M2TR: Multi-modal Multi-scale Transformers for Deepfake Detection

Junke Wang, Zuxuan Wu, Jingjing Chen, Yu-Gang Jiang
Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University
{17300240009, zxwu, chenjingjing, ygj}@fudan.edu.cn

ABSTRACT

The widespread dissemination of forged images generated by Deepfake techniques has posed a serious threat to the trustworthiness of digital information. This demands effective approaches that can detect perceptually convincing Deepfakes generated by advanced manipulation techniques. Most existing approaches combat Deepfakes with deep neural networks by mapping the input image to a binary prediction without capturing the consistency among different pixels. In this paper, we aim to capture the subtle manipulation artifacts at different scales for Deepfake detection. We achieve this with transformer models, which have recently demonstrated superior performance in modeling dependencies between pixels for a variety of recognition tasks in computer vision. In particular, we introduce a **M**ulti-modal **M**ulti-scale **TR**ansformer (**M2TR**), which uses a multi-scale transformer that operates on patches of different sizes to detect the local inconsistency at different spatial levels. To improve the detection results and enhance the robustness of our method to image compression, M2TR also takes frequency information, which is further combined with RGB features using a cross modality fusion module. Developing and evaluating Deepfake detection methods requires large-scale datasets. However, we observe that samples in existing benchmarks contain severe artifacts and lack diversity. This motivates us to introduce a high-quality Deepfake dataset, SR-DF, which consists of 4,000 DeepFake videos generated by state-of-the-art face swapping and facial reenactment methods. On three Deepfake datasets, we conduct extensive experiments to verify the effectiveness of the proposed method, which outperforms state-of-the-art Deepfake detection methods.

1 INTRODUCTION

Recent years have witnessed the rapid development of Deepfake techniques [26, 29, 43, 52], which enable attackers to manipulate the facial area of an image and generate a forged image. As synthesized images are becoming more photo-realistic, it is extremely difficult to distinguish whether an image/video has been manipulated even for human eyes. At the same time, these forged images might be distributed on the Internet for malicious purposes, which could bring societal implications. The above challenges have driven the development of Deepfake forensics using deep neural networks [1, 5, 25, 32, 34, 40, 67]. Most existing approaches take as inputs a face region cropped out of an entire image and produce a binary real/fake prediction with deep CNN models. These methods capture artifacts from the face regions in a single scale with stacked convolutional operations. While decent detection results are achieved by stacked convolutions, they excel at modeling local information but fails to consider the relationships of pixels globally due to constrained receptive field.

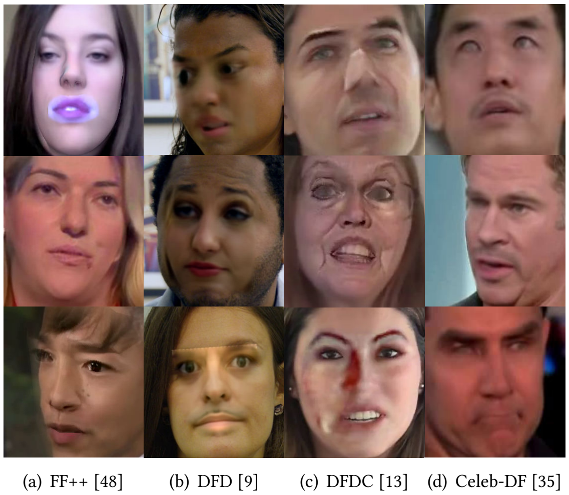
(a) FF++ [48] (b) DFD [9] (c) DFDC [13] (d) Celeb-DF [35]

Figure 1: Visual artifacts of forged images in existing datasets, including color mismatch (row 1 col 1, row 2 col 3, row 3 col 1, row 3 col 2, row 3 col 3) , shape distortion (row 1 col 3, row 2 col 1), visible boundaries (row 2 col 2), and facial blurring (row 1 col 2, row 4 col 1, row 4 col 2, row 4 col3).

We posit that relationships among pixels are particularly useful for Deepfake detection, since pixels in certain artifacts are clearly different from the remaining pixels in the image. On the other hand, we observe that forgery patterns vary in sizes. For instance, Figure 1 gives examples from popular Deepfake datasets. We can see that some forgery traces such as color mismatch occur in small regions (like the mouth corners), while other forgery signals such as visible boundaries that almost span the entire image (see row 3 col 2 in Figure 1). Therefore, how to effectively explore regions of different scales in images is extremely critical for Deepfake detection.

To address the above limitations, we explore transformers to model the relationships of pixels due to their strong capability of long-term dependency modeling for both natural language processing tasks [12, 46, 58] and computer vision tasks [2, 14, 68]. Unlike traditional transformers operating on a single-scale, we propose a multi-scale architecture to capture forged regions that potentially have different sizes. Furthermore, [15, 24, 44, 59, 65] suggest that the artifacts of forged images will be destroyed by perturbations such as JPEG compression, making them imperceptible in the RGB domain but can still be detected in the frequency domain. This motivates us to use frequency information as a complementary modality in order to reveal artifacts that are no longer perceptible in the RGB domain.

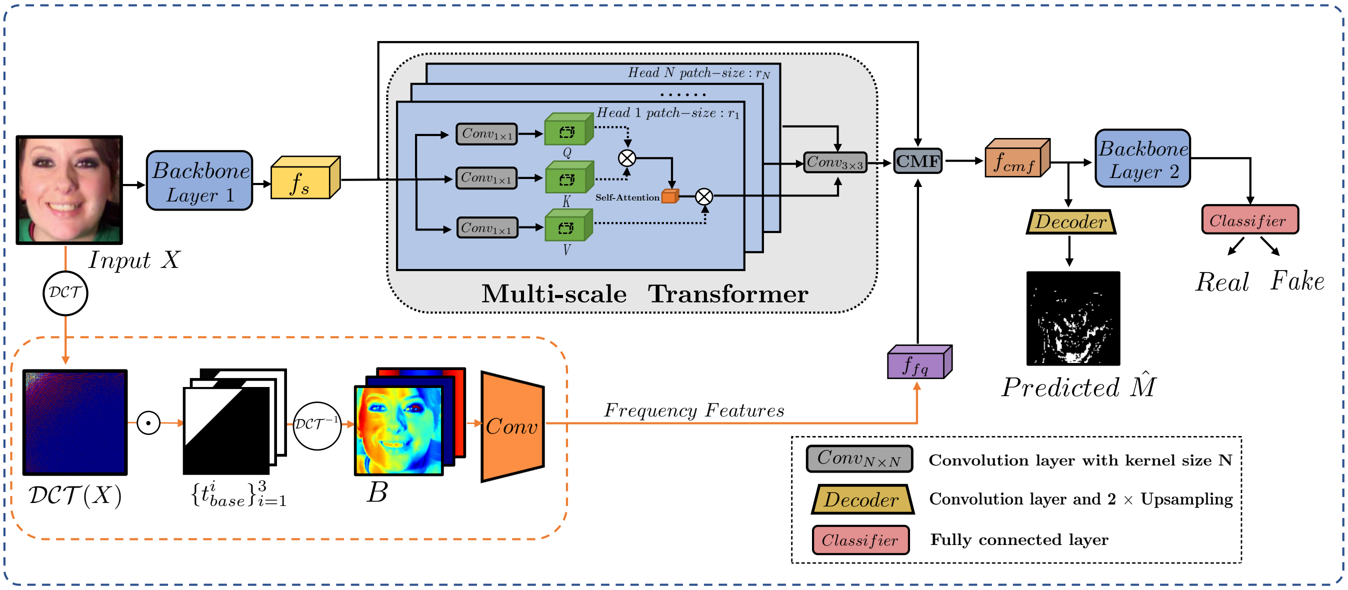

Figure 2: Overview of the proposed M2TR. The input is a suspicious face image (H x W x C), and the output includes both a forgery detection result and a predicted mask (H x W x 1), which locates the forgery regions.

To this end, we introduce M2TR, a Multi-modal Multi-scale Transformer, for Deepfake detection. M2TR is a multimodal framework, consisting of a Multi-scale Transformer (MT) module and a Cross Modality Fusion (CMF) module. In particular, M2TR first extracts features of an input image with a few convolutional layers. We then generate patches of different sizes from the feature map, which are used as inputs to different heads of the transformer. Similarities of spatial patches across different scales are calculated to capture the inconsistency among different regions at multiple scales. This benefits the discovery of forgery artifacts, since certain subtle forgery clues, *e.g.*, blurring and color inconsistency, are often times hidden in small local patches. The outputs from the multi-scale transformer are further augmented with frequency information to derive fused feature representations using a cross modality fusion module. Finally, the integrated features are used as inputs to several convolutional layers to generate prediction results. In addition to binary classification, we also predict the manipulated regions of the face image in a multi-task manner. The rationale behind is that binary classification tends to result in easily overfitted models. Therefore, we use face masks as additional supervisory signals to mitigate overfitting.

The availability of large-scale training data is an essential factor in the development of Deepfake detection methods. Existing Deepfake datasets include the UADFV dataset [63], the DeepFake-TIMIT dataset (DF-TIMIT) [28], the FaceForensics++ dataset (FF++) [48], the Google DeepFake detection dataset (DFD) [9], the FaceBook DeepFake detection challenge (DFDC) dataset [13], the WildDeepfake dataset [69], and the Celeb-DF dataset [35]. However, the quality of visual samples in current Deepfake datasets is limited, containing clear artifacts (see Figure 1) like color mismatch, shape distortion, visible boundaries, and facial blurring. Therefore, there is still a huge gap between the images in existing datasets and forged images in the wild which are circulated on the Internet. Although the visual quality of Celeb-DF [35] is relatively high compared to others, they use only one face swapping method to generate forged images, lacking sample diversity. In addition, there are no unbiased and comprehensive evaluation metrics to measure the quality of Deepfake datasets, which is not conducive to the development of subsequent Deepfake research.

In this paper, we present a large-scale and high-quality Deepfake dataset, **S**wapping and **R**eenactment **D**eep**F**ake (**SR-DF**) dataset, which is generated using the state-of-the-art face swapping and facial reenactment methods for the development and evaluation of Deepfake detection methods. We visualize in Figure 4 the sampled forged faces in the proposed SR-DF dataset. Besides, we propose a set of evaluation criteria to measure the quality of Deepfake datasets from different perspectives. We hope the release of SR-DF dataset and the evaluation systems will benefit the future research of Deepfake detection.

Our work makes the following key contributions:

- We propose a Multi-modal Multi-scale Transformer (**M2TR**) for Deepfake forensics, which uses a multi-scale transformer to detect local inconsistency at different scales and leverages frequency features to improve the robustness of detection. Extensive experiments demonstrate that our method achieves state-of-the-art detection performance on different datasets.
- We introduce a large-scale and challenging Deepfake dataset **SR-DF**, which is generated with state-of-the-art face swapping and facial reenactment methods.
- We construct the most comprehensive evaluation system and demonstrate that SR-DF dataset is well-suited for the training Deepfake detection methods due to its visual quality and diversity.

2 RELATED WORK

Deepfake Generation Existing Deepfake generation methods can be divided into two categories: face swapping and facial reenactment. Face swapping methods [8, 10, 18, 38] replace the face of a source person in an image with the face of a target person. Most of these methods consist of three steps: segmenting the facial regions from the target image, manipulating the expression to the source person, and blending with the background. Facial reenactment methods [54, 55, 61], on the other hand, transfer the face expressions of the source person to the target person while preserving the identity information of the source person. Specially, [23] use a landmark decoder to model expression motions and a content encoder to extract identity information, thus handling the task by feature disentanglement.

Deepfake Detection To mitigate the security threat brought by Deepfakes, a variety of methods have been proposed for Deepfake detection. [67] uses a two-stream architecture to capture facial manipulation clues and patch inconsistency separately, while [40] simultaneously identifies forged faces and locate the manipulated regions with multi-task learning. Recently, [32] proposes to detect the blending boundaries based on an observation that the step of blending a forged face into the background is commonly used by most existing face manipulation methods. [60] extracts features from the face image using a CNN model, which are then fed to a traditional single-scale transformer for forgery detection. However, most of them only focus on the features in RGB domain, thus failing to detect forged images which are manipulated subtly in the color-space. Instead, [5] uses RGB and frequency information collaboratively to extract comprehensive forgery features. In this paper, we use a multi-scale transformer to capture local inconsistency at different scales for forgery detection, and additionally introduce frequency modality to improve the robustness of our method to various image compression algorithms.

Visual Transformers Transformers [58] have demonstrated impressive performance for natural language processing tasks due to their strong ability in modeling long-range context information. Recently, researchers have demonstrated remarkable interests in using the transformer for a variety of computer vision tasks. In particular, Vision Transformer (ViT) reshapes the image into a sequence of flattened patches and input them to the transformer encoder for image classification [14]. DETR uses a common CNN to extract semantic features from the input image, which are then input to a transformer-based encoder-decoder architecture for object detection [2]. In this paper, we adopt a multi-scale transformer which integrates multi-scale information for Deepfake detection.

3 APPROACH

Our goal is to detect the subtle forgery artifacts that are hidden in the inconsistency of local patches and improve the robustness to image compression with frequency features. In this section, we introduce the Multi-modal Multi-scale Transformer (M2TR) for Deepfake detection, which consists of a multi-scale transformer in Sec 3.1, and a cross modality fusion module in Sec 3.2. Figure 2 gives an overview of the framework.

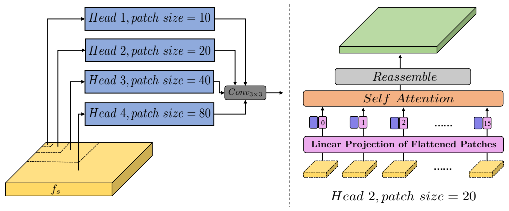
$Head\ 1, patch\ size = 10$
$Head\ 2, patch\ size = 20$
$Head\ 3, patch\ size = 40$
$Head\ 4, patch\ size = 80$
$Conv_{3\times3}$
f_s
Reassemble
Self Attention
0
1
2
......
15
Linear Projection of Flattened Patches
......
$Head\ 2, patch\ size = 20$

Figure 3: Illustration of the Multi-scale Transformer.

3.1 Multi-scale Transformer

We wish to locate regions that contain manipulation artifacts and thus are inconsistent with their other regions in the image. This requires modeling long-range relationships in images, *i.e.*, calculating the similarity of regions not only in a local neighborhood but also lie far apart. Inspired by the great success of transformer models in capturing long-term context information, we use transformers for Deepfake detection. Unlike recent approaches that directly split an input image into multiple patches of the same size as inputs to transformers [14], we introduce a multi-scale transformer, which generates patches of different scales. The intuition behind is to cover regions with different sizes so as to identify artifacts generated by manipulation methods.

More formally, denote the input image as $X \in \mathbb{R}^{H\times W\times 3}$, where H and W are the height and width of the image, respectively. f represents the backbone network, and $\boldsymbol{f_t}$ is the feature map extracted from the t-th layer. The feature map $\boldsymbol{f_s} \in \mathbb{R}^{(H/4)\times(W/4)\times C}$ is first extracted from the shallow layers of f. Then to capture the forgery patterns at multiple scales, we split the feature map into spatial patches of different sizes and calculate patch-wise self-attention in different heads. Specifically, we extract spatial patches $\{p^h\}_{h=1}^N$ of shape $r_h \times r_h \times C$ from f_s where $N = (H/4r_h) \times (W/4r_h)$, and reshape them into 1-dimension vectors for the h-th head. After that, we use fully-connected layers to embed the flattened vectors into query embeddings $\{q^h\}_{h=1}^N$. Similar operations are implemented to obtain key and value emebddings, respectively. Then we calculate the patch-wise similarities by matrix multiplication and $softmax$ function:

$$\alpha_{i,j}^h = softmax\left(\frac{\boldsymbol{q}_i^h \cdot (\boldsymbol{k}_j^h)^T}{\sqrt{r_h \times r_h \times C}}\right), 1 \leq i,j \leq N, \tag{1}$$

where $\boldsymbol{q}_i^h$ denotes the i-th query embedding and $\boldsymbol{k}_j^h$ denotes the j-th key embedding. Then we obtain the output for the query patch by weighted sum the values from relevant patches:

$$\boldsymbol{o}_i^h = \sum_{j=1}^{N} \alpha_{i,j}^h \boldsymbol{v}_j^h. \tag{2}$$

After receiving the output for all patches, we stitch them together and reshape to the original spatial resolution. Finally, the features from different heads are concatenated and further passed through a 2D residual block to obtain the output $\boldsymbol{f_{mt}} \in \mathbb{R}^{(H/4)\times(W/4)\times C}$. The detailed architecture of the multi-scale transformer is illustrated in Figure 3.

3.2 Cross Modality Fusion

It has been shown that artifacts in manipulated images and videos are no longer perceptible with compression approaches like JPEG compression [15, 24, 59, 65]. Following [5, 44], we also compute features from the frequency domain to complement RGB features. The resulting frequency features are combined with RGB features with a cross modality fusion module.

In particular, we first apply the Discrete Cosine Transform (DCT) to transform the input image X from the RGB domain to the frequency domain and obtain $\mathcal{DCT}(X) \in \mathbb{R}^{H\times W\times 1}$. Benefited from the properties of DCT, low-frequency responses are placed in the top-left corner of $\mathcal{DCT}(X)$, while high-frequency responses are located in the bottom-right corner. Following [44], we separate the frequency domain into low, middle, and high frequency bands with three hand-crafted binary base filters $\{\mathbf{t}^i_{base}\}^3_{i=1}$ and obtain the decomposed frequency components:

$$\boldsymbol{d}_i = \mathcal{DCT}(X) \odot \boldsymbol{t}^i_{base}, i = \{1, 2, 3\}, \tag{3}$$

where $\odot$ denotes the element-wise dot-product. Empirically, we manually design the base filters with the following pattern: the low frequency band $\boldsymbol{t}^1_{base}$ is the first 1/16 of the entire spectrum, the middle frequency band $\boldsymbol{t}^2_{base}$ is between 1/16 and 1/8 of the spectrum, and the high frequency band $\boldsymbol{t}^3_{base}$ is the last 7/8 of the spectrum.

To preserve the shift invariance and local consistency of natural images and explore the representative capability of CNNs, we then invert $\boldsymbol{d}_i$ back into the RGB domain via IDCT: $\boldsymbol{b}_i = \mathcal{DCT}^{-1}(\boldsymbol{d}_i), i = \{1, 2, 3\}$. Finally, we re-assemble $\{\boldsymbol{b}_i\}^3_{i=1}$ along the channel axis to obtain the frequency-aware spatial map $\boldsymbol{B} \in \mathbb{R}^{H\times W\times 3}$, and input it to several stacked convolution layers to extract frequency features f_{fq}, the size of which is the same as f_s.

Given RGB features f_s and frequency features f_{fq}, we use a Cross Modality Fusion (CMF) to combine them into a unified representation. Inspired by the architecture of self-attention in transformers, we design a fusion block using the query-key-value mechanism. Specifically,we first embed f_s and f_{fq} into $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ using 1× 1 convolutions $Conv_q$, $Conv_k$, and $Conv_v$, respectively:

$$\boldsymbol{Q} = Conv_q(f_s), \boldsymbol{K} = Conv_k(f_{fq}), \boldsymbol{V} = Conv_v(f_{fq}), \tag{4}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ retain the original spatial sizes. Then we flatten them along the spatial dimension to obtain the 2D embeddings $\widetilde{\boldsymbol{Q}}$, $\widetilde{\boldsymbol{K}}$, and $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{(HW/16)\times C}$, and calculate the fused features as:

$$f_{fuse} = softmax\left(\frac{\widetilde{\boldsymbol{Q}}\widetilde{\boldsymbol{K}}^T}{\sqrt{H/4\times W/4\times C}}\right)\widetilde{\boldsymbol{V}}. \tag{5}$$

Finally, we employ a residual connection by adding f_s, f_{mt}, and f_{fuse}, and use a convolutional layer to obtain the output f_{cmf}:

$$f_{cmf} = Conv_{3\times 3}(f_s + f_{mt} + f_{fuse}). \tag{6}$$

3.3 Loss functions

Cross-entropy loss. The feature maps f_{cmf} are then passed through several layers, followed by a single-scale Transformer (patch size equal to 2 × 2) to obtain global semantic features, which are finally used to predict whether the input image is real or fake using a cross-entropy loss $\mathcal{L}_{cls}$:

$$\mathcal{L}_{cls} = ylog\hat{y} + (1-y)log(1-\hat{y}), \tag{7}$$

where y is set to 1 if the face image has been manipulated, otherwise it is set to 0; $\hat{y}$ denotes the predicted label by our network.

Segmentation loss It is worth noting using a binary classifier tends to result in overfitted models. We additionally predict the the face region as an auxiliary task to enrich the supervision for training the networks. Specifically, we input the fused feature map f_{lt} to a decoder to produce a binary mask $\hat{\boldsymbol{M}}\ in\ \mathbb{R}^{H\times W}$:

$$\mathcal{L}_{seg} = \sum_{i,j} M_{i,j}log\hat{M}_{i,j} + (1-M_{i,j})log(1-\hat{M}_{i,j}), \tag{8}$$

where $M_{i,j}$ is the ground-truth mask, with 1 indicating the manipulated pixels and 0 otherwise.

Contrastive loss Deepfake images generated by different facial manipulation methods differ in forgery patterns, while the distribution of real images is relatively stable. To improve the generalization ability of our detection model, we first calculate the feature centers of N_p real samples $\boldsymbol{C}_{pos} = \frac{1}{N_p}\sum^{N_p}_{i=1} f^{pos}_i$ and additionally use a contrastive loss to make features from pristine samples to be closer towards the feature center than manipulated samples. Formally, the contrastive loss is defined as:

$$L_{con} = \frac{1}{N_p}\sum^{N_p}_{i=1} d(f^{pos}_i, C_{pos}) - \frac{1}{N_n}\sum^{N_n}_{i=1} d(f^{neg}_i, C_{pos}), \tag{9}$$

where N_n denotes the number of negative samples, and d computes distance with cosine similarity. Finally, combining Eqn. 7, Eqn. 8 and Eqn. 9, the training objective can be written as:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1\mathcal{L}_{seg} + \lambda_2\mathcal{L}_{con}, \tag{10}$$

where λ_1 and λ_2 are the balancing hyper-parameters. By default, we set $\lambda_1 = 1$ and $\lambda_2 = 0.001$.

4 SR-DF DATASET

To stimulate research for Deepfake forensics, we introduce a large-scale and challenging dataset SR-Df. SR-DF is built upon the pristine videos in the FF++ dataset, which contain a diverse set of samples in different genders, ages, and ethnic groups. We first crop face regions in each video frame using [27], and then generate forged videos using state-of-the-art Deepfake generation techniques. Finally, we use the image harmonization method in [6] for post-processing. Below we introduce these steps in detail.

4.1 Dataset Construction

Synthesis Approaches To guarantee the diversity of synthesized images, we use four facial manipulation methods, including two face swapping methods: **FSGAN** [42] and **FaceShifter** [31], and two facial reenactment methods: **First-order-motion** [50] and **IcFace** [56]. Note that the manipulation methods we leverage are all identity-agnostic—they can be applied to arbitrary face images without training in pairs, which is different from the FF++ [48] dataset. The detailed forgery images generation process will be described in Appendix.

Post-processing In order to resolve the color mismatch between the face regions and the background and to eliminate the stitched

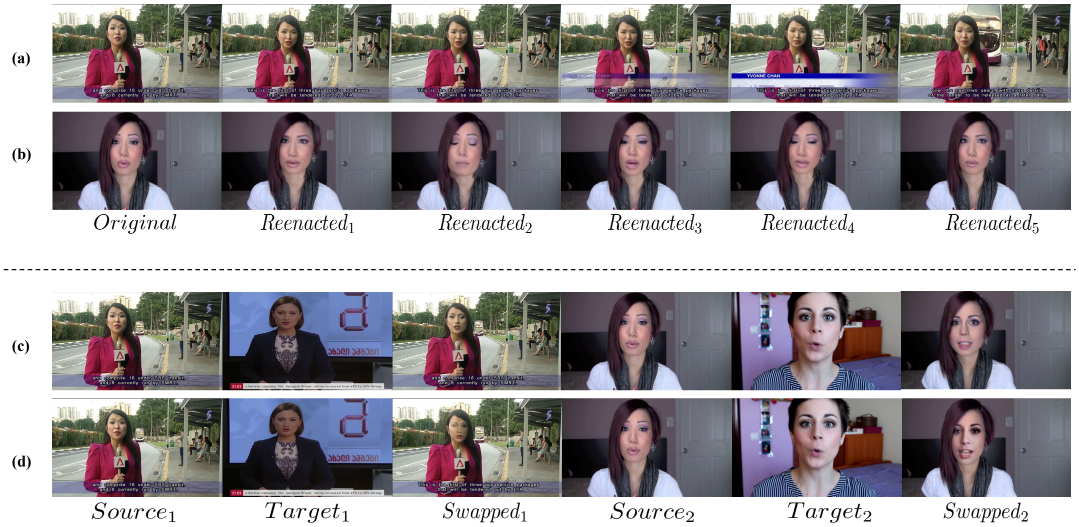

Figure 4: Example frames from the SR-DF dataset. The first two rows are generated by manipulating facial expressions: (a) First-order-motion and (b) IcFace, while the last two rows are generated by manipulating facial identity: (c) FaceShifter and (d) FSGAN.

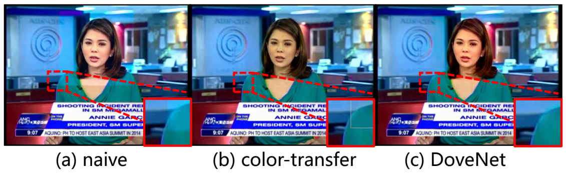

Figure 5: Synthesized images of blending the altered face into the background image. We compare three blending methods: naive stitching (left), stitching with color transfer (middle), and stitching with DoveNet (right).

Table 1: A comparison of SR-DF dataset with existing datasets for Deepfake detection. LQ: low-quality, HQ: high-quality.

Dataset	**Real**		**Forged**	
	Video	Frame	Video	Frame
UADFV [63]	49	17.3k	49	17.3k
DF-TIMIT-LQ [28]	320	34.0k	320	34.0k
DF-TIMIT-HQ [28]	320	34.0k	320	34.0k
FF++ [48]	1,000	509.9k	4000	1,830.1k
DFD [9]	363	315.4k	3,068	2,242.7k
DFDC [13]	1,131	488.4k	4,113	1,783.3k
WildDeepfake [69]	3,805	440.5k	3,509	739.6k
Celeb-DF [35]	590	225.4k	5,639	2,116.8k
SR-DF (ours)	1,000	509.9k	4,000	2,078.4k

boundaries, we use DoveNet [6] for post-processing, which is a state-of-the-art image harmonization method to make the foreground compatible with the background. Note that the masks that we use to distinguish foreground and background are generated using a face parsing model [16]. We compare the blending results with naive stitching and a color transfer algorithm [47] adopted by [35], and show an example of synthesized image in Figure 5.

Existing Deepfake Datasets The **UADFV** dataset [63] contains 49 real videos collected from YouTube and 49 Deepfake videos that are generated using FakeAPP [20]. The **DeepFake-TIMIT** dataset [28] includes 320 real videos and 640 Deepfake videos (320 high-quality and 320 low-quality) generated with faceswap-GAN [19]. The FaceForensics++ (**FF++**) dataset [48] has 1,000 real videos from YouTube and 4,000 corresponding Deefake videos that are generated with 4 face manipulation methods: Deepfakes [10], FaceSwap [18], Face2Face [55], and NeuralTextures [54]. The Google/Jigsaw DeepFake detection (**DFD**) [9] dataset includes 3,068 Deepfake videos that are generated based on 363 original videos. The Facebook DeepFake detection challenge (**DFDC**) dataset [13] is part of the DeepFake detection challenge, which has 1,131 original videos and 4,113 Deepfake videos. The **WildDeepfake** [69] dataset consists of 3,805 real videos and 3,509 fake videos that are collected from the Internet. The **Celeb-DF** dataset [35] contains 590 real videos and 5,639 Deepfake videos created using the same synthesis algorithm. We summarize the basic information of these existing datasets and our SR-DF dataset in Table 1.

4.2 Visual Quality Assessment

As mentioned above, how to measure the quality of forged images in these datasets is under-explored. Therefore, we introduce a variety of quantitative metrics to benchmark the quality of current datasets from four perspectives: identity retention, authenticity, temporal smoothness, and diversity. To the best of our knowledge, this is the most comprehensive evaluation system to measure the quality of Deepfake datasets.

Mask-SSIM First, we follow [35] to adopt the Mask-SSIM score as a measurement of synthesized Deepfake images. Mask-SSIM refers to the SSIM score between the face regions of the forged image and the corresponding original image. We use the [16] to generate facial masks and compute the Mask-SSIM on our face swapping subsets. Table 2 demonstrates the average Mask-SSIM scores of all compared datasets, and SR-DF dataset achieves the highest scores.

Perceptual Loss *Perceptual loss* is usually used in face inpainting approaches [39, 64] to measure the similarity between the restored

Table 2: Average Mask-SSIM scores and perceptual loss of different Deepfake datasets. The value of Mask-SSIM is in the range of [0,1], with the higher value corresponding to better image quality. We follow [35] to calculate Mask-SSIM on videos that we have exact corresponding correspondences for DFD and DFDC dataset. For perceptual loss, lower value indicates the better image quality.

Dataset	FF++	DFD	DFDC	Celeb-DF	Ours
Mask-SSIM ↑	0.82	0.86	0.85	0.91	**0.92**
Perceptual Loss ↓	0.67	0.69	0.63	**0.59**	0.60

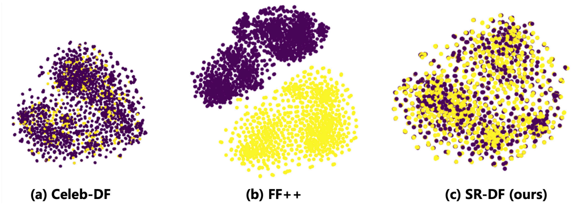

Figure 6: A feature perspective comparison of Celeb-DF, FF++ dataset (RAW) and SR-DF dataset. We use an ImageNet-pretrained ResNet-18 network to extract features and t-SNE [57] for dimension reduction. Note that we only select one frame in each video for visualization.

Table 3: Average E_{warp} values of different datasets, with lower value corresponding to smoother temporal results. We also calculate the E_{warp} of pristine videos in our dataset.

Dataset	DFD	FF++	Celeb-DF	Ours	Real
E_{warp}	69.53	73.16	**49.10**	56.95	14.28

faces and corresponding complete faces. Inspired by this, we use the *relu*1_1, *relu*2_1, *relu*3_1, *relu*4_1 and *relu*5_1 of the pretrained VGG-19 network on ImageNet [11] to calculate the perceptual loss between the feature maps of forged faces and that of corresponding real faces. Note that we use [27] to crop the facial regions. We compare the perceptual loss of different datasets in Table 2. Although the perceptual loss of SR-DF dataset is slightly higher than that of Celeb-DF, it is lower than other datasets by a large margin.

Ewarp The warping error E_{warp} is used by [4, 22, 30] to measure the temporal inconsistency for video style transfer. We use it to compute the E_{warp} of consecutive forged frames in different datasets to quantitatively measure the short-term consistency. Following [30], we use the method in [49] to calculate occlusion map and PWC-Net [51] to obtain optical flow. E_{warp} of different Deepfake datasets are shown in Table 3 for comparison.

Feature Space Distribution As can be seen from the above, Celeb-DF dataset [35] has a decent performance in visual quality. However, they only used one face swapping method to generate all the forged images, which results in limited diversity of data distribution. We illustrate this by visualizing the feature space of Celeb-DF [35], FF++ dataset [48], and SR-DF in Figure 6. We can see the data distribution of the Celeb-DF dataset is more concentrated, while the real and forged images of FF++ dataset can be easily separated in the feature space. On the other hand, the data in SR-DF dataset are more scattered in the 2D space.

5 EXPERIMENTS

5.1 Experimental Settings

Datasets We conduct experiments on FaceForensics++ (FF++) [48], Celeb-DF [35], and the newly proposed SR-DF dataset. FF++ consists of 1,000 original videos with real faces, in which 720 videos are used for training, 140 videos are reserved for validation and 140 videos for testing. Each video is manipulated by four Deepfake methods, *i.e.*, Deepfakes [10], FaceSwap [18], Face2Face [55], and NeuralTextures [54]. Different degrees of compression are implemented on both real and forged images to produce high-quality (HQ) version and low-quality (LQ) version of FF++, respectively. Celeb-DF is comprised of 890 real videos and 5,639 Deepfake videos, in which 6,011 videos are used for training and 518 videos are for testing. For SR-DF, we build on the 1,000 original videos in FF++, and generate 4,000 forged videos using four state-of-the-art subject-agnostic Deepfake generation techniques (see details above). We use the same training, validation and test set partitioning of FF++.

When training on FF++ dataset and SR-DF dataset, following [44, 66], we augment the real images four times by repeated sampling to balance the number of real and fake samples. For FF++, we sample 270 frames from each video, following the setting in [44, 48].

Evaluation Metrics We apply the Accuracy score (Acc) and Area Under the RoC Curve (AUC) as our evaluation metrics, which are commonly used in various classification tasks including Deepfake detection [32, 41, 44, 48, 66].

Implementation Details For all real and forgery images, we use dlib [27] to crop the face regions as inputs with a size of 320 × 320. The patch sizes in Sec.4.2.1 are set to (80 × 80), (40 × 40), (20 × 20), and (10 × 10). For our backbone network, we use Efficient-b4 [53] pretrained on ImageNet [11]. We use Adam for optimization with a learning rate of 0.0001. The learning rate is decayed 10 times every 40 steps. We set the batch size to 24, and train the complete network for 90 epoches.

5.2 Evaluation on FaceForensics++

FF++ [48] is a widely used dataset in various Deepfake detection approaches [1, 7, 21, 32, 44, 66]. Therefore, we compare M2TR with current state-of-the-art Deepfake detection methods. We test the frame-level detection performance on RAW, HQ, and LQ, respectively, and report the AUC scores (%) in Table 4. We compare with top-notch methods, including: i.e., (i) Steg. Features [21], which assembles diverse noise component models to build a joint steganography detectors, (ii) LD-CNN [7], which uses CNN models as residual-based local descriptors for forgery detection, (iii) MesoNet [1], which mines the mesoscopic properties of forged images with the shallow layers of convolutional networks, (iv) Face X-ray [32], which detects the discrepancies across blending boundaries, (v) F^3-Net [44], which adopts a two-branch architecture where one makes use of frequency clues to recognize forgery patterns and the other extracts the discrepancy of frequency statistics between real and fake images, and (vi) MaDD [66], which proposes a multi-attentional

Table 4: Quantitative frame-level detection results on FaceForensics++ dataset under all quality settings. The best results are marked as bold.

Methods	LQ		HQ		RAW	
	ACC	AUC	ACC	AUC	ACC	AUC
Steg.Features [21]	55.98	-	70.97	-	97.63	-
LD-CNN [7]	58.69	-	78.45	-	98.57	-
MesoNet [1]	70.47	-	83.10	-	95.23	-
Face X-ray [32]	-	61.6	-	87.4	-	-
F^3-Net [44]	90.43	93.30	97.52	98.10	**99.95**	99.80
MaDD [66]	88.69	90.40	97.60	99.29	-	-
Ours	**92.35**	**94.22**	**98.23**	**99.48**	99.21	**99.91**

Deepfake detection framework to capture artifacts with multiple attention maps.

Table 4 summarizes the results and comparisons. We can see that our method achieves state-of-the-art performance on all versions (*i.e.*, LQ, HQ, and RAW) of FF++. This suggests the effectiveness of our approach in detecting Deepfakes of different visual qualities. Comparing across different versions of the FF++ dataset, we see that while most approaches achieve high performance on the high-quality version of FF++, we observe a significant performance degradation on FF++ (LQ) where the forged images are compressed. This could be remedied by leveraging frequency information. While both F3-Net and M2TR uses frequency features, M2TR achieves an accuracy of 92.35% in the LQ setting, outperforming the F3-Net approach by 1.92%.

5.3 Evaluation on Celeb-DF and SR-DF

In this section, we conduct experiments to evaluate the detection accuracy of our M2TR on Celeb-DF [35] dataset and SR-DF dataset at frame-level, respectively. Note that we do not report the quantitative results of certain state-of-the-art Deepfake detection methods including [32, 44, 66] because the code and models are not publicly available. The results are reported in Table 5. We observe that our M2TR achieves 99.9% and 90.5% on CeleDF and SR-DF, respectively, which demonstrate that our method outperforms all the other Deepfake detection methods over different datasets. This suggests that our approach is indeed effective for Deepfake detection across different datasets.

In addition, the quality of different Deepfake datasets can be evaluated by comparing the detection accuracy of the same detection method on different datasets. Given that Celeb-DF [35] contains high-quality samples (as discussed in 4.2, Celeb-DF achieves the best results on *Mask-SSIM*, *Perceptual loss* and *Ewarp* metrics in the available Deepfake dataset.), we calculate the average frame-level AUC scores of all compared detection methods on Celeb-DF dataset and SR-DF, and report them in the last row of Table 5. The overall performance on SR-DF is 9.2% lower than that of Celeb-DF, which demonstrates that SR-DF is more challenging.

Table 5: Frame-level AUC scores (%) of various Deepfake detection methods on Celeb-DF and SR-DF dataset.

Methods	Celeb-DF	SR-DF
Xception [48]	97.6	88.2
Multi-task [40]	90.5	85.7
Capsule [41]	93.2	81.5
DSW-FPA [33]	94.8	86.6
DCViT [60]	97.2	87.9
Ours	99.9	90.5
Avg	95.5	86.7

5.4 Generalization Ability

The generalization ability is at the core of Deepfake detection. We evaluate the generalization of our M2TR by separately training on FF++ (HQ) and SR-DF dataset, and test on other datasets. We follow [66] to sample 30 frames for each video and calculate the frame-level AUC scores. The comparison results are shown in Table 6. Note that for the Deepfake detection models that are not publicly available, we only use the results reported in their paper. The results in Table 6 demonstrate that our method achieves better generalization than most existing methods.

5.5 From Frames to Videos

Existing methods on Deepfake detection mainly perform evaluation based on frames extracted from videos albeit videos are provided. However, in real-world scenarios, most Deepfake data circulating on the Internet are fake videos, therefore, we also conduct experiments to evaluate our M2TR on video-level Deepfake detection. The most significant difference between videos and images is the additional temporal information between frame sequences. We demonstrate that M2TR can be easily extended for video modeling by adding a temporal transformer to combine frame-level features generated by M2TR. We refer to such an extension as spatial-temporal M2TR (ST-M2TR).

In particular, we sampled 16 frames at intervals from one video, and directly use the model trained at the frame-level to extract features of different frames. These features are then input to a transformer block (it has 4 stacked encoders, each with 8 attention heads, and an MLP head that has two fc layers) to obtain video-level predictions. We report the AUC scores (%) and compare with (1) P3D [45], which simplifies 3D convolutions with 2D filters on spatial dimension and 1D temporal connections; (2)R3D [62], which encodes the video sequences using a 3D fully convolutional networks and then generates candidate temporal fragments for classification; (3)I3D [3], which expands 2D CNNs with an additional temporal dimension to introduce a two-stream inflated 3D convolutional network; (4) M2TR$_{mean}$, which averages the features of different frames by M2TR for video-level prediction. Note that (1) and (3) are designed for video action recognition, while (2) is for temporal activity detection, and we modify them for video-level Deepfake detection. The results are summarized in Table 7. We can see that our method achieves the best performance on both FF++ and SR-DF.

5.6 Ablation study

Effectiveness of Different Components The Multi-scale Transformer (MT) of our method is designed to capture local inconsistency between patches of different sizes, while the Cross Modality Fusion (CMF) module is utilized to introduce the frequency modality features and fuse it with RGB modality features effectively. To evaluate the effectiveness of MT and CMF, we remove them separately from M2TR and demonstrate the performance on FF++. The quantitative results are listed in Table 8, which validates that the

Table 6: AUC scores (%) for cross-dataset evaluation on FF++, Celeb-DF, and SR-DF datasets. Note that some methods have not made their code public, so we directly use the data reported in their paper. "–" denotes the results are unavailable.

Training Set	**Testing Set**	Xception [48]	Multi-task [40]	Capsule [41]	DSW-FPA [33]	Two-Branch [36]	F3-Net [44]	MaDD [66]	DCViT [60]	Ours
FF++	FF++	99.7	76.3	96.6	93.0	98.7	98.1	99.3	98.3	99.5
	Celeb-DF	48.2	54.3	57.5	64.6	**73.4**	65.2	67.4	60.8	65.7
	SR-DF	37.9	38.7	41.3	44.0	-	-	-	57.8	62.6
SR-DF	SR-DF	88.2	85.7	81.5	86.6	-	-	-	87.9	90.5
	FF++	63.2	58.9	60.6	69.1	-	-	-	62.6	77.9
	Celeb-DF	59.4	51.7	52.1	62.9	-	-	-	63.7	80.7

Table 7: Quantitative video-level detection results on different versions of FF++ dataset and SR-DF dataset. M2TR $_{mean}$ denotes averaging the extracted features obtained by M2TR for all frames as the video-level representation, while M2TR $_{vtf}$ denotes using VTF Block for temporal fusion. The best results are marked as bold.

Method	FF++ (RAW)	FF++ (HQ)	FF++ (LQ)	SR-DF
P3D [45]	80.9	75.23	67.05	65.97
R3D [62]	96.15	95.00	87.72	73.24
I3D [3]	98.23	96.70	93.18	80.11
M2TR $_{mean}$	98.19	98.77	93.28	82.09
ST-M2TR	**99.96**	**99.30**	**94.16**	**84.62**

Table 8: Ablation results on FF++ (HQ) and FF++ (LQ) with and without Multi-scale Transformer and CMF.

Method	LQ		HQ	
	ACC (%)	AUC (%)	ACC (%)	AUC (%)
Baseline (Efficient-b4)	87.34	88.76	93.79	96.01
w/o MT	90.44	91.89	95.36	97.17
w/o CMF	90.97	92.58	96.25	98.54
Ours	**92.35**	**94.22**	**98.23**	**99.48**

Table 9: Ablation results on FF++ (HQ) using multi-scale Transformer (MT) or single-scale transformer.

Patch size	ACC (%)	AUC (%)
40 × 40	95.62	97.33
20 × 20	96.81	98.29
10 × 10	97.55	98.94
Ours	**98.23**	**99.48**

Table 10: AUC (%) for cross-dataset evaluation on FF++ (HQ), Celeb-DF, and SR-DF with (denoted as M2TR) and without (denoted as M2TR $_{ncl}$) the supervision of constrative loss.

Training Set	**Testing Set**	M2TR $_{ncl}$	M2TR
FF++	Celeb-DF	63.9	65.7
	SR-DF	59.1	62.6
SR-DF	FF++	74.2	77.9
	Celeb-DF	78.5	80.7

use of of MT and CMF can effectively improve the detection performance of our model. In particular, the proposed CMF module brings a remarkable improvement to our method under the low-quality (LQ) setting, *i.e.*, about 1.7% performance gain on AUC score, which is mainly benefited from the complementary information from the frequency modality.

Effectiveness of the Multi-scale Design To verify the effectiveness of using multi-scale patches in different heads in our multi-scale transformer, we replace MT with several single-scale transformers with different patch sizes, and conduct experiments on FF++ (HQ). The results in Table 9 demonstrate that our full model achieves the best performance with MT, *i.e.*, 2.2%, 1.2%, and 0.5% higher than 40 × 40, 20 × 20 and 10 × 10 single-scale transformer on AUC score. This confirms the the use of a multi-scale transformer is indeed effective.

Effectiveness of the Contrastive Loss To illustrate the contribution of the constrastive loss in improving the the generalization ability of our method, we conduct experiments to train M2TR without its supervision and evaluate the cross-dataset detection accuracy. The comparison results are reported in Table 10. We can see that 1) When training on FF++ without the constrastive loss, the accuracy decreases by 2.7% and 5.6% in Celeb-DF and SR-DF, respectively. 2) When training on SR-DF dataset without the constrastive loss, the accuracy decreases by 4.7% and 2.7%, respectively.

6 CONCLUSION

In this paper, we presented a Multi-modal Multi-scale Transformer (M2TR) for Deepfake detection, which uses a multi-scale transformer to capture subtle local inconsistency at multiple scales. Additionally, we also introduced a cross modality fusion module to improve the robustness against image compression. Besides, we introduced a challenging dataset **SR-DF** that are generated with several state-of-the-art face swapping and facial reenactment methods. We also built the most comprehensive evaluation system to quantitatively verify that the SR-DF dataset is better than existing datasets in terms of visual quality and data diversity. Extensive experiments on different datasets demonstrate the effectiveness and generalization ability of the proposed method.

REFERENCES

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. 2018. Mesonet: a compact facial video forgery detection network. In *WIFS*.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *ECCV*.

[3] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

[4] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017. Coherent online video style transfer. In *ICCV*.

[5] Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. 2021. Local Relation Learning for Face Forgery Detection. In *AAAI*.

[6] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. 2020. DoveNet: Deep Image Harmonization via Domain Verification. In *CVPR*.

[7] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. 2017. Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In *Workshop on IH&MMSec*.

[8] Kevin Dale, Kalyan Sunkavalli, Micah K Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. 2011. Video face replacement. In *SIGGRAPH Asia*.

[9] DeepFake Detection Dataset. 2019. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html.

[10] Deepfakes. 2018. github. https://github.com/deepfakes/faceswap.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[13] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. 2020. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397* (2020).

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[15] Ricard Durall, Margret Keuper, Franz-Josef Pfreundt, and Janis Keuper. 2019. Unmasking deepfakes with simple features. *arXiv preprint arXiv:1911.00686* (2019).

[16] Face-parsing. 2019. github. https://github.com/zllrunning/face-parsing.PyTorch.

[17] FaceShifter. 2020. github. https://github.com/mindslab-ai/faceshifter.

[18] Faceswap. 2018. github. https://github.com/MarekKowalski/FaceSwap/.

[19] Faceswap-GAN. 2019. github. https://github.com/shaoanlu/faceswap-GAN.

[20] Fakeapp. 2018. https://www.fakeapp.com/.

[21] Jessica Fridrich and Jan Kodovsky. 2012. Rich models for steganalysis of digital images. *TIFS* (2012).

[22] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. 2017. Real-time neural style transfer for videos. In *CVPR*.

[23] Po-Hsiang Huang, Fu-En Yang, and Yu-Chiang Frank Wang. 2020. Learning Identity-Invariant Motion Representations for Cross-ID Face Reenactment. In *CVPR*.

[24] Ying Huang, Wenwei Zhang, and Jinzhuo Wang. 2020. Deep frequent spatial temporal learning for face anti-spoofing. *arXiv preprint arXiv:2002.03723* (2020).

[25] Hyeonseong Jeon, Youngoh Bang, and Simon S Woo. 2020. FDFtNet: Facing off fake images using fake detection fine-tuning network. In *ICT Systems Security and Privacy Protection.*

[26] Ira Kemelmacher-Shlizerman. 2016. Transfiguring portraits. *ACM TOG* (2016).

[27] Davis E King. 2009. Dlib-ml: A machine learning toolkit. *JMLR* (2009).

[28] Pavel Korshunov and Sébastien Marcel. 2018. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685* (2018).

[29] Mohammad Rami Koujan, Michail Christos Doukas, Anastasios Roussos, and Stefanos Zafeiriou. 2020. Head2head: Video-based neural head synthesis. *arXiv preprint arXiv:2005.10954* (2020).

[30] Chenyang Lei, Yazhou Xing, and Qifeng Chen. 2020. Blind video temporal consistency via deep video prior. *arXiv preprint arXiv:2010.11838* (2020).

[31] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. 2019. Faceshifter: Towards high fidelity and occlusion aware face swapping. *arXiv preprint arXiv:1912.13457* (2019).

[32] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. 2020. Face x-ray for more general face forgery detection. In *CVPR*.

[33] Yuezun Li and Siwei Lyu. 2018. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656* (2018).

[34] Yuezun Li and Siwei Lyu. 2019. Exposing DeepFake Videos By Detecting Face Warping Artifacts. In *CVPRW*.

[35] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. 2020. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *CVPR*.

[36] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. 2020. Two-branch recurrent network for isolating deepfakes in videos. In *ECCV*.

[37] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612* (2017).

[38] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. 2018. Rsgan: face swapping and editing using face and hair representation in latent spaces. *arXiv preprint arXiv:1804.03447* (2018).

[39] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. 2019. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212* (2019).

[40] Huy H Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. 2019. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *BTAS*.

[41] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. 2019. Use of a capsule network to detect fake images and videos. *arXiv preprint arXiv:1910.12467* (2019).

[42] Yuval Nirkin, Yosi Keller, and Tal Hassner. 2019. Fsgan: Subject agnostic face swapping and reenactment. In *ICCV*.

[43] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. 2018. Ganimation: Anatomically-aware facial animation from a single image. In *ECCV*.

[44] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. 2020. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*.

[45] Zhaofan Qiu, Ting Yao, and Tao Mei. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*.

[46] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[47] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. 2001. Color transfer between images. *IEEE CG&A* (2001).

[48] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. 2019. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*.

[49] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. 2016. Artistic style transfer for videos. In *GCPR*.

[50] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2020. First order motion model for image animation. *arXiv preprint arXiv:2003.00196* (2020).

[51] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*.

[52] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. 2017. Synthesizing obama: learning lip sync from audio. *ACM TOG* (2017).

[53] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*.

[54] Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred neural rendering: Image synthesis using neural textures. *ACM TOG* (2019).

[55] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Niessner. 2016. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *CVPR*.

[56] Soumya Tripathy, Juho Kannala, and Esa Rahtu. 2020. Icface: Interpretable and controllable face reenactment using gans. In *WACV*.

[57] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR* (2008).

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[59] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. 2020. CNN-generated images are surprisingly easy to spot... for now. In *CVPR*.

[60] Deressa Wodajo and Solomon Atnafu. 2021. Deepfake Video Detection Using Convolutional Vision Transformer. *arXiv preprint arXiv:2102.11126* (2021).

[61] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. 2018. Reenactgan: Learning to reenact faces via boundary transfer. In *ECCV*.

[62] Huijuan Xu, Abir Das, and Kate Saenko. 2017. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*.

[63] Xin Yang, Yuezun Li, and Siwei Lyu. 2019. Exposing deep fakes using inconsistent head poses. In *ICASSP*.

[64] Yang Yang, Xiaojie Guo, Jiayi Ma, Lin Ma, and Haibin Ling. 2019. Lafin: Generative landmark guided face inpainting. *arXiv preprint arXiv:1911.11394* (2019).

[65] Ning Yu, Larry S Davis, and Mario Fritz. 2019. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *ICCV*.

[66] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. 2021. Multi-attentional Deepfake Detection. In *CVPR*.

[67] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. 2017. Two-stream neural networks for tampered face detection. In *CVPRW*.

[68] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2020. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv preprint arXiv:2010.04159* (2020).

[69] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. 2020. WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection. In *ACM MM*.

A SYNTHESIS METHODS

A.1 FSGAN

FSGAN [42] follows the following pipeline to swap the faces of the source image I_s to that of the target image I_t. First, the swap generator G_r estimates the swapped face I_r and its segmentation mask S_r based on I_t and a heatmap encoding the facial landmark of I_s, while G_s estimates the segmentation mask S_s of the source image I_s. Then the inpainting generator G_c inpaints the missing parts of I_r based on S_s to estimate the complete swapped face Ic. Finally, using the segmentation mask S_s , the blending generator G_b blends I_c and I_s to generate the final output I_b which preserves the posture of I_s but owns the identity of I_t. For our dataset, we directly use the pretrained model provided by [42] and inference on our pristine videos.

A.2 FaceShifter

There are two networks in *FaceShifter* [31] for full pipeline: AEI-Net for face swapping, and HEAR-Net for occlusion handling. As the author of [31] have not public their code, we use the code from [17] who only implements AEI-Net and we train the model on our data. Specifically, AEI-Net is composed of three components: 1) an Identity Encoder which adopts a pretrained state-of-the-art face recognition model to provide representative identity embeddings. 2) a Multi-level Attributes Encoder which encodes the features of facial attributes . 3) an AAD-Generator which integrates the information of identity and attributes in multiple feature levels and generates the swapped faces. We use the parameters declared in [31] to train the model.

A.3 First-order-motion

First-order-motion [50] decouples appearance and motion information for subject-agnostic facial reenactment. Their framework comprises of two main modules: the motion estimation module which uses a set of learned keypoints along with their local affine transformations to predict a dense motion field, and an image generation module which combines the appearance extracted from the source image and the motion derived from the driving video to model the occlusions arising during target motions. To process our dataset, we use the pretrained model on VoxCeleb dataset [37], which contains speech videos from speakers spanning a wide range of different ethnic groups, accents, professions and ages, and reenact the faces in our real videos.

A.4 IcFace

IcFace [56] is a generic face animator that is able to transfer the expressions from a driving image to a source image. Specifically, the generator G_N takes the source image and neutral facial attributes as input and produces the source identity with central pose and neutral expression. Then the generator G_A takes the neutral image and attributes extracted from the driving image as an input and produces an image with the source identity and driving image's attributes. We train the complete model on our real videos in a self-supervised manner, using the parameters that they use to train on VoxCeleb dataset [37].

B VISUALIZATION OF MASK DETECTION RESULTS

Our method also predicts the manipulated regions of the face image for Deepfake localization. We demonstrate some examples of the mask detection results in Figure 7, from which we can see that the manipulated areas could be accurately located.

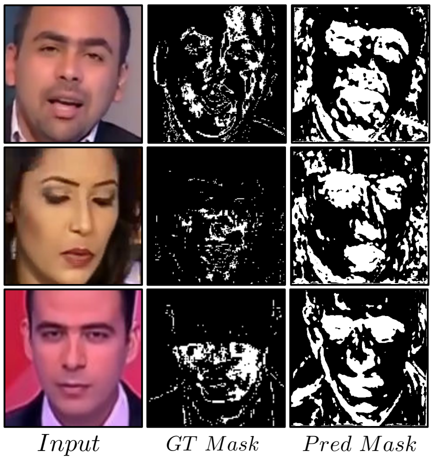

Figure 7: Visual examples of the input image, the ground-truth mask, and the predicted mask.