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

Junke Wang  
Shanghai Key Lab of Intel.  
Info. Proc., School of CS,  
Fudan University  
wangjk21@m.fudan.edu.cn

Zuxuan Wu  
Shanghai Key Lab of Intel.  
Info. Proc., School of CS,  
Fudan University  
zxwu@fudan.edu.cn

Wenhao Ouyang  
Shanghai Key Lab of Intel.  
Info. Proc., School of CS,  
Fudan University  
ouywh21@m.fudan.edu.cn

Jingjing Chen  
Shanghai Key Lab of Intel.  
Info. Proc., School of CS,  
Fudan University  
chenjingjing@fudan.edu.cn

Xiyi Zhong  
Shanghai Key Lab of Intel.  
Info. Proc., School of CS,  
Fudan University  
zhongxy21@m.fudan.edu.cn

Ser-Nam Lim  
Meta AI  
sernamlim@fb.com

Yu-Gang Jiang†  
Shanghai Key Lab of Intel.  
Info. Proc., School of CS,  
Fudan University  
ygj@fudan.edu.cn

ABSTRACT

The widespread dissemination of Deepfakes demands effective approaches that can detect perceptually convincing forged images. In this paper, we aim to capture the subtle manipulation artifacts at different scales using transformer models. In particular, we introduce a Multi-modal Multi-scale Transformer (M2TR), which operates on patches of different sizes to detect local inconsistencies in images at different spatial levels. M2TR further learns to detect forgery artifacts in the frequency domain to complement RGB information through a carefully designed cross modality fusion block. In addition, to stimulate Deepfake detection research, we 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. We conduct extensive experiments to verify the effectiveness of the proposed method, which outperforms state-of-the-art Deepfake detection methods by clear margins.

CCS CONCEPTS

• Computing methodologies → Computer vision.

KEYWORDS

Deepfake detection, Multiscale transformer, Deepfake dataset

ACM Reference Format:

Junke Wang, Zuxuan Wu, Wenhao Ouyang, Xintong Han, Jingjing Chen, Ser-Nam Lim, and Yu-Gang Jiang†. 2022. M2TR: Multi-modal Multi-scale Transformers for Deepfake Detection. In Proceedings of the 2022 International Conference on Multimedia Retrieval (ICMR ’22), June 27–30, 2022, Newark, NJ, USA. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3512527.3531415

1 INTRODUCTION

Recent years have witnessed the rapid development of Deepfake techniques [28, 31, 54], which enable attackers to manipulate the facial regions of an image and generate a forged image. As the synthesized images are becoming more photo-realistic, it is extremely difficult to distinguish whether an image has been manipulated even for the 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, 6, 27, 34, 36, 42, 72]. 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 [7, 22, 52]. These methods capture artifacts from the face regions in a single scale with stacked convolutional operations [55, 70]. While decent detection results are achieved by stacked convolutions, they excel at modeling local information but fail to consider the relationships of pixels globally due to constrained receptive field.

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 size. For instance, Figure 1 gives examples from the DFDC dataset [15]. 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. 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 [14, 48, 61] and computer vision tasks [2, 16, 62, 63, 73]. Unlike traditional vision transformers that usually operate on a single-scale, we propose a multi-scale architecture to capture forged regions that potentially have different sizes. Furthermore, [17, 26, 46, 64] 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.

To this end, we introduce M2TR, a Multi-modal Multi-scale Transformer, for Deepfake detection. As illustrated in Figure 2, M2TR follows a two-stream architecture, where the RGB stream captures the inconsistency among different regions within an image at multiple scales in RGB domain, and the frequency stream adopts learnable frequency filters to filter out forgery features in frequency domain. We also design a cross modality fusion block to combine the information from both streams more effectively in an interactive fashion. Finally, the integrated features are input to fully connected 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. However, the quality of visual samples in current Deepfake datasets [15, 30, 37, 49, 68] 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 [37] 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, Swapping and Reenactment DeepFake (SR-DF) dataset, which is generated using the state-of-the-art face swapping and facial reenactment methods [19, 44, 51, 59] for the development and evaluation of Deepfake detection methods. 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 inconsistencies at different scales and leverages frequency features to improve the robustness. 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 training Deepfake detection methods due to its quality and diversity.

2 RELATED WORK

Deepfake Detection To mitigate the security threat brought by Deepfakes, a variety of methods have been proposed for Deepfake detection. [72] uses a two-stream architecture to capture facial manipulation clues and patch inconsistency separately, while [42] simultaneously identifies forged faces and locates the manipulated regions with multi-task learning.

Recently, Face X-ray [34] 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 facial manipulation methods. DCViT [65] extracts features from the face image using a CNN model, which are then fed to a traditional single-scale transformer for forgery detection. MaDD [71] proposes a multi-attentional Deepfake detection framework to capture artifacts with multiple attention maps. However, most of them only focus on the features in the RGB domain, thus failing to detect forged images which are manipulated subtly in the color-space. Instead, F³-Net [46] 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. In this paper, we use a multi-scale transformer to capture local inconsistencies 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 [61] have demonstrated impressive performance for natural language processing tasks due to strong abilities in modeling long-range context information. Recently, researchers have demonstrated remarkable interests in using the transformer for a variety of computer vision tasks. Typically, visual transformers [2, 16, 58] model the interactions between tokens of the same scale with the self-attention mechanisms. ViT reshapes an image into a sequence of flattened patches and inputs them to the transformer encoder for image classification [16]. 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]. Unlike these approaches, we propose to split the inputs into patches of different sizes, and integrate multi-scale information for better visual representation with vision transformers.
3 APPROACH

We aim to detect the subtle forgery artifacts that are hidden in the forged images and improve the robustness to image compression with frequency features. In this section, we introduce the Multimodal Multi-scale Transformer (M2TR) for Deepfake detection, which consists of stacked multi-scale transformers (Sec 3.1), frequency filters (Sec 3.2), and cross modality fusion blocks (Sec 3.3).

Figure 2 gives an overview of the framework.

More formally, we denote an 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. We first use several convolutional layers to extract features $F \in \mathbb{R}^{(H/4) \times (W/4) \times C}$ of $X$, which are then input to successive multi-scale transformer and frequency filters for forgery clues detection.

The intuition of using convolutions here is to ensure faster convergence and more stable training [66].

3.1 Multi-scale Transformer

To capture forgery patterns at multiple scales, we introduce a multi-scale transformer which operates on patches of different sizes. Taking the output of the previous cross modality fusion block $M_{i-1}$ (which is initialized as $F$) as input, we split it into spatial patches of different sizes and calculate patch-wise self-attention in different heads. Specifically, we first extract patches of shape $r_h \times r_h \times C$ from $M_{i-1}$, and reshape them into 1-dimensional vectors for the $h$-th head. After that, we use fully-connected layers to embed the flattened vectors into query embeddings $Q^h_i \in \mathbb{R}^{N \times C_h}$, where $N = (H/4r_h) \times (W/4r_h)$, and $C_h = r_h \times r_h \times C$. Similar operations are implemented to obtain key embeddings $K^h_i$ and value embeddings $V^h_i$, respectively. Then we calculate the attention matrix through the following process:

$$A^h_i = \text{softmax} \left( \frac{Q^h_i (K^h_i)^T}{C_h} \right) V^h_i, \quad (1)$$

$A^h_i$ is then reshaped 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 $T_i \in \mathbb{R}^{(H/4) \times (W/4) \times C}$.

3.2 Frequency Filter

It has been shown that artifacts in manipulated images and videos are no longer perceptible with compression approaches like JPEG compression [17, 26, 64]. Therefore, we extract the forgery features in the frequency domain to complement RGB features.

Specifically, we first apply 2D FFT along the spatial dimensions to transform $M_{i-1}$ to the frequency domain, and obtain the spectrum representation $F(M_{i-1}) \in \mathbb{R}^{H/4 \times W/4 \times C}$. We then multiply $F(M_{i-1})$ with a learnable filter $G_i \in \mathbb{R}^{H/4 \times W/4 \times C}$ to model the dependencies of different frequency band components:

$$\hat{G}_i = G_i \odot F(M_{i-1}), \quad (2)$$

where $\odot$ denotes the Hadamard product. Finally, we perform the inverse FFT to covert $\hat{G}_i$ back to the spatial domain and obtain frequency-aware features $W_i$.

3.3 Cross Modality Fusion

Given RGB features $T_i$ and frequency features $W_i$, we use a Cross Modality Fusion (CMF) block to fuse 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 $T_i$ and $W_i$ into $Q, K,$ and $V$ using 1x1 convolutions $\text{conv}_q, \text{conv}_k,$ and $\text{conv}_v$, respectively. Then we flatten them along the spatial dimension to obtain the 2D embeddings $\tilde{Q}, \tilde{K},$ and $\tilde{V} \in \mathbb{R}^{HW/16 \times C}$, and calculate the fused features as:

$$\tilde{M}_i = \text{softmax} \left( \frac{Q \tilde{K}^T}{\sqrt{H/4 \times W/4 \times C}} \right) \tilde{V}. \quad (3)$$
Finally, we employ a residual connection by adding \( \hat{M}_i \) and \( T_i \), and use a 3 x 3 convolution to obtain the output \( M_i \):

\[
M_i = \text{conv}_{3x3}(\hat{M}_i + T_i),
\]

where \( M_i \in \mathbb{R}^{H/4 \times W/4 \times C} \) combines the forgery features in both RGB domain and frequency domain.

We stack the multi-scale transformer, frequency filter and CMF block for \( N \) times (\( N = 4 \) in this paper). The integrated features \( M_{out} \) are calculated by iterative update. Finally, we pass \( M_{out} \) through several convolutional layers to obtain global semantic features \( f \).

### 3.4 Loss functions

**Cross-entropy loss.** We input \( f \) to several fully-connected layers to predict whether the input image is real or fake using a cross-entropy loss \( \mathcal{L}_{cls} \):

\[
\mathcal{L}_{cls} = y \log \hat{y} + (1 - y) \log(1 - \hat{y}),
\]

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 face region as an auxiliary task to enrich the supervision for training the networks. Specifically, we input the feature map \( M_{out} \) to a decoder (stacked convolutional layers and interpolation upsampling layers) to produce a binary mask \( \hat{M} \) in \( \mathbb{R}^{H \times W} \):

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

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 \( \mu_{pos} = \frac{1}{N_p} \sum_{i=1}^{N_p} f_{i, pos} \) 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:

\[
\mathcal{L}_{con} = \frac{1}{N_p} \sum_{i=1}^{N_p} d(f_{i, pos}^p, \mu_{pos}) - \frac{1}{N_n} \sum_{i=1}^{N_n} d(f_{i, neg}^p, \mu_{pos}),
\]

where \( N_p \) denotes the number of negative samples, and \( d \) computes distance with cosine similarity. Finally, combining Eqn. 5, Eqn. 6 and Eqn. 7, the training objective can be written as:

\[
\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{seg} + \lambda_2 \mathcal{L}_{con},
\]

where \( \lambda_1 \) and \( \lambda_2 \) are the balancing hyperparameters. 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 contains a diverse set of samples in different genders, ages, and ethnic groups. We first crop face regions in each video frame using [29], and then generate forged videos using state-of-the-art Deepfake generation techniques. Finally, we use the image harmonization method in [8] 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 [44] and FaceShifter [19], and two facial reenactment methods: First-order-motion [51] and iFace [59]. 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++ [49] dataset. The detailed forgery images generation process is described below.

FSGAN [44] 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 \( G_s \) estimates the swapped face \( I_r \) and its segmentation mask \( S_r \) based on \( I_s \) and a heatmap encoding the facial landmark of \( I_s \), while \( G_r \) estimates the segmentation mask \( S_c \) of the source image \( I_s \). Then the inpainting generator \( G_t \) inpaints the missing parts of \( I_r \) based on \( S_r \) to estimate the complete swapped face \( I_c \). Finally, using the segmentation mask \( S_r \), the blending generator \( G_b \) blends \( I_c \) and \( I_t \) to generate the final output \( I_f \), which preserves the posture of \( I_t \) but owns the identity of \( I_r \). For our dataset, we directly use the pretrained model provided by [44] and inference on our pristine videos.

FaceShifter [33] consists of two networks for full pipeline: AEI-Net for face swapping, and HEAR-Net for occlusion handling. As the author of [33] have not public their code, we use the code from [19] 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 [33] to train the model. **First-order-motion** [51] decouples appearance and motion information for subject-agnostic facial reenactment. Their framework consists 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 [40], 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.

**iFace** [59] 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 [40].

**Post-processing** In order to resolve the color mismatch between the face regions and the background and to eliminate the stitched boundaries, we use DoveNet [8] 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 [18].

### 4.2 Comparisons to current Deepfake Datasets

We summarize the basic information of these existing Deefake datasets and our SR-DF dataset in Table 1. In addition, as mentioned above, how to measure the quality of forged images in these datasets remains 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.

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

| Dataset          | Real Video | Real Frame | Forged Video | Forged Frame |
|------------------|------------|------------|--------------|--------------|
| UADFV [68]       | 49         | 17.3k      | 49           | 17.3k        |
| DF-TIMIT-LQ [30] | 320        | 34.0k      | 320          | 34.0k        |
| DF-TIMIT-HQ [30] | 320        | 34.0k      | 320          | 34.0k        |
| FF++ [49]        | 1,000      | 509.9k     | 4000         | 1,830.1k     |
| DFD [10]         | 363        | 315.4k     | 3,068        | 2,242.7k     |
| DFDC [15]        | 1,131      | 488.4k     | 4,113        | 1,783.3k     |
| WildDeepfake [74]| 3,805      | 440.5k     | 3,509        | 739.6k       |
| Celeb-DF [37]    | 590        | 225.4k     | 5,639        | 2,116.8k     |
| ForgeryNet [23]  | 99.6k      | 1,438.2k   | 121.6k       | 1457.9k      |
| SR-DF (ours)     | 1,000      | 509.9k     | 4,000        | 2,078.4k     |

**Mask-SSIM** First, we follow [37] to adopt the Mask-SSIM score as a measurement of synthesized Deepfake images. Mask-SSIM refers to the SSIM score between the restored faces and the corresponding original image. We use the FaceParsing [18] 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 [41, 69] to measure the similarity between the restored 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 [12] to calculate the perceptual loss between the feature maps of forged faces and that of corresponding real faces. We use the dilib [29] 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 $E_{\text{warp}}$ is used by [4, 25, 32] to measure the temporal inconsistency for video style transfer. We use it to compute the $E_{\text{warp}}$ of consecutive forged frames in different datasets to quantitatively measure the short-term consistency. Following [32], we use the method in [50] to calculate occlusion map and PWC-Net [53] to obtain optical flow. $E_{\text{warp}}$ of different Deepfake datasets are shown in Table 2 for comparison.

**Feature Space Distribution** As can be seen from the above, Celeb-DF dataset [37] 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 [37], FF++ dataset [49], and SR-DF in Figure 3. 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.

Table 2: Average Mask-SSIM scores, perceptual loss, and $E_{\text{warp}}$ values of different Deepfake datasets, with the higher value corresponding to better image quality. For perceptual loss, lower value indicates the better image quality, and for $E_{\text{warp}}$, lower value corresponding to smoother temporal results.

| 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 |
| $E_{\text{warp}}$ ↓ | 73.16 | 69.53 | -   | **49.10** | 56.95 |

5 EXPERIMENTS

5.1 Experimental Settings

**Datasets** We conduct experiments on FaceForensics++ (FF++) [49], Celeb-DF [37], and the proposed SR-DF dataset, and ForgeryNet [23].
**Table 3: Quantitative frame-level detection results on FaceForensics++ dataset under all quality settings. The best results are marked in bold.**

| Methods      | LQ ACC | LQ AUC | HQ ACC | HQ AUC | RAW ACC | RAW AUC |
|--------------|--------|--------|--------|--------|---------|---------|
| Steg.Features [21] | 55.98  | 70.97  | 97.63  | -      | -       | -       |
| LD-CNN [9]   | 58.69  | 78.45  | 98.57  | -      | -       | -       |
| MesosNet [1] | 70.47  | 83.10  | 95.23  | -      | -       | -       |
| Face X-ray [34] | -     | 61.60  | -      | 87.40  | -       | -       |
| F³-Net [46]  | 90.43  | 93.30  | 97.52  | 98.10  | 99.95   | 99.80   |
| MaDD [71]    | 88.69  | 90.40  | 97.60  | 99.29  | -       | -       |
| M2TR (Xception) | 91.07 | 94.25  | 97.28  | 99.05  | 98.79   | 99.24   |
| M2TR (Ours)  | **92.89** | **95.31** | **97.93** | **99.51** | **99.50** | **99.92** |

**Table 4: Frame-level AUC scores (%) of various Deepfake detection methods on Celeb-DF and SR-DF dataset.**

| Methods       | Celeb-DF [37] AUC | SR-DF AUC |
|---------------|-------------------|-----------|
| Xception [49] | 97.6              | 88.2      |
| Multi-task [42]| 90.5              | 85.7      |
| Capsule [43]  | 93.2              | 81.5      |
| DSW-FPA [35]  | 94.8              | 86.6      |
| DCViT [65]    | 97.2              | 87.9      |
| Ours          | 99.8              | 91.2      |
| Avg           | 95.5              | 86.7      |

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 [11], FaceSwap [20], FaceFace [57], and NeuralTextures [56]. 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 consists of 890 real videos and 5,639 Deepfake videos, in which 6,011 videos are used for training and 518 videos are for testing. ForgeryNet dataset is constructed by 15 manipulation approaches and 36 kinds of distortions for post-processing. It contains 99,630 real videos and 121,617 fake videos. We follow ForgeryNet [23] to train our model on the training set, and evaluate on the validation set. 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 as FF++.

When training on FF++ dataset and SR-DF dataset, following [46, 71], 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 [46, 49].

**Evaluation Metrics** We apply the Accuracy score (Acc) and Area Under the RoC Curve (AUC) as our evaluation metrics, which are commonly used in Deepfake detection tasks [34, 43, 46, 49, 71].

**Implementation Details** We use RetinaFace [13] to crop the face regions (detected boxes enlarged 1.3×) 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 backbone network, we use Efficient-b4 [55] pretrained on ImageNet [12]. 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 epochs. We will release code upon publication.

**5.2 Evaluation on FaceForensics++**

FF++ [49] is a widely used dataset in various Deepfake detection approaches [1, 9, 21, 34, 46, 71]. We compare M2TR with top-notch methods on it, including: Steg. Features [21], LD-CNN [9], MesosNet [1], Face X-ray [34], F³-Net [46], and MaDD [71].

We test the frame-level detection performance on RAW, HQ, and LQ of FF++, respectively, and report the AUC scores (%) in Table 3. We can see that our method achieves state-of-the-art performance on all versions (i.e., LQ, HQ, and RAW), which suggests the effectiveness of our approach in detecting Deepfakes of different visual qualities. We also evaluate M2TR using different backbones, and the performances verify that our framework is not restricted by the backbone networks. 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 use 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 [37] 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 [34, 46, 71] because the code and models are not publicly available. The results are reported in Table 4. We observe that our M2TR achieves 99.9% and 90.5% on Celeb-DF 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 [37] 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 4. 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: Quantitative comparison of various Deepfake detection methods on ForgeryNet dataset.

| Methods          | 2 way | 3 way | 16 way |
|------------------|-------|-------|--------|
| MobileNetV3 [24] | 76.24 | -     | -      |
| Xception [7]     | 80.78 | 73.00 | 58.81  |
| P3-Net [46]      | 80.86 | 74.45 | 59.82  |
| GramNet [38]     | 80.89 | 73.30 | 56.77  |
| SNRFilters-Xception [5] | 81.09 | -     | -      |
| Ours             | 82.52 | 75.12 | 69.12  |

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 | Method | FF++ (RAW) | FF++ (HQ) | FF++ (LQ) | SR-DF |
|--------------|-------------|--------|------------|-----------|-----------|-------|
| FF++         | FF++        | P3D [47]         | 80.9       | 75.23     | 67.05     | 65.97 |
|              | Celeb-DF    | R3D [67]         | 96.15      | 95.00     | 87.72     | 73.24 |
|              | SR-DF       | 3D [3]           | 98.23      | 96.70     | 93.18     | 80.11 |
| SR-DF        | FF++        | M2TR mean        | 99.06      | 98.23     | 93.95     | 82.27 |
|              | Celeb-DF    | ST-M2TR          | 99.87      | 99.42     | 95.31     | 85.32 |

5.4 Evaluation on ForgeryNet

ForgeryNet [23] is the most recently released largest scale deepfake detection dataset, which provides three types of forgery labels, i.e., two-way (real / fake), three-way (real / fake with identity-replaced forgery approaches / fake with identity-remained forgery approaches), and n-way (n = 16, real and 15 respective forgery approaches) labels. Using the rich annotations, we conduct two-/three-/n-way classification experiments. The comparison results in Table 5 demonstrate that M2TR outperforms state-of-the-art methods on classification tasks of different granularity. In particular, the most remarkable performance gain i.e., 15.5% compared with F3-Net, is achieved on the most challenging 16-way classification, which is benefited from the capability of our method to identify the multi-scale forgery features.

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 [47], which simplifies 3D convolutions with 2D filters on spatial dimension and 1D temporal connections; (2) R3D [67], 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 6. We can see that our method achieves the best performance on FF++, ForgeryNet, and SR-DF.

5.6 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 [71] to sample 30 frames for each video and calculate the frame-level AUC scores. The comparison results are shown in Table 7. 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 7 demonstrate that our method achieves better generalization than most existing methods.

5.7 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 Frequency Filter (FF) is utilized to capture the subtle forgery traces in frequency domain. To evaluate the effectiveness of MT and FF, we remove them separately from M2TR and demonstrate the performance degradation on FF++. We also replace the Cross Modality Fusion (CMF)
blocks with naive concatenation operations to verify the usefulness of CMF for feature fusion.

The quantitative results are listed in Table 8, which validates that the use of MT, FF and CMF can effectively improve the detection performance of our model. In particular, the proposed frequency filter 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., 3.9%, 1.9%, and 0.7% higher than 40 × 40, 20 × 20 and 10 × 10 single-scale transformer on AUC score. This confirms the use of a multi-scale transformer is indeed effective.

Effectiveness of the Contrastive Loss To illustrate the contribution of contrastive loss in improving the generalization ability of our method, we also 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 observe that 1) When training on FF++ without the contrastive loss, the accuracy decreases by 3.8% and 5.2% in Celeb-DF and SR-DF, respectively. 2) When training on SR-DF dataset without the contrastive loss, the accuracy decreases by 5.8% and 3.0%, respectively.

## 6 CONCLUSION
In this paper, we presented a two-stream network Multi-modal Multi-scale Transformer (M2TR) for Deepfake detection, which uses multi-scale transformers to capture subtle local inconsistency at multiple scales and frequency filters to improve the robustness against image compression. Forgery feature from two streams are adaptively fused through cross modality fusion blocks. 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 of the proposed method.

## Acknowledgement
This work was supported by National Natural Science Foundation of Project (62072116). Y.-G. Jiang was sponsored in part by “Shuguang Program” supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission (20SG01).

## 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] Dongdong Chen, Xiaotao Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. 2021. Local Relation Learning for Face Forgery Detection. In AAAI.
[6] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. 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 BMASec.
[8] DeepFake Detection. Dataset. 2019. http://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html.
[9] Deepfakes. 2018. https://github.com/deepfakes/faceswap.
[10] 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.
[11] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. 2020. Retinaface: Single-shot multi-level face localisation in the wild. In CVPR.
[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL-HLT.
[13] Brian Dollhansky, Russ Howes, Ben Pluam, Nicole Baram, and Cristiano Canton Ferrer. 2020. The deepfake detection challenge (dffc) dataset. arXiv preprint arXiv:2006.07097 (2020).
[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In ICLR.
Special Session 2A: Transformer-based Multimedia Understanding: Model Design, Learning, Distillation