (see Table 1). UADFV [49] contains 98 videos, *i.e.*, 49 real videos from YouTube and 49 fake ones generated by FakeAPP [5]. DeepFake-TIMIT [27] manually selects 16 similar looking pairs of people from VidTIMIT [39] database. For each of the 32 subjects, they generate about 10 videos using low-quality and high-quality versions of faceswap-GAN [4], resulting in a total of 620 fake videos. Celeb-DF [30] includes 408 YouTube videos, mostly of celebrities, from which 795 fake videos are synthesized. FaceForensics++ [37] is the first large-scale face forensic dataset that consists of 4000 fake videos manipulated by four methods (*i.e.*, DeepFakes [2], Face2Face [42], FaceSwap [3], NeuralTextures [41]), and 1000 real videos from YouTube. Afterwards, Google joins FaceForensics++ and contributes Deep Fake Detection [9] dataset with 3431 real and fake videos from 28 actors. Recently, Facebook invites 66 individuals and builds the DFDC preview dataset [14], which includes 5214 original and tampered videos with three types of augmentations.

In comparison, we invite 100 paid actors and collect high-resolution (1920×1080) source data with various poses, expressions, and illuminations. 3DMM blendshapes [10] are taken as reference to supplement some extremely exaggerated expressions. We get consents from all the actors for using and manipulating their faces. In contrast to prior works, we also propose a new end-to-end face swap-

Table 1: The most relevant datasets compared to our dataset. DeeperForensics-1.0 is an order of magnitude in scale than existing datasets *w.r.t.* both real and fake parts. We build a professional indoor environment to better control the important attributes of the collected data. 100 paid actors give consents to the use and manipulation of their faces by signing a formal agreement. We employ seven types of perturbations at five intensity levels, leading to 35 perturbations in total. The video may be subjected to a mixture of more than one perturbation. In contrast to prior works, we also introduce a new end-to-end high-fidelity face swapping method.

Dataset	Total videos	Ratio (real : fake)	Controlled Capture	Consented Actors	Perturbations (total number)	Perturbations (mixture)	New Method
UADFV [49]	98	1 : 1	×	—	—	×	×
DeepFake-TIMIT [27]	620	only fake	×	—	—	×	×
Celeb-DF [30]	1203	1 : 1.95	×	—	—	×	×
FaceForensics++ [37]	5000	1 : 4	×	—	2	×	×
Deep Fake Detection [9] (joins FaceForensics++)	3431	1 : 8.5	×	28	—	×	×
DFDC Preview Dataset [14]	5214	1 : 3.6	×	66	3	×	×
DeeperForensics-1.0 (Ours)	60000	5 : 1	✓	100	35	✓	✓

ping method (*i.e.*, DF-VAE) and systematically apply seven types of perturbations to the fake videos at five intensity levels. The mixture of distortions to a single video make our dataset better imitate real-world scenarios. Ultimately, we construct DeeperForensics-1.0 dataset, which contains up to 60,000 high-quality videos with a total of 17.6 million frames.

Face forgery detection benchmarks. Recently, a new prominent benchmark, FaceForensics Benchmark [37], for facial manipulation detection has been proposed. The benchmark includes six image-level face forgery detection baselines [6, 8, 12, 13, 15, 36]. Although FaceForensics Benchmark adds distortions to the videos by converting them into different compression rates, a deeper exploration of more perturbation types and their mixture is missing. Celeb-DF [30] also provides a face forgery detection benchmark including seven methods [6, 12, 29, 32, 34, 49, 51] trained and tested on different datasets. In aforementioned benchmarks, the test set usually shares a similar distribution with the training set. Such an assumption inherently introduces biases and renders these methods impractical for face forgery detection in real-world settings with much more diverse and unknown fake videos.

In our benchmark, we introduce a challenging hidden test set with manipulated videos that achieve high deceptive scores in user studies, to better simulate *real-world* distribution. Various perturbations are analyzed to make our benchmark more comprehensive. In addition, we mainly exploit *video-level* forgery detection baselines [11, 17, 19, 43, 45] in this work. Temporal information – a significant cue for video forgery detection besides single-frame quality – has been considered. We will elaborate our benchmark in Section 4.

3. A New Large-Scale Face Forensics Dataset

The main contribution of this paper is a new large-scale dataset for real-world face forgery detection, DeeperForensics-1.0, which provides an alternative to ex-

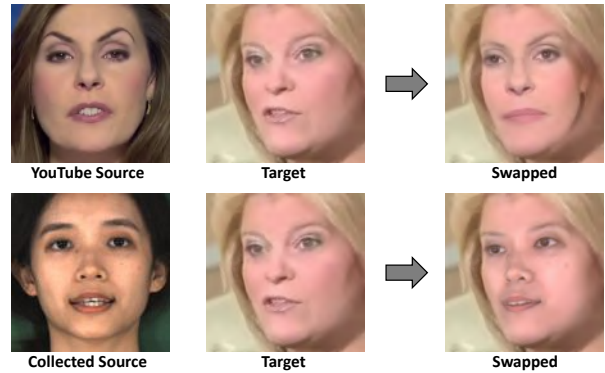


Figure 2: Comparison of using only YouTube video and the collected video as source data, with the same method and setting.

isting databases. DeeperForensics-1.0 consists of 60,000 videos with 17.6 million frames in total, including 50,000 original collected videos and 10,000 manipulated videos. To construct a dataset more suitable for real-world face forgery detection, we design this dataset with careful consideration of *quality*, *scale*, and *diversity*. In Section 3.1 and 3.2, we will discuss the details of data collection and methodology (*i.e.*, DF-VAE) to improve *quality*. In Section 3.3, we will show how to ensure large *scale* and high *diversity* of DeeperForensics-1.0.

3.1. Data Collection

Source data is the first factor that highly affects *quality*. Taking results in Figure 2 as an example, the source data collection increases the robustness of our face swapping method to extreme poses, since videos on the internet usually have limited head pose variations.

We refer to the identity in the driving video as the “target” face and the identity of the face that is swapped onto the driving video as the “source” face. Different from previous works, we find that the source faces play a much more critical role than the target faces in building a high-quality



Figure 3: Diversity in identities, poses, expressions, and illuminations in our collected source data.

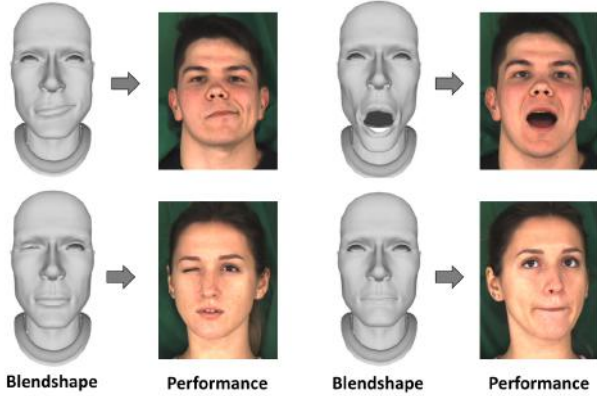


Figure 4: Examples of 3DMM blendshapes in our data collection.

dataset. Specifically, the expressions, poses, and lighting conditions of source faces should be much richer in order to perform robust face swapping. Hence, our data collection mainly focuses on source face videos. Figure 3 shows the diversity in different attributes of our data collection.

We invited 100 paid actors to record the source videos. Similar to [9, 14], we obtained consents from all the actors for using and manipulating their faces to avoid the portrait right issues. The participants were carefully selected to ensure variability in genders, ages, skin colors, and nationalities. We tried to maintain a roughly equal proportion *w.r.t.* each of the attributes above. In particular, we invited 53 males and 47 females from 26 countries. The age ranges from 20 to 45 years old to match the most common age group appearing on real-world videos. The actors have four typical skin tones: *white*, *black*, *yellow*, *brown*, with ratio 1:1:1:1. All faces are clean without glasses or decorations.

Different from previous data collection in the wild (see Table 1), we built a professional *indoor* environment for a more controllable data collection. We only use the facial regions (detected and cropped by LAB [47]) of the source data, so we can neglect the background. We set seven HD cameras from different angles: front, left, left-front, right, right-front, oblique-above, oblique-below. The resolution of our recorded videos is high (1920×1080). We trained the



Figure 5: Examples of style mismatch problems in prominent face forensics datasets.

actors in advance to keep the collection process smooth. We requested the actors to turn their heads and speak naturally with eight expressions: neutral, angry, happy, sad, surprise, contempt, disgust, fear. The head poses range from -90° to $+90^\circ$. Furthermore, the actors were asked to perform 53 expressions defined in 3DMM blendshapes [10] (see Figure 4) to supplement some extremely exaggerated expressions. When performing 3DMM blendshapes, the actors also spoke naturally to avoid excessive frames that show a closed mouth.

In addition to expressions and poses, we systematically set nine lighting conditions from various directions: uniform, left, top-left, bottom-left, right, top-right, bottom-right, top, bottom. The actors were only asked to turn their heads under uniform illumination, so the lighting remains unchanged on specific facial regions to avoid many duplicate data samples recorded by the cameras set at different angles. In the end, our collected data contain over 50,000 videos with a total of 12.6 million frames – an order of magnitude more than existing datasets.

3.2. DeepFake Variational Auto-Encoder

To tackle low visual *quality* problems of previous works, we consider three key requirements in formulating a high-fidelity face swapping method: 1) It should be general and scalable for us to generate large number of videos with high quality. 2) The problem of face style mismatch caused by appearance variations need to be addressed. Some failure cases of existing methods are shown in Figure 5. 3) Temporal continuity of generated videos should be taken into consideration.

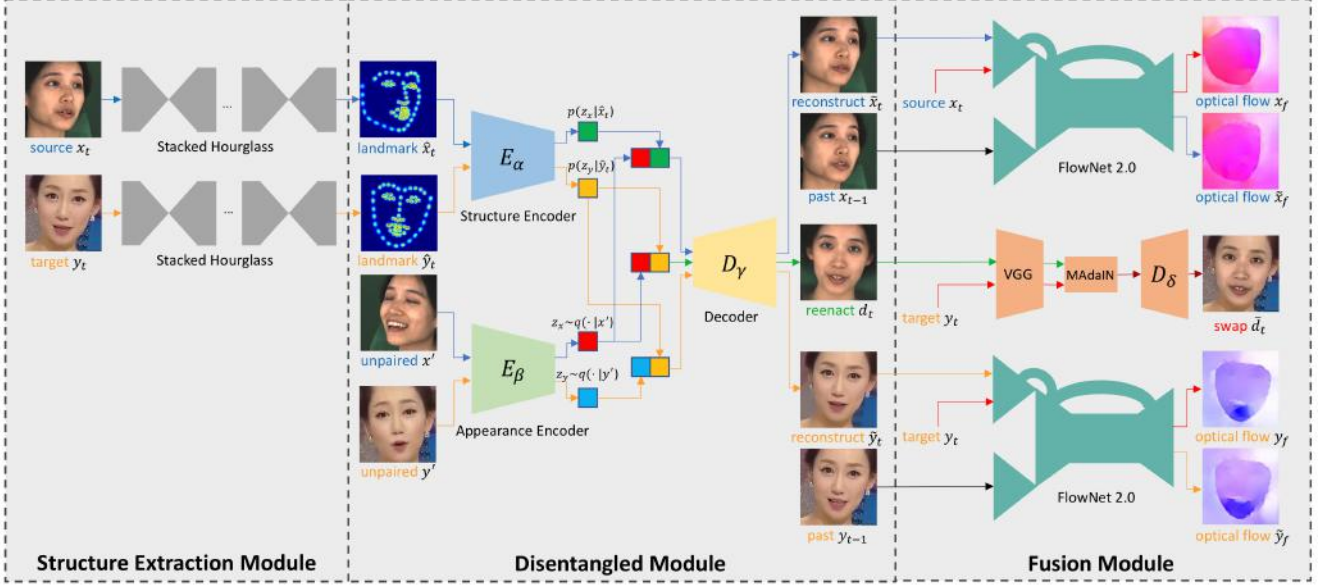


Figure 6: The main framework of DeepFake Variational Auto-Encoder. In training, we reconstruct the source and target faces in blue and orange arrows, respectively, by extracting landmarks and constructing an unpaired sample as the condition. Optical flow differences are minimized after reconstruction to improve temporal continuity. In inference, we swap the latent codes and get the reenacted face in green arrows. Subsequent MAdaIN module fuses the reenacted face and the original background resulting in the swapped face.

Based on the aforementioned requirements, we propose DeepFake Variational Auto-Encoder (DF-VAE), a novel learning-based face swapping framework. DF-VAE consists of three main parts, namely a structure extraction module, a disentangled module, and a fusion module. We will give a brief and intuitive understanding of the DF-VAE framework below. Please refer to the *Appendix* for detailed derivations and results.

Disentanglement of structure and appearance. The first step of our method is face reenactment – animating the source face with similar expression as the target face, without any paired data. Face swapping is considered as a subsequent step of face reenactment that performs fusion between the reenacted face and the target background. For robust and scalable face reenactment, we should cleanly disentangle structure (*i.e.*, expression and pose) and appearance representation (*i.e.*, texture, skin color, *etc.*) of a face. This disentanglement is rather difficult because structure and appearance representation are far from independent. We describe our solution as follows.

Let $\mathbf{x}_{1:T} \equiv \{x_1, x_2, \dots, x_T\} \in X$ be a sequence of source face video frames, and $\mathbf{y}_{1:T} \equiv \{y_1, y_2, \dots, y_T\} \in Y$ be the sequence of corresponding target face video frames. We first simplify our problem and only consider two specific snapshots at time t , x_t and y_t . Let $\tilde{x}_t, \tilde{y}_t, d_t$ represent the reconstructed source face, the reconstructed target face, and the reenacted face, respectively.

Consider the reconstruction procedure of the source face x_t . Let s_x denotes the structure representation and a_x denotes the appearance information. The face generator can

be depicted as the posteriori estimate $p_\theta(x_t | s_x, a_x)$. The solution of our reconstruction goal, marginal log-likelihood $\tilde{x}_t \sim \log p_\theta(x_t)$, by a common Variational Auto-Encoder (VAE) [26] can be written as:

$$\log p_\theta(x_t) = D_{KL}(q_\phi(s_x, a_x | x_t) || p_\theta(s_x, a_x | x_t)) + L(\theta, \phi; x_t), \quad (1)$$

where q_ϕ is an approximate posterior to achieve the evidence lower bound (ELBO) in the intractable case, and the second RHS term $L(\theta, \phi; x_t)$ is the variational lower bound *w.r.t.* both the variational parameters ϕ and generative parameters θ .

In Eq. 1, we assume that both s_x and a_x are latent priors computed by the same posterior x_t . However, the separation of these two variables in the latent space is rather difficult without additional conditions. Therefore, we employ a simple yet effective approach to disentangle these two variables.

The blue arrows in Figure 6 demonstrate the reconstruction procedure of the source face x_t . Instead of feeding a single source face x_t , we sample another source face x' to construct unpaired data in the source domain. To make the structure representation more evident, we use the stacked hourglass networks [33] to extract landmarks of x_t in the structure extraction module and get the heatmap \hat{x}_t . Then we feed the heatmap \hat{x}_t to the Structure Encoder E_α , and x' to the Appearance Encoder E_β . We concatenate the latent representations (small cubes in red and green) and feed it to the Decoder D_γ . Finally, we get the reconstructed face \tilde{x}_t , *i.e.*, marginal log-likelihood of x_t .

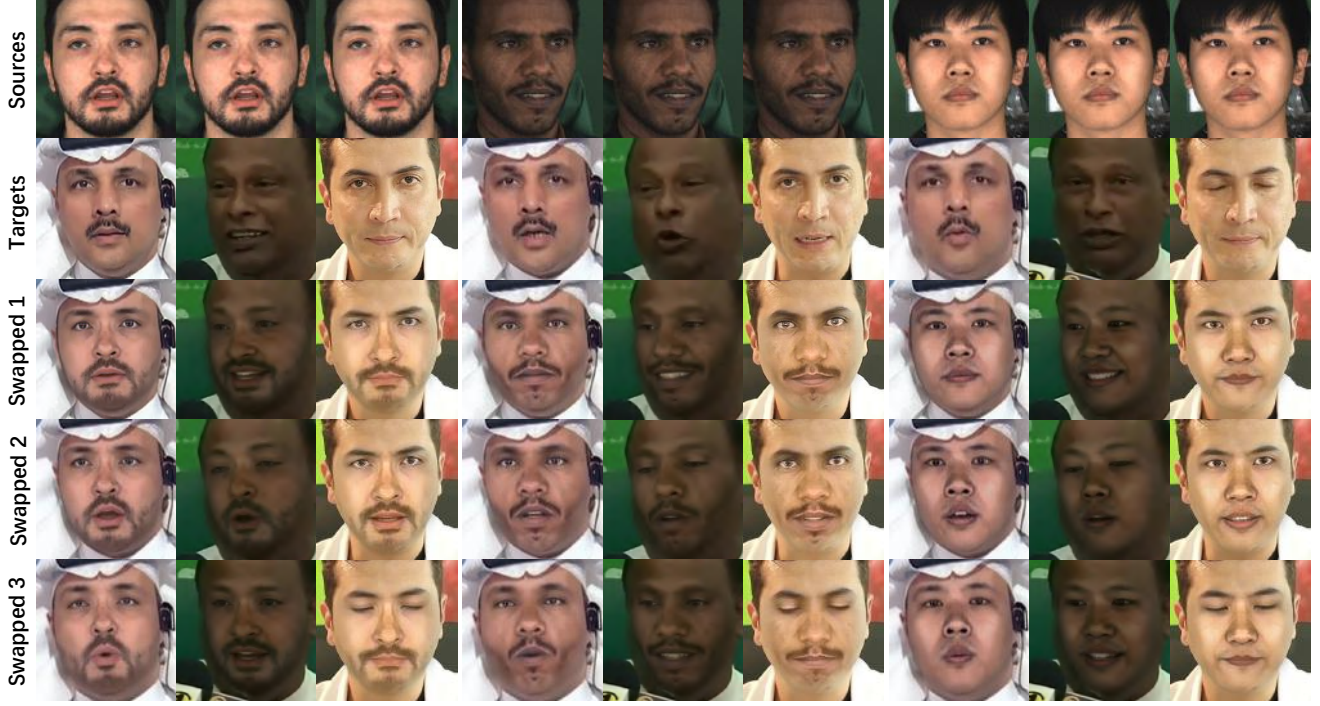


Figure 7: Many-to-many (three-to-three) face swapping by a **single** model with obvious reduction of style mismatch problems. This figure shows the results between three source identities and three target identities. The whole process is end-to-end.

Therefore, the latent structure representation s_x in Eq. 1 becomes a more evident heatmap representation \hat{x}_t , which is introduced as a new condition. The unpaired sample x' with the same identity *w.r.t.* x_t is another condition, being a substitute for a_x . Eq. 1 can be rewritten as a conditional log-likelihood:

$$\log p_\theta(x_t | \hat{x}_t, x') = D_{KL}(q_\phi(z_x | x_t, \hat{x}_t, x') \| p_\theta(z_x | x_t, \hat{x}_t, x')) + L(\theta, \phi; x_t, \hat{x}_t, x'), \quad (2)$$

The first RHS term KL-divergence is non-negative, we get:

$$\log p_\theta(x_t | \hat{x}_t, x') \geq L(\theta, \phi; x_t, \hat{x}_t, x') = \mathbb{E}_{q_\phi(z_x | x_t, \hat{x}_t, x')} [-\log q_\phi(z_x | x_t, \hat{x}_t, x') + \log p_\theta(x_t, z_x | \hat{x}_t, x')], \quad (3)$$

and $L(\theta, \phi; x_t, \hat{x}_t, x')$ can also be written as:

$$L(\theta, \phi; x_t, \hat{x}_t, x') = -D_{KL}(q_\phi(z_x | x_t, \hat{x}_t, x') \| p_\theta(z_x | \hat{x}_t, x')) + \mathbb{E}_{q_\phi(z_x | x_t, \hat{x}_t, x')} [\log p_\theta(x_t | z_x, \hat{x}_t, x')]. \quad (4)$$

We let the variational approximate posterior be a multi-variate Gaussian with a diagonal covariance structure:

$$\log q_\phi(z_x | x_t, \hat{x}_t, x') \equiv \log \mathcal{N}(z_x; \mu, \sigma^2 \mathbf{I}), \quad (5)$$

where \mathbf{I} is an identity matrix. Exploiting the reparameterization trick [26], the non-differentiable operation of

sampling can become differentiable by an auxiliary variable with independent marginal. In this case, $z_x \sim q_\phi(z_x | x_t, \hat{x}_t, x')$ is implemented by $z_x = \mu + \sigma\epsilon$ where ϵ is an auxiliary noise variable $\epsilon \sim \mathcal{N}(0, 1)$. Finally, the approximate posterior $q_\phi(z_x | x_t, \hat{x}_t, x')$ is estimated by the separated encoders, Structure Encoder E_α and Appearance Encoder E_β , in an end-to-end training process by standard gradient descent.

We discuss the whole workflow of reconstructing the source face. In the target face domain, the reconstruction procedure is the same, as shown by orange arrows in Figure 6.

During training, the network learns structure and appearance information in both the source and the target domains. It is noteworthy that even if both y_t and x' belong to arbitrary identities, our effective disentangled module is capable of learning meaningful structure and appearance information of each identity. During inference, we concatenate the appearance prior of x' and the structure prior of y_t (small cubes in red and orange) in the latent space, and the reconstructed face d_t shares the same structure with y_t and keeps the appearance of x' . Our framework allows concatenations of structure and appearance latent codes extracted from arbitrary identities in inference and permits *many-to-many face reenactment*.

In summary, DF-VAE is a new conditional variational auto-encoder [25] with robustness and scalability. It condi-



Figure 8: Comparison of the swapped face styles without or with MAdaIN module.

tions on two posteriors in different domains. In the disentangled module, the separated design of two encoders E_α and E_β , the explicit structure heatmap, and the unpaired data construction jointly force E_α to learn structure information and E_β to learn appearance information.

Style matching and fusion. To fix the obvious style mismatch problems as shown in Figure 5, we introduce a masked adaptive instance normalization (MAdaIN) module. We place a typical AdaIN [20] network after the reenacted face d_t . In the face swapping scenario, we only need to adjust the style of the face area and use the original background. Therefore, we use a mask m_t to guide AdaIN [20] network to focus on style matching of the face area. To avoid boundary artifacts, we apply Gaussian Blur to m_t and get the blurred mask m_t^b .

In our face swapping context, d_t is the content input of MAdaIN, y_t is the style input. MAdaIN adaptively computes the affine parameters from the face area of the style input:

$$\text{MAdaIN}(c, s) = \sigma(s) \left(\frac{c - \mu(c)}{\sigma(c)} \right) + \mu(s), \quad (6)$$

where $c = m_t^b \cdot d_t$, $s = m_t^b \cdot y_t$. With the very low-cost MAdaIN module, we reconstruct d_t again by Decoder D_δ . The blurred mask m_t^b is used again to fuse the reconstructed image with the background of y_t . At last, we get the swapped face \tilde{d}_t . Figure 8 shows the effectiveness of MAdaIN module for style matching and fusion.

The MAdaIN module is jointly trained with the disentangled module in an end-to-end manner. Thus, by a *single* model, DF-VAE can perform *many-to-many face swapping* with obvious reduction of style mismatch and facial boundary artifacts (see Figure 7 for the face swapping between three source identities and three target identities). Even if there are multiple identities in both the source domain and the target domain, the quality of face swapping does not degrade.

Temporal consistency constraint. Temporal discontinuity of fake videos leads to obvious flickering of the face area, making them very easy to be spotted by forgery detection methods and human eyes. To improve temporal continuity, we let the disentangled module to learn temporal information of both the source face and the target face.

Table 2: Seven types of distortions in DeeperForensics-1.0.

No.	Distortion Type
1	Color saturation change
2	Local block-wise distortion
3	Color contrast change
4	Gaussian blur
5	White Gaussian noise in color components
6	JPEG compression
7	Video compression rate change

For simplification, we make a Markov assumption that the generation of the frame at time t sequentially depends on its previous P frames $\mathbf{x}_{(t-P):(t-1)}$. In our experiment, we set $P = 1$ to balance quality improvement and training time.

In order to build the relationship between a current frame and previous ones, we further make an intuitive assumption that the optical flows should remain unchanged after reconstruction. We use FlowNet 2.0 [21] to estimate the optical flow \tilde{x}_f w.r.t. \tilde{x}_t and x_f w.r.t. x_t and x_{t-1} . Since face swapping is sensitive to minor facial details which can be greatly affected by flow estimation, we do not warp x_{t-1} by the estimated flow like [46]. Instead, we minimize the difference between \tilde{x}_f and x_f to improve temporal continuity while keeping stable facial detail generation. To this end, we propose a new temporal consistency constraint, which can be written as:

$$L_{\text{temporal}} = \frac{1}{CHW} \|\tilde{x}_f - x_f\|_1, \quad (7)$$

where $C = 2$ for a common form of optical flow.

We only discuss the temporal continuity w.r.t. the source face in this section because the case of the target face is the same. If multiple identities exist in one domain, temporal information of all these identities can be learned in an end-to-end manner.

3.3. Scale and Diversity

Our extensive data collection and the proposed DF-VAE method are designed to improve the *quality* of manipulated videos in DeeperForensics-1.0 dataset. In this section, we will mainly discuss the *scale* and *diversity* aspects.

We provide 10,000 manipulated videos with 5 million frames. It is also an order of magnitude more than the previous datasets. Thanks to the scalability and multimodality of DF-VAE, the time overhead of model training and data generation is reduced to 1/5 compared to the common Deepfakes methods, with no degradation in quality. Thus, a larger-scale dataset construction is possible.

We take 1000 refined YouTube videos collected by FaceForensics++ [37] as the target videos. Each face of our collected 100 identities is swapped onto 10 target videos, thus 1000 raw manipulated videos are generated directly by DF-VAE in an end-to-end process.

To ensure *diversity*, we apply various perturbations to better simulate videos in real scenes. Specifically, as shown in Table 2, seven types of distortions defined in Image Quality Assessment (IQA) [31, 35] are included. Each of these distortions is divided into five intensity levels. We apply random-type distortions to the 1000 raw manipulated videos at five different intensity levels, producing a total of 5000 manipulated videos. Besides, 1000 more robust manipulated videos are generated by adding random-type, random-level distortions to the 1000 raw manipulated videos. Moreover, in contrast to all the previous datasets, each sample of another 3000 manipulated videos in DeeperForensics-1.0 is subjected to a mixture of more than one distortion. The variability of perturbations improves the *diversity* of DeeperForensics-1.0 to better imitate the data distribution of real-world scenarios.

DeeperForensics-1.0 is a new *large-scale* dataset consisting of over 60,000 videos with 17.6 million frames for real-world face forgery detection. *High-quality* source videos and manipulated videos constitute two main contributions of the dataset. The *diversity* of perturbations applying to the manipulated videos ensures the robustness of DeeperForensics-1.0 to simulate real scenes. The whole dataset will be released, free to all research communities, for developing face forgery detection and more general human face related research.

3.4. User Study

To examine the quality of our DeeperForensics-1.0 dataset, we engage 100 professional participants, most of whom specialize in computer vision research. We believe these participants are qualified and well-trained in assessing realness of tempered videos. The user study is conducted on DeeperForensics-1.0 and six former datasets, *i.e.*, UADFV [49], DeepFake-TIMIT [27], Celeb-DF [30], FaceForensics++ [37], Deep Fake Detection [9], DFDC [14]. We randomly select 30 video clips from each of these datasets and prepare a platform for the participants to evaluate their realness. Similar to the user study of [23], the participants are asked to provide their feedbacks to the statement “The video clip looks real.” and give scores at five levels (1-clearly disagree, 2-weakly disagree, 3-borderline, 4-weakly agree, 5-clearly agree. We assume that users who give a score of 4 or 5 think the video is “real”). The user study results are shown in Table 3. The quality of our dataset is appreciated by most of the participants. Compared to the previous datasets, DeeperForensics-1.0 achieves the highest realism rating. Although Celeb-DF [30] also gets very high realness scores, the scale of our dataset is much larger.

4. Video Forgery Detection Benchmark

Dataset split. In our benchmark, we exploit 1000 raw manipulated videos in Section 3.3 and 1000 YouTube videos from FaceForensics++ [37] as our *standard* set. The videos

Table 3: The percentage of user study ratings for UADFV, DeepFake-TIMIT, Celeb-DF, FaceForensics++, Deep Fake Detection, DFDC, and DeeperForensics-1.0 dataset. A higher score means the users think the videos are more realistic.

Dataset	1	2	3	4	5	“real”
UADFV [49]	29.2	36.0	20.7	8.9	5.2	14.1%
DeepFake-TIMIT [27]	31.4	31.4	24.8	9.6	2.7	12.3%
Celeb-DF [30]	5.6	14.8	18.6	24.2	36.9	61.0%
FaceForensics++ [37]	46.8	31.4	13.4	4.4	4.0	8.4%
Deep Fake Detection [9]	26.0	28.0	24.1	11.5	10.3	21.9%
DFDC [14]	25.4	29.7	22.0	11.9	11.1	23.0%
DeeperForensics-1.0 (Ours)	4.3	8.9	22.6	29.8	34.3	64.1%

are split into training, validation, and test set with a ratio of 7 : 1 : 2. The identities of the swapped faces may be duplicated because the faces of 100 invited actors are swapped onto 1000 driving videos. To avoid data leak, we randomly choose unrepeated 70, 10 and 20 identities, and group all the videos according to the identities. Similar to [37], the test and training sets share a close distribution in our *standard* set.

Other experiments in our benchmark are different variants of the standard set. They share the same 1000 driving videos with the standard set. We will detail the variants in Section 4.2. For a fair comparison, all the experiments are conducted in the same split setting.

Hidden test set. For real-world scenarios, some experiments conducted in previous works [30, 37] may not perform a convincing evaluation due to the huge biases caused by a close distribution between the training and the test sets. The aforementioned standard set has the same setting with these works. As a result, strong detection baselines obtain very high accuracy on the standard test set as demonstrated in Section 4.2. However, the ultimate goal of the face forensics dataset is to help detect forgery in real scenes. Even if the accuracy on the standard test set is high, the models may easily fail in real-world scenarios.

We argue that the test set of *real-world* face forgery detection *should not* share a close distribution with the training set. What we need is a test set that better simulates the real-world setting. We call it “hidden” test set. To better imitate fake videos in the real scene, the hidden test set should satisfy three factors: 1) *Diverse distortions*. Different perturbations should be taken into consideration. 2) *Multiple sources*. Fake videos in-the-wild should be manipulated by different unknown methods. 3) *High quality*. Fake videos that are threatening should have high quality to fool human eyes.

Thus, in our initial benchmark, we introduce a challenging hidden test set with 400 carefully selected videos. First, we collect fake videos generated by several unknown face swapping methods to ensure multiple sources. Then, we obscure all selected videos multiple times with diverse hidden distortions that are commonly seen in real scenes. Finally, we only select videos that can fool at least 50 out of 100 human observers in a user study. The ground truth labels

Table 4: The binary detection accuracy of the baselines on the hidden test set when trained on four manipulated methods in FaceForensics++ (FF++): DeepFakes (DF), Face2Face (F2F), FaceSwap (FS), NeuralTextures (NT), and DeeperForensics-1.0 standard training set without distortions.

Train Test (acc)	FF++ DF hidden	FF++ F2F hidden	FF++ FS hidden	FF++ NT hidden	DeeperForensics-1.0 hidden
C3D [43]	57.50	57.75	52.13	58.25	74.75
TSN [45]	57.63	57.25	53.50	57.38	77.00
I3D [11]	56.63	58.38	54.63	63.63	79.25
ResNet+LSTM [17, 19]	57.38	56.13	54.88	59.50	78.25
XceptionNet [12]	57.38	58.75	54.75	57.38	77.00

are hidden and are used on our host server to evaluate the accuracy of detection models. Besides, the hidden test set will be enlarged constantly to get future versions along with development of Deepfakes technology. Fake videos manipulated by future face swapping methods will be included as long as they can pass the human test supported by us.

4.1. Baselines

Existing studies [30, 37] primarily provide image-level face forgery detection benchmark. However, fake videos in-the-wild are much more menacing than manipulated images. We propose to conduct evaluation mainly based on video classification methods for two reasons. First, image-level face forgery detection methods do not consider any temporal information – an important cue for video-based tasks. Second, image-level face forgery detection methods have been widely studied. We only choose one image-level method XceptionNet [12], which achieves the best performance in [37], as one part of our benchmark for reference. The other four video-based baselines are C3D [43], TSN [45], I3D [11], and ResNet+LSTM [17, 19], which have achieved promising results in video classification tasks. Details of all the baselines will be introduced in our *Appendix*.

4.2. Results and Analysis

Owing to the goal of detecting fakes in real-world scenarios, we mainly explore how common distortions appearing in real scenes affect the model performance. Accuracies of face forgery detection on the standard test set and the introduced hidden test set are evaluated under various settings.

Evaluation of effectiveness of DeeperForensics-1.0. For a fair comparison, we evaluate DeeperForensics-1.0 and the state-of-the-art FaceForensics++ [37] dataset because they use the same driving videos. In this setting, we use 1000 raw manipulated videos without distortions in the standard set of DeeperForensics-1.0. For FaceForensics++, the same split is applied to its *four* subsets. All the models are tested on the hidden test set (see Table 4).

The baselines trained on the standard training set of DeeperForensics-1.0 achieve much better performance on the hidden test set than all the *four* subsets of FaceForensics++. This proves the higher *quality* of DeeperForensics-

Table 5: The binary detection accuracy of the baselines when trained and tested on DeeperForensics-1.0 dataset with different distortion perturbations. We analyze different training and testing settings on the standard set without distortions (std), the standard set with single-level distortions (std/sing), and the standard set with random-level distortions (std/rand).

Train Test (acc)	std std	std std/sing	std std/rand	std/sing std/sing	std/rand std/rand	std/sing std/sing	std/rand std/rand
C3D [43]	98.50	87.63	92.38	95.38	96.63	96.75	94.00
TSN [45]	99.25	91.50	95.00	98.25	98.88	98.12	99.12
I3D [11]	100.00	90.75	96.88	99.50	99.63	99.63	98.00
ResNet+LSTM [17, 19]	100.00	90.63	97.13	100.00	98.63	100.00	97.25
XceptionNet [12]	100.00	88.38	94.75	99.63	99.63	99.75	99.00

1.0 over prior works, making it more useful for real-world face forgery detection. In Table 4, I3D [11] obtains the best performance on the hidden test set when trained on the standard training set. We conjecture that the temporal discontinuity of fake videos leads to higher accuracy by this video-level forgery detection method.

Evaluation of dataset perturbations. We study the effect of perturbations towards the forgery detection model performance. In contrast to prior work [37], we try to evaluate the baseline accuracies when applying different distortions to the training and the test sets, in order to explore the function of perturbations in face forensics dataset.

In this setting, we conduct all the experiments on DeeperForensics-1.0 dataset with high diversity of perturbations. We use 1000 manipulated videos in the standard set (std), 1000 manipulated videos with single-level (level-5), random-type distortions (std/sing), 1000 manipulated videos with random-level, random-type distortions (std/rand). The data split is the same as the standard set with a ratio of 7 : 1 : 2.

In Column 2 of Table 5, we find the accuracy is nearly 100% when the models are trained and tested on the standard set. This is reasonable because the strong baselines perform very well in a clean dataset with the same distribution. In Columns 3 and 4, the accuracy decrease compared to Column 2, when we choose std/sing and std/rand as the test set. Most of the video-level methods except C3D [43] are more robust to perturbations on test set than XceptionNet [12]. This setting is very common because different distributions of the training and the test sets lead to decrease in model accuracies. Hence, the lack of perturbations in the face forensics dataset cutbacks the model performance for real-world face forgery detection with even more complex data distribution.

When we apply corresponding distortions to the training and test set, the accuracy will increase compared to Column 3 and 4 (see Column 5 and 6 in Table 5). However, this setting is impractical because the distributions of the training and test sets are still the same. We should augment the test set to better simulate the real-world distribution. Thus, some evaluation settings in previous works [30, 37] are unreasonable. If we swap the training set and the test

Table 6: The binary detection accuracy of the baselines on the hidden test set when trained on DeeperForensics-1.0 dataset with the standard set without distortions (std), combination of std and the standard set with single-level distortions (std+std/sing), combination of std and the standard set with random-level distortions (std+std/rand), combination of std and the standard set with the mixed distortions(std+std/mix).

Train Test (acc)	std hidden	std+std/sing hidden	std+std/rand hidden	std+std/mix hidden
C3D [43]	74.75	78.25	78.13	78.88
TSN [45]	77.00	78.75	79.50	79.50
I3D [11]	79.25	80.13	80.13	80.13
ResNet+LSTM [17, 19]	78.25	80.25	79.50	80.25
XceptionNet [12]	77.00	79.75	79.75	79.88

set of std/sing and std/rand to further randomize the condition, results shown in Column 7 and 8 indicate that the high accuracy keep unchanged. This evaluation setting shows the possibility that with the same generation method, exerting appropriate distortions to the training set can make face forgery detection models more robust to real-world perturbations.

Evaluation of variants of training set for real-world face forgery detection. We have conducted several experiments for evaluations of possible perturbations. Nevertheless, the case is more complex in real scenes because information about the fake videos is not available. The video may be subjected to more than one type and diverse levels of distortions. In addition to distortions, the method manipulating the faces is unknown.

From the evaluation of perturbations, we find the possibility of augmenting the training set to improve detection model performance. Thus, we further evaluate baseline performance on the hidden test set by devising some variants of the training set. We perform experiments on DeeperForensics-1.0. In this setting, other than std, std/sing, and std/rand, we use 1000 more manipulated videos, each of which is subjected to a mixture of three random-level, random-type distortions (std/mix). We combine std with std/sing, std/rand, and std/mix, respectively, to form three new training sets (with the same data split as the former settings).

Column 2 in Table 6 shows the low accuracy when the models trained on std and tested on the hidden test set (the same as Column 6 in Table 4). Columns 3 and 4 indicate the accuracy of all the baseline models increase when trained on std+std/sing and std+std/rand. The accuracy of I3D [11] and ResNet+LSTM [17, 19], are over 80% in some cases. In a more complex setting, when the models are trained on std+std/mix, Column 5 shows the accuracy of all the detection baselines further increase.

It proves that designing suitable training set variants has the potential to help increase the face forgery detection accuracy, and applying various distortions to ensure the *diversity* of DeeperForensics-1.0 is necessary. In addition, compared to image-level method, video-level face forgery

detection methods have more potential capabilities to crack real-world fake videos as shown in Table 6.

Despite the accuracy on the challenging hidden test set is still not very high, we provide two initial directions for future real-world face forgery detection research: 1) Improving the source data collection and generation method to ensure the *quality* of the training set; 2) Augmenting the training set by various distortions to ensure its *diversity*. We welcome researchers to make our benchmark more comprehensive.

5. Discussion

In this work, we propose a new large-scale dataset named DeeperForensics-1.0 to facilitate the research of face forgery detection towards *real-world* scenarios. We make several efforts to ensure the good quality, large scale, and high diversity of the proposed dataset. Based on the dataset, we further benchmark the results of existing representative forgery detection methods, offering insights into the current status and future improvisation strategy in face forgery detection. Several topics are considered as the future works. 1) We will keep on collecting the identities of source and target videos to further expand DeeperForensics gradually over time. 2) We will ask interested researchers for any results of future video falsification methods to enlarge our hidden test set, as long as the fakes can pass the human test supported by us. 3) A better evaluation metric for face forgery detection methods is also an interesting research topic.

Acknowledgments. We gratefully acknowledge the exceptional help from Hao Zhu and Keqiang Sun for source data collection and coordination.

References

- [1] Deepfacelab. <https://github.com/iperov/DeepFaceLab/>. Accessed: 2019-08-20. **1**
- [2] Deepfakes. <https://github.com/deepfakes/faceswap/>. Accessed: 2019-08-16. **1, 2, 14**
- [3] Faceswap. <https://github.com/MarekKowalski/FaceSwap/>. Accessed: 2019-08-18. **2**
- [4] faceswap-gan. <https://github.com/shaoanlu/faceswap-GAN/>. Accessed: 2019-08-16. **1, 2, 14**
- [5] Fakeapp. <https://www.fakeapp.com/>. Accessed: 2019-07-25. **2**
- [6] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *WIFS*, 2018. **3**
- [7] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra. A sift-based forensic method for copy-move attack detection and transformation recovery. *TIFS*, 6:1099–1110, 2011. **2**
- [8] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *IH & MMSEC*, 2016. **3**

- [9] Google AI Blog. Contributing data to deepfake detection research. <https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html>. Accessed: 2019-09-25. 1, 2, 3, 4, 8
- [10] Chen Cao, Yanlin Weng, Shun Zhou, Yiyong Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *TVCG*, 20:413–425, 2013. 2, 4, 15, 17
- [11] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 3, 9, 10, 15
- [12] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 3, 9, 10, 15
- [13] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In *IH & MMSEC*, 2017. 3
- [14] Brian Dolhansky, Russ Howes, Ben Pfau, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint*, arXiv:1910.08854, 2019. 1, 2, 3, 4, 8
- [15] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *TIFS*, 7:868–882, 2012. 3
- [16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 14
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 9, 10, 15
- [18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 14
- [19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 1997. 3, 9, 10, 15
- [20] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 7, 13, 14
- [21] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 7, 14
- [22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 15
- [23] Hyeonwoo Kim, Pablo Carrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM TOG*, 37:163, 2018. 8, 14, 16
- [24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980, 2014. 14
- [25] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *NeurIPS*, 2014. 6
- [26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*, arXiv:1312.6114, 2013. 5, 6, 13
- [27] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint*, arXiv:1812.08685, 2018. 2, 3, 8
- [28] Paweł Korus and Jiwu Huang. Multi-scale analysis strategies in prnu-based tampering localization. *TIFS*, 12:809–824, 2016. 2
- [29] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint*, arXiv:1811.00656, 2018. 3
- [30] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A new dataset for deepfake forensics. *arXiv preprint*, 2019. 2, 3, 8, 9
- [31] Kwan-Yee Lin and Guangxiang Wang. Hallucinated-iqa: No-reference image quality assessment via adversarial learning. In *CVPR*, 2018. 8
- [32] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *WACVW*, 2019. 3
- [33] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hour-glass networks for human pose estimation. In *ECCV*, 2016. 5, 14
- [34] Huy H Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. *arXiv preprint*, arXiv:1906.06876, 2019. 3
- [35] Nikolay Ponomarenko, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, et al. Image database tid2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication*, 30:57–77, 2015. 8
- [36] Nicolas Rahmouni, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Distinguishing computer graphics from natural images using convolution neural networks. In *WIFS*, 2017. 3
- [37] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. *arXiv preprint*, arXiv:1901.08971, 2019. 1, 2, 3, 7, 8, 9, 14, 15
- [38] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 14
- [39] Conrad Sanderson. The vidtimit database. Technical report, IDIAP, 2002. 2
- [40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, arXiv:1409.1556, 2014. 13
- [41] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *arXiv preprint*, arXiv:1904.12356, 2019. 2
- [42] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016. 2
- [43] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 3, 9, 10, 15

- [44] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint*, arXiv:1607.08022, 2016. 13
- [45] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 3, 9, 10, 15
- [46] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint*, arXiv:1808.06601, 2018. 7
- [47] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, 2018. 4
- [48] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. Reenactgan: Learning to reenact faces via boundary transfer. In *ECCV*, 2018. 14
- [49] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP*, pages 8261–8265, 2019. 2, 3, 8
- [50] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Detecting image splicing in the wild (web). In *ICMEW*, 2015. 2
- [51] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Two-stream neural networks for tampered face detection. In *CVPRW*, 2017. 3

Appendix

A. Derivation

The core equations of DF-VAE are Eq. 2, Eq. 3, and Eq. 4. We will provide the detailed mathematical derivations in this section.

Derivation of Eq. 2 and Eq. 3:

$$\begin{aligned}
& \log p_{\theta}(x_t | \hat{x}_t, x') \\
&= \mathbb{E}_{q_{\phi}(z_x | x_t, \hat{x}_t, x')} (\log p_{\theta}(x_t | \hat{x}_t, x')) \\
&= \mathbb{E}_{q_{\phi}(z_x | x_t, \hat{x}_t, x')} \left[\log \frac{p_{\theta}(x_t, z_x | \hat{x}_t, x')}{p_{\theta}(z_x | x_t, \hat{x}_t, x')} \right] \\
&= \mathbb{E}_{q_{\phi}(z_x | x_t, \hat{x}_t, x')} \left[\log \frac{q_{\phi}(z_x | x_t, \hat{x}_t, x')}{p_{\theta}(z_x | x_t, \hat{x}_t, x')} \cdot \frac{p_{\theta}(x_t, z_x | \hat{x}_t, x')}{q_{\phi}(z_x | x_t, \hat{x}_t, x')} \right] \\
&= \int q_{\phi}(z_x | x_t, \hat{x}_t, x') \left[\log \frac{q_{\phi}(z_x | x_t, \hat{x}_t, x')}{p_{\theta}(z_x | x_t, \hat{x}_t, x')} + \log \frac{p_{\theta}(x_t, z_x | \hat{x}_t, x')}{q_{\phi}(z_x | x_t, \hat{x}_t, x')} \right] dz_x \\
&= D_{KL}(q_{\phi}(z_x | x_t, \hat{x}_t, x') \| p_{\theta}(z_x | x_t, \hat{x}_t, x')) + L(\theta, \phi; x_t, \hat{x}_t, x'),
\end{aligned}$$

where

$$\begin{aligned}
L(\theta, \phi; x_t, \hat{x}_t, x') &= \mathbb{E}_{q_{\phi}(z_x | x_t, \hat{x}_t, x')} \left[\log \frac{p_{\theta}(x_t, z_x | \hat{x}_t, x')}{q_{\phi}(z_x | x_t, \hat{x}_t, x')} \right], \\
D_{KL}(q_{\phi}(z_x | x_t, \hat{x}_t, x') \| p_{\theta}(z_x | x_t, \hat{x}_t, x')) &\geq 0.
\end{aligned}$$

Derivation of Eq. 4:

$$\begin{aligned}
& L(\theta, \phi; x_t, \hat{x}_t, x') \\
&= \mathbb{E}_{q_{\phi}(z_x | x_t, \hat{x}_t, x')} \left[\log \frac{p_{\theta}(x_t, z_x | \hat{x}_t, x')}{q_{\phi}(z_x | x_t, \hat{x}_t, x')} \right] \\
&= \mathbb{E}_{q_{\phi}(z_x | x_t, \hat{x}_t, x')} \left[\log \frac{p_{\theta}(x_t | z_x, \hat{x}_t, x') p_{\theta}(z_x | \hat{x}_t, x')}{q_{\phi}(z_x | x_t, \hat{x}_t, x')} \right] \\
&= \int q_{\phi}(z_x | x_t, \hat{x}_t, x') \left[-\log \frac{q_{\phi}(z_x | x_t, \hat{x}_t, x')}{p_{\theta}(z_x | \hat{x}_t, x')} + \log p_{\theta}(x_t | z_x, \hat{x}_t, x') \right] dz_x \\
&= -D_{KL}(q_{\phi}(z_x | x_t, \hat{x}_t, x') \| p_{\theta}(z_x | \hat{x}_t, x')) \\
&\quad + \mathbb{E}_{q_{\phi}(z_x | x_t, \hat{x}_t, x')} [\log p_{\theta}(x_t | z_x, \hat{x}_t, x')].
\end{aligned}$$

B. Objective

Reconstruction loss. In the reconstruction, the source face and target face share the same forms of loss functions. The reconstruction loss of the source face, L_{recon_x} , can be written as:

$$L_{recon_x} = \lambda_{r_1} L_{pixel}(\tilde{x}, x) + \lambda_{r_2} L_{ssim}(\tilde{x}, x). \quad (8)$$

L_{pixel} indicates pixel loss. It calculates the Mean Absolute Error (MAE) after reconstruction, which can be written as:

$$L_{pixel}(\tilde{x}, x) = \frac{1}{CHW} \|\tilde{x} - x\|_1. \quad (9)$$

L_{ssim} denotes ssim loss. It computes the Structural Similarity (SSIM) of the reconstructed face and the original face, which has the form of:

$$L_{ssim}(\tilde{x}, x) = \frac{(2\mu_{\tilde{x}}\mu_x + C_1)(2\sigma_{\tilde{x}x} + C_2)}{(\mu_{\tilde{x}}^2 + \mu_x^2 + C_1)(\sigma_{\tilde{x}}^2 + \sigma_x^2 + C_2)}. \quad (10)$$

λ_{r_1} and λ_{r_2} are two hyperparameters that control the weights of two parts of the reconstruction loss. For the target face, we have the similar form of reconstruction loss:

$$L_{recon_y} = \lambda_{r_1} L_{pixel}(\tilde{y}, y) + \lambda_{r_2} L_{ssim}(\tilde{y}, y). \quad (11)$$

Thus, the full reconstruction loss can be written as:

$$L_{recon} = L_{recon_x} + L_{recon_y}. \quad (12)$$

KL loss. Since DF-VAE is a new conditional variational auto-encoder, reparameterization trick is utilized to make the sampling operation differentiable by an auxiliary variable with independent marginal. We use the typical KL loss in [26] with the form of:

$$L_{KL}(q_{\phi}(z), p_{\theta}(z)) = \frac{1}{2} \sum_{j=1}^J (1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2), \quad (13)$$

where J is the dimensionality of the latent prior z , μ_j and σ_j are the j -th element of variational mean and s.d. vectors, respectively.

MAdaIN loss. The MAdaIN module is jointly trained with the disentangled module in an end-to-end manner. We apply MAdaIN loss for this module, in a similar form as described in [20]. We use the VGG-19 [40] to compute MAdaIN loss to train Decoder D_{δ} :

$$L_{MAdaIN} = L_c + \lambda_{ma} L_s. \quad (14)$$

L_c denotes the content loss, which is the Euclidean distance between the target features and the features of the swapped face. L_c has the form of:

$$L_c = \|o - c\|_2, \quad (15)$$

where $o = m_t^b \cdot \bar{d}_t$, $c = m_t^b \cdot d_t$. m_t^b is the blurred mask described in Section 3.2.

L_s represents the style loss, which matches the mean and standard deviation of the style features. Like [20], we match the IN [44] statistics instead of using Gram matrix loss which can produce similar results. L_s can be written as:

$$\begin{aligned}
L_s &= \sum_{i=1}^L \|\mu(\Phi_i(o)) - \mu(\Phi_i(s))\|_2 \\
&\quad + \sum_{i=1}^L \|\sigma(\Phi_i(o)) - \sigma(\Phi_i(s))\|_2,
\end{aligned} \quad (16)$$

where $o = m_t^b \cdot \bar{d}_t$, $s = m_t^b \cdot y_t$. m_t^b is the blurred mask described in Section 3.2. Φ_i denotes the layer used in VGG-19 [40]. Similar to [20], we use relu1_1, relu2_1, relu3_1, relu4_1 layers with equal weights.

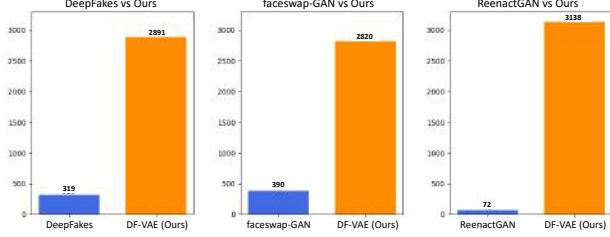


Figure 9: Results of user study comparing methods. The bar charts show the number of users who give preference in each compared pair of manipulated videos.

Table 7: The FID and IS scores of DeepFakes [2], faceswap-GAN [4], ReenactGAN [48], and DF-VAE (Ours).

Method	FID	IS
DeepFakes [2]	25.771	1.711
faceswap-GAN [4]	24.718	1.685
ReenactGAN [48]	26.325	1.690
DF-VAE (Ours)	22.097	1.714

λ_{ma} is the weight of style loss to balance two parts of MAdaIN loss.

Temporal loss. We have given a detailed introduction of temporal consistency constraint in Section 3.2. The temporal loss has the form of Eq. 7. We will not repeat it here.

Total objective. DF-VAE is an end-to-end many-to-many face swapping framework. We jointly train all parts of the networks. The problem can be described as the optimization of the following total objective:

$$L_{total} = \lambda_1 L_{recon} + \lambda_2 L_{KL} + \lambda_3 L_{MAdaIN} + \lambda_4 L_{temporal}, \quad (17)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the weight hyperparameters of four types of loss functions introduced above.

C. Implementation Details

The whole DF-VAE framework is end-to-end. We use the pretrained stacked hourglass networks [33] to extract landmarks. The numbers of stacks and blocks are set to 4 and 1, respectively. We exploit FlowNet 2.0 network [21] to estimate optical flows. The typical AdaIN network [20] is applied to our style matching and fusion module. The learning rate is set to 0.00005 for all parts of DF-VAE. We utilize Adam [24] and set $\beta_1 = 0.5, \beta_2 = 0.999$. All the experiments are conducted on NVIDIA Tesla V100 GPUs.

D. User Study of Methods

In addition to user study based on datasets to examine the quality of DeeperForensics-1.0 dataset, we also carry out a user study to compare DF-VAE with state-of-the-art face manipulation methods. We will present the user study of methods in this section.

Baselines. We choose three learning-based open-source methods as our baselines: DeepFakes [2], faceswap-GAN [4], and ReenactGAN [48]. These three methods are representative, which are based on different architectures. DeepFakes [2] is a well-known method based on Auto-Encoders (AE). It uses a shared encoder and two separated decoders to perform face swapping. faceswap-GAN [4] is based on Generative Adversarial Networks (GAN) [16], which has a similar structure as DeepFakes [2] but also uses a paired discriminators to improve face swapping quality. ReenactGAN [48] makes a boundary latent space assumption and uses a transformer to adapt the boundary of source face to that of target face. As a result, ReenactGAN can perform many-to-one face reenactment. After getting the reenacted faces, we use our carefully designed fusion method to obtain the swapped faces. For a fair comparison, DF-VAE utilizes the same fusion method when compared to ReenactGAN [48].

Results. We randomly choose 30 real videos from DeeperForensics-1.0 as the source videos and 30 real videos from FaceForensics++ [37] as the target videos. Thus, each method generates 30 fake videos. Same as the user study based on datasets, we conduct the user study based on methods among 100 professional participants who specialize in computer vision research. Because there are corresponding fake videos, we let the users directly choose their preferred fake videos between those generated by other methods and those generated by DF-VAE. Finally, we got 3210 answers for each compared pair. The results are shown in Figure 9. We can see that DF-VAE shows an impressive advantage over the baselines, underscoring the *high quality* of DF-VAE-generated fake videos.

E. Quantitative Evaluation Metrics

Frechet Inception Distance (FID) [18] is a widely exploited metric for generative models. FID evaluates the *similarity* of distribution between the generated images and the real images. FID correlates well with the visual quality of the generated samples. A lower value of FID means a better quality.

Inception Score (IS) [38] is an early and somewhat widely adopted objective evaluation metric for generated images. IS evaluates two aspects of generation quality: *articulation* and *diversity*. A higher value of IS means a better quality.

Table 7 shows the FID and IS scores of our method compared to other methods. DF-VAE outperforms all the three baselines in quantitative evaluations by FID and IS.

F. Ablation Study

Ablation study of temporal loss. Since the swapped faces do not have the ground truth, we evaluate the effectiveness of temporal consistency constraint, *i.e.*, temporal loss, in a self-reenactment setting. Similar to [23], we quantify the re-rendering error by Euclidean distance of per pixel in RGB

channels ([0, 255]). Visualized results are shown in Figure 10. Without the temporal loss, the re-rendering error is higher, hence demonstrating the effectiveness of temporal consistency constraint.

Ablation study of different components. We conduct further ablation studies *w.r.t.* different components of our DF-VAE framework under many-to-many face swapping setting (see Figure 11). The source and target faces are shown in Column 1 and Column 2. In Column 3, our full method, DF-VAE, shows high-fidelity face swapping results. In Column 4, style mismatch problems are very obvious if we remove the MAdaIN module. If we remove the hourglass (structure extraction) module, the disentanglement of structure and appearance is not very thorough. The swapped face will be a mixture of multiple identities, as shown in Column 5. When we perform face swapping without constructing unpaired data in the same domain (see Column 6), the disentangled module will completely reconstruct the faces on the side of E_β , thus the disentanglement is not established at all. Therefore, the quality of face swapping will degrade if we remove any component in DF-VAE framework.

G. Details of Benchmark Baselines

We will elaborate on five baselines used in our face forgery detection benchmark in this section. Our benchmark contains four video-level face forgery detection methods, C3D [43], Temporal Segment Networks (TSN) [45], Inflated 3D ConvNet (I3D) [11], and ResNet+LSTM [17, 19]. One image-level detection method, XceptionNet [12], which achieves the best performance in FaceForensics++ [37], is evaluated as well.

- **C3D** [43] is a simple but effective method, which incorporates 3D convolution to capture the spatiotemporal feature of videos. It includes 8 convolutional, 5 max-pooling, and 2 fully connected layers. The size of the 3D convolutional kernels is $3 \times 3 \times 3$. When training C3D, the videos are divided into non-overlapped clips with 16-frames length, and the original face images are resized to 112×112 .
- **TSN** [45] is a 2D convolutional network, which splits the video into short segments and randomly selects a snippet from each segment as the input. The long-range temporal structure modeling is achieved by the fusion of the class scores corresponding to these snippets. In our experiment, we choose BN-Inception [22] as the backbone and only train our model with the RGB stream. The number of segments is set to 3 as default, and the original images are resized to 224×224 .

- **I3D** [11] is derived from Inception-V1 [22]. It inflates the 2D ConvNet by endowing the filters and pooling kernels with an additional temporal dimension. In the training, we use 64-frame snippets as the input, whose starting frames are randomly selected from the videos. The face images are resized to 224×224 .
- **ResNet+LSTM** [17, 19] is based on ResNet [17] architecture. As a 2D convolutional framework, ResNet [17] is used to extract spatial features (the output of the last convolutional layer) for each face image. In order to encode the temporal dependency between images, we place an LSTM [19] module with 512 hidden units after ResNet-50 [17] to aggregate the spatial features. An additional fully connected layer serves as the classifier. All the videos are downsampled with a ratio of 5, and the images are resized to 224×224 before feeding into the network. During training, the loss is the summation of the binary entropy on the output at all time steps, while only the output of the last frame is used for the final classification in inference.
- **XceptionNet** [12] is a depthwise-separable-convolution based CNN, which has been used in [37] for image-level face forgery detection. We exploit the same XceptionNet model as [37] but without freezing the weights of any layer during training. The face images are resized to 299×299 . In the test phase, the prediction is made by averaging classification scores of all frames within a video.

H. More Examples of Data Collection

In this section, we will show more examples of our extensive source video data collection (see Figure 12). Our high-quality collected data vary in identities, poses, expressions, emotions, lighting conditions, and 3DMM blend-shapes [10]. The source videos will also be released for further research.

I. Perturbations

We will also show some examples of perturbations in DeeperForensics-1.0. Seven types of perturbations and the mixture of two (Gaussian blur, JPEG compression) / three (Gaussian blur, JPEG compression, white Gaussian noise in color components) / four (Gaussian blur, JPEG compression, white Gaussian noise in color components, color saturation change) perturbations are shown in Figure 13. These perturbations are very common distortions existing in real life. The comprehensiveness of perturbations in DeeperForensics-1.0 ensures its *diversity* to better simulate fake videos in real-world scenarios.

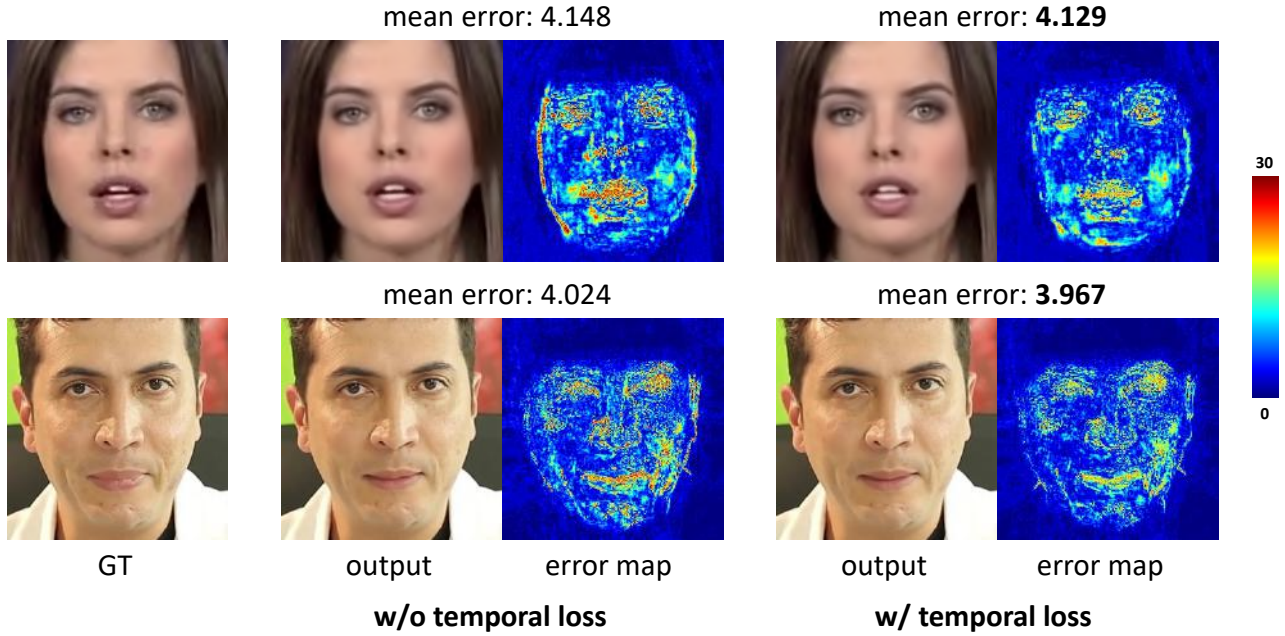


Figure 10: The quantitative evaluation of the effectiveness of temporal loss. Similar to [23], we use the re-rendering error in a self-reenactment setting, where the ground truth is known. The error maps show Euclidean distance of per pixel in RGB channels ([0, 255]). The mean errors are shown above the images. The corresponding color scale *w.r.t.* error values is shown on the right side of the images.

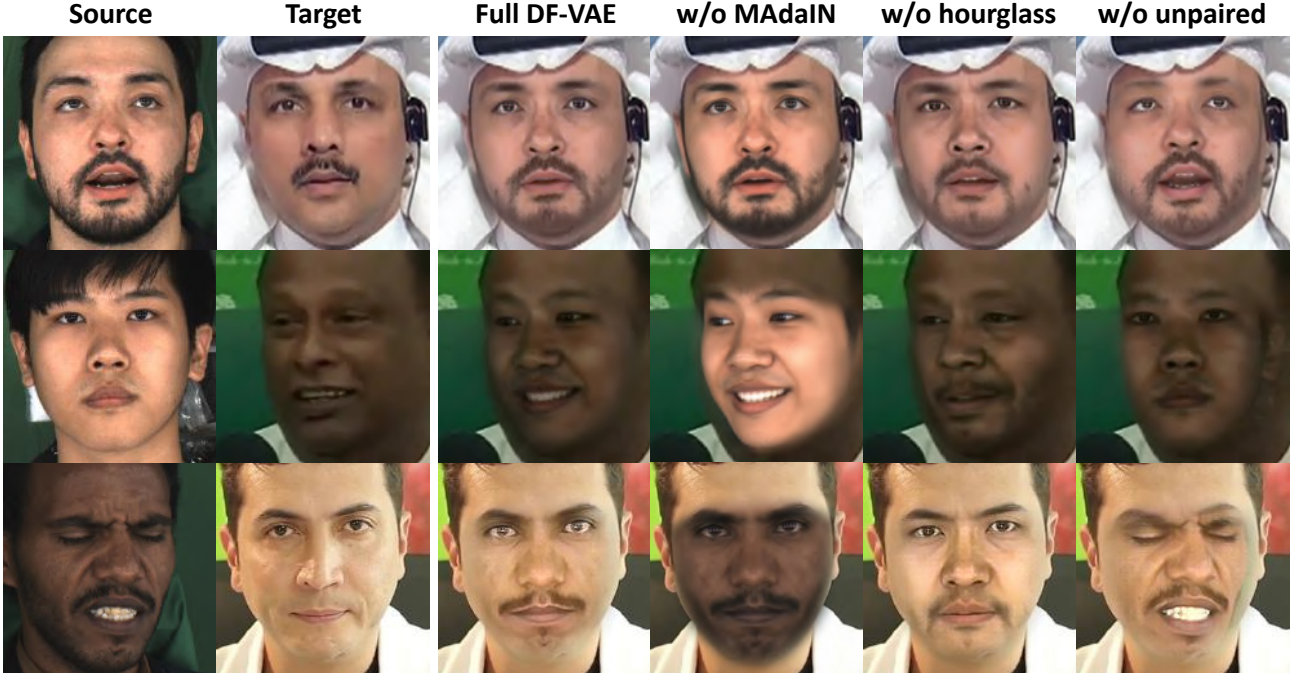


Figure 11: The ablation studies of different components in DF-VAE framework in the many-to-many face swapping setting. Column 1 and Column 2 show the source face and the target face, respectively. Column 3 shows the results of the full method. Column 4, 5, 6 show the results when removing MAdaln module, hourglass (structure extraction) module, and unpaired data construction, respectively.



Figure 12: More examples of the source video data collection. Our high-quality collected data vary in identities, poses, expressions, emotions, lighting conditions, and 3DMM blendshapes [10].

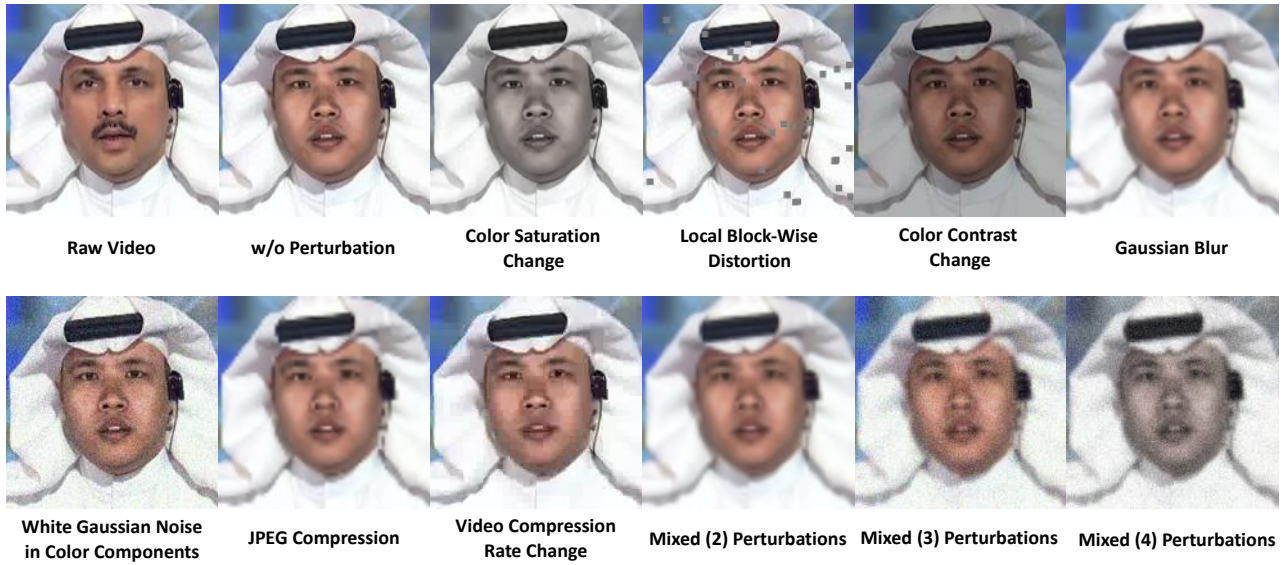


Figure 13: Seven types of perturbations and the mixture of two (Gaussian blur, JPEG compression) / three (Gaussian blur, JPEG compression, white Gaussian noise in color components) / four (Gaussian blur, JPEG compression, white Gaussian noise in color components, color saturation change) perturbations in DeeperForensics-1.0.