Hindawi
Security and Communication Networks
Volume 2022, Article ID 9278715, 15 pages
https://doi.org/10.1155/2022/9278715

Research Article

FD^{2}Foremer: Thinking Face Forgery Detection in Midfrequency Geometry Details

Wentao Li[1] and Zhidong Shen[1,2]

[1]*Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of CyberScience and Engineering, Wuhan University, Wuhan 430079, China*
[2]*Engineering Research Center of Cyberspace, Yunnan University, 650000 Yunan, China*

Correspondence should be addressed to Zhidong Shen; shenzd@whu.edu.cn

Received 10 June 2022; Accepted 22 August 2022; Published 29 September 2022

Academic Editor: Feng Ding

Face forgery by DeepFake has caused widespread concern in community because of the synthesized media's risks to the society. However, advances in recent years have been able to produce synthetic images indistinguishable from real images in the RGB space. Extracting midfrequency facial geometry details, including person-specific details and dynamic expression-dependent ones on facial geometry surfaces, is a promising way to highlight forgery clues during face forgery detection. In this paper, we use 3D face reconstruction to generate the displacement map from a single input face image, which is able to represent middle and fine scale details by indicating signed distance from the point in UV space. The cropped face images can also provide eyes and mouse information, so we use face image and its displacement map to extract the image features. Besides, we save the computation cost and maintain competitive performance using a universal transformer architecture and introduce a manifold distillation strategy to train our model from a more complex transformer backbone. Extensive experiments on various public DeepFake datasets indicate the effectiveness of the extracted facial geometry details, and proposed method achieves competitive performance.

1. Introduction

The development of deep learning and the availability of large-scale datasets have led to powerful deep generative models, which enables the facial manipulation and images generation. Deep generative models can promote the emerging development of entertainment and cultural industry but can also be used for malicious purposes. One such application of the generative models is DeepFake [1]. DeepFake (e.g., fake images, audios, and videos) has become a real threat to our society due to its realism and impact scopes. It can be used to mislead public opinion, disrupt social order, and even threaten face recognition system [2], intervene in government elections [3], and subvert state power [4]. It has become the most advanced new form of cyberattacks.

In this work, we are committed to detecting the facial manipulation, related to the dataset of FaceForensics++ [5], Celeb-DF [6], and DFDC [7], which is difficult to discover the complex artifacts from facial appearance only. At the same time, lots of researchers have been driven to focus on extracting manipulation artifacts from other perspectives besides the RGB space.

Some researches [8, 9] extract forgery artifacts using both different low-level local parts and high-level semantic features. The assumption is that the object details should include rapidly changing motion part and subject identities. For example, Feichtenhofer et al. [8] used a two-pathway SlowFast model for video recognition. One pathway is designed to capture semantic information that can be given by images or a few sparse frames. The other pathway is responsible for capturing rapidly changing motion. Zhao et al. [9] believe the difference between the real and fake images in face forgery detection is often subtle and local. So, they use multiple spatial attention heads to make the network attend to different local parts and use textural feature

enhancement block to zoom in the subtle artifacts in shallow features. They further use the attention maps to aggregate the low-level textural feature and high-level semantic features. However, these details need to be extracted in multiframes, and we do not know how these details effect the faces from which extracted.

In our work, for face forgery detection, we try to extract facial geometry details that are often connected with high frequency including person-specific details and dynamic expression-dependent ones (e.g. deeper skin wrinkles and creases) through 3D faces reconstruction [10]. First, a tentative 3D face reconstruction is conducted with the help of FLAME [11]. 2D image is encoded into a latent code, which consist of albedo parameters α, geometry parameter θ, ψ, β lighting coefficient l, and camera parameters c. A coarse geometry reconstruction can be achieved with the help of above latent code. For the supplement of identity shape details, we then embed another coefficient δ with dimension of 128, representing identity-specific information, to extract geometry details. With the help of a neural network, a displacement map is produced using expression ψ and subject-specific details δ, to augment FLAME mesh with facial geometry information, which refines the low-frequency geometry with higher frequency information. Though the detail reconstruction image lost some high-frequency clues, differing it a little bit from the original face. Displacement map still captures person-specific and dynamic expression-dependent details that often lost in DeepFakes, so we call it "midfrequency" geometry details to help us with the detection.

When detecting forgery details with neural networks, we consider using transformer [12] architecture with low computation cost and high efficiency. Good balance of computation cost and high efficiency is often tough to achieve. Some architectures require extremely high computational resources, such as the popular ViTs are heavyweight, harder to optimize [13], and need L2 regularization to prevent overfitting [14, 15]. So, we use the "MetaFormer" [16] structure with pooling operation for transformer architecture and propose a Forgery-Detection-with-Facial-Detail Transformer (FD^{2}Foremer). To bring the light FD^{2}Foremer with good performance, we introduce the manifold learning during training. We first pretrained swin transformer architecture with forgery details and face images as inputs. Then, the simplified patch embedding manifold loss [17] is used to provide the appropriate constraints on params of the light FD^{2}Foremer with pretrained netw4ork as teacher network.

(i) In summary, our contributions can be summarized in three-fold:

(ii) We start with 3D face reconstruction during forgery detection and output facial geometry details for subtle artifacts capture.

(iii) We introduce the "MetaFormer" architecture into our network and propose a light transformer FD^{2}Foremer with face displacement maps for DeepFake detection.

(iv) We introduce the manifold learning during the light network training. The experimental results on three different public datasets show that our method achieves competitive performance.

2. Related Work

Forgery creation, of particular interest in faces, has recently received a lot of attention given its widespread use. To eliminate the risks of misleading forged faces, face forgery detection becomes an increasingly emerging field of research. In this section, we provide a brief overview of several studies have been proposed relevant to our work.

2.1. Conventional Image Forgery Detection. Though several techniques have been proposed in the past decades to detect forgery in digital images, those conventional techniques cannot handle the detection of artifacts produced by neural networks well [18–23]. First, forgery is assumed to be done using linear or cubic interpolation by conventional techniques in most cases. Besides, recent advanced forgery techniques leave almost no visible artifacts on tampered faces, which can easily fool sensitive conventional detectors. Furthermore, face forgeries are much smaller and have typical shape, which requires specialized treatments.

2.2. Image Forgery Detection with Neural Networks. Nataraj et al. [24] used cooccurrence matrices to exhibit discriminative features of manipulated regions in boundaries shared with neighboring non-manipulated pixels. Their method passed the cooccurrence matrices through a CNN-LSTM model, allowing the network to learn important cooccurrence matrices essential features. Qian et al. found the awareness of frequency, especially under the compression condition, could be a cure. So, they applied frequency-aware decomposition and local frequency statistics on DeepFake detections, finally achieved outstanding performance on low quality media. Zhao et al. [9] used a multi-attentional DeepFake detection network to treat face forgery detection as a fine-grained classification problem, mainly focused on different local parts and the subtle artifacts in shallow features. Zhou et al. [25] extracted tampering artifacts and local noise residual features by exploring steganalysis features. However, these methods extract artifacts from pixel levels or image features, i.e., only considering exploring the synthesis clues from the facial appearance.

There is also research considering exploring forgery clues among the shape, pose, and the lighting condition of the head. Yang et al. [26] confirmed the significant difference in the estimated head poses in DeepFakes, by comparing head poses estimated from 2D landmarks in the real and faked parts of the face. De Carvalho et al. [27] proposed to explore forgery clues from the 2D illuminant maps of the image segments considering the inconsistency. Zhu et al. [28] disentangled the face image into common texture, identity texture, 3D shape, ambient light, and direct light roughly. And the identity texture and direct light are combined as the facial detail to be fed into a neural network. However, these

methods are difficult in recovering small facial details from the input image due to the limited representation power and the extensive calculation.

2.3. Transformer Architecture. Transformers are first proposed to learn long-range sequential dependence for translation tasks [12] and then get widely used in numerous natural NLP tasks. With large-scale unlabeled text corpus, transformers achieve amazing performance in language pretraining tasks [29, 30]. Motivated by the success of transformers in NLP, the attention mechanism and transformers are applied to deal with vision tasks, such as image classification [31], object detection [32], image segmentation [14], and image captioning [33]. Notably, Chen et al. introduced iGPT [34], where a transformer is applied to image pixels after reducing image resolution and color space for self-supervised learning. Google proposed a visual transformer (ViT) that achieved state-of-the-art performance on ImageNet classification [31]. They show that ViT need pretraining on large datasets, such as ImageNet-22k and JFT-300M, and huge computation resources to achieve excellent performance in supervised image classification tasks. In 2020, Touvron et al. [14] proposed DeiT with adjusted network architecture, trying to tackle the data-inefficiency problem through data augmentation and knowledge distillation. However, the good performance comes at a high computational cost. To save the computational cost, we shift our attention to the architecture of transformers and what is responsible for the success of the transformers.

3. Extract Facial Geometry Detail though 3D Reconstruction

We regard the face forgery detection problem as a binary end-to-end classification task about extracted features. The motivation of our work lies in the fact that geometry details (i.e. wrinkles) of individual faces are related with some unchanged identity details, whereas individual expressions affect details either, which contributes to the face forgery detection. Consequently, we extract geometry details in a trial-and-error way during 3D face reconstruction, which consist of expression-related dynamic information, such as wrinkles [35] and identity-specific static information. Dynamic geometry information is often influenced by all kinds of expressions that different for the same individuals, whereas static identity information varies cross different humans. By reconstructing 3D faces, a kind of facial geometry details called displacement map is inferred from both dynamic expressional domain and static identity-specified domain for forgery detection (see Figure 1).

3.1. 3D Faces Reconstruction. With the help of FLAME, we use an analysis-by-synthesis method to reconstruct 3D face of the input image: a latent code is regressed using an input image I. Then another image I_r is synthesized by encoding the latent code. As shown in Figure 2, the fully connected layer is connected with a ResNet50 [36] network as the encoder E_c for the regression of the latent code. To synthesize images later, the latent code is divided into 50 albedo parameters α, 100 shape coefficient β, 15 pose parameters θ, 50 expression parameters ψ, 21 lighting coefficients l , and camera parameters c. In total, E_c predicts a 236-dimensional latent code.

Then, we use FLAME to reconstruct 3D mesh coarsely. FLAME [11] is a 3D statistical face model that can construct the mesh with number of 5023 vertices. Given the pose parameters $\theta \in \mathbb{R}^{15}$, shape coefficient $\beta \in \mathbb{R}^{|\beta|}$, and expression parameters $\psi \in \mathbb{R}^{|\psi|}$, we can express the mesh as

$$M(\beta, \theta, \psi) = W\left(J(\beta), T_P(\beta, \theta, \psi), \mathscr{W}, \theta\right), \tag{1}$$

where $W(J, T, \mathscr{W}, \theta)$ indicates the blend skinning function, $J \in \mathbb{R}^{3k}$ are joints and $T \in \mathbb{R}^{3n}$ presents vertices that need rotated, $\mathscr{W} \in \mathbb{R}^{k \times n}$ indicates the weights used for linear smoothing. Then,

$$T_P(\theta, \beta, \psi) = B_P(\theta; P) + B_s(\beta; \mathcal{S}) + B_E(\psi; \mathrm{E}) + T, \tag{2}$$

indicates that we add pose correctives $B_P(\theta; P)$: $\mathbb{R}^{3k+3} \longrightarrow \mathbb{R}^{3n}$, shape blendshapes $B_s(\beta; \mathcal{S}) : \mathbb{R}^{|\beta|} \longrightarrow \mathbb{R}^3$and expression blendshapes $B_E(\psi; \mathscr{E})$: $\mathbb{R}^{|\psi|} \longrightarrow \mathbb{R}^{3n}$, to the no-poses template T, controlled by the pose P, learned identity $\mathcal{S}$, and expression bases $\mathscr{E}$. More details can be found in [11].

To extract face texture, the Basel Face Model [37] is adopted and converted into FLAME UV space for the consistence with the FLAME mesh. Given the albedo coefficient $\alpha \in \mathbb{R}^{|\alpha|}$, the albedo int UV space $A(\alpha) \in \mathbb{R}^{d \times d \times 3}$ are output using Basel Face Model. As for the camera settings, orthographic model is employed for the projection of FLAME meshes based on the assumption that individual faces are shot at a distance. The projection of meshes can be expressed as:

$$v = s\Pi\left(M_i\right) + t, \tag{3}$$

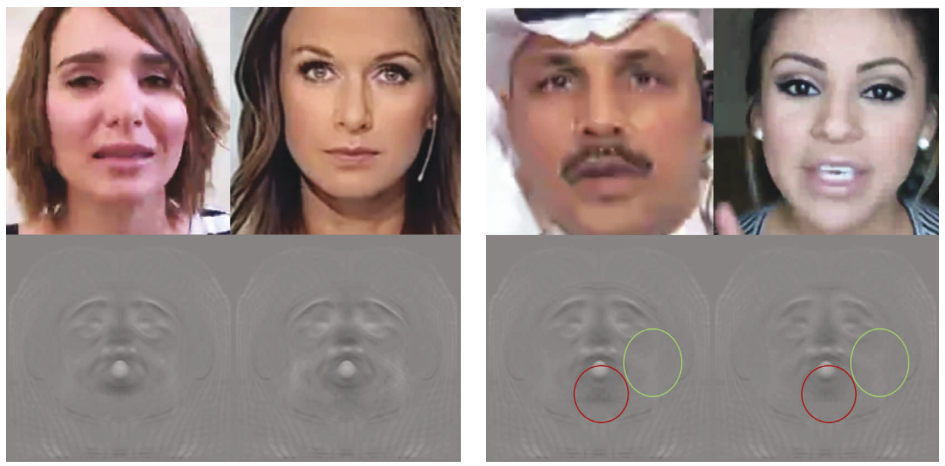
where the vertex $M_i \longrightarrow \mathbb{R}^3$ is among M, $\Pi \in \mathbb{R}^{2 \times 3}$ are weights that project 3D vertexes into 2D image space orthographically, $t \in \mathbb{R}^2$ indicates 2D translation and $s \in \mathbb{R}$ denotes isotropic scale.

We use the most frequently-employed Spherical Harmonics (SH) [38] as illumination model. Assuming the Lambertian facial reflectance and distant light source, we shade the face images as the equation below:

$$B\left(\alpha, \mathrm{I}, N_{uv}\right)_{i,j} = A(\alpha)_{i,j} \odot \sum_{k=1}^{9} \mathrm{I}_k H_k\left(N_{i,j}\right), \tag{4}$$

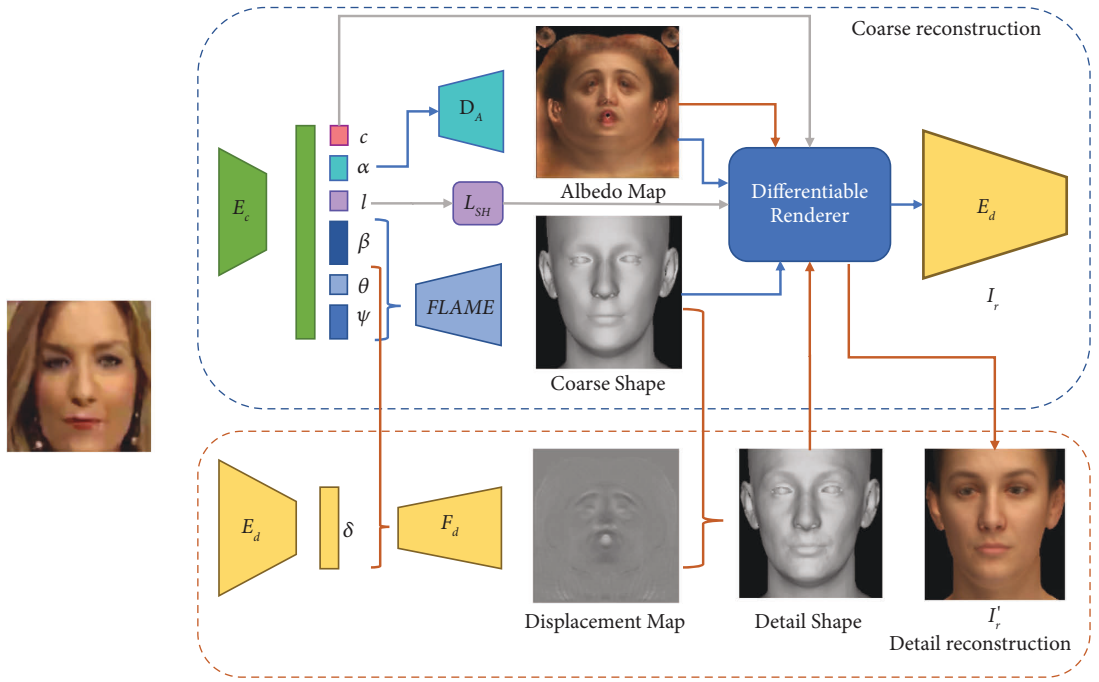
where shaded texture $B_{i,j} \in \mathbb{R}^3$, albedo $A_{i,j} \in \mathbb{R}^3$, and surface normal $N_{i,j} \in \mathbb{R}^3$ corresponds to a specific pixel (i, j) of UV images. The SH coefficients are defined as $\mathrm{I} = [\mathrm{I}_1^T, \ldots, \mathrm{I}_9^T]^T$, with $\mathrm{I}_k^T \in \mathbb{R}^3$, and H_k: $\mathbb{R}^3 \longrightarrow \mathbb{R}$ means basis . $\odot$ denotes the Hadamard product. The blue box in Figure 2 shows the coarse reconstruction branch for input face image.

3.2. Facial Geometry Details Generation. The analysis-by-synthesis method utilizes the latent code of numerous expressions, poses, and shapes to reconstruct the face with the



(a) (b)

Figure 1: Midfrequency geometry details on facial surfaces during face reconstruction. The uniformity of gray scale in green line area is greater than the same area in real images, which indicates that displacement map extracts less geometry details from fake images. And the unnatural concentration of gray scale in red line circles different themselves from the displacement map of real images. (a) Real images and their midfrequency facial geometry details, (b) fake images and their midfrequency facial geometry details.



c: camera code
α: albedo code
l: light code
β: shape code
θ: pose code
ψ: expression code
δ: detail code

Figure 2: Extracting facial geometry details during 3D faces reconstruction. During coarse reconstruction, FLAME is used to reconstruct coarse geometry, whose representational power is limited by the low mesh resolution. Then, the convolutional networks are used to produce the displacement map with dynamic expression-related information and static identity-specific information as inputs. The displacement map augments 3D faces reconstruction, which refines the low-frequency geometry with higher frequency information, regarded as clues for face forgery detection.

help of FLAME model. But the small number of FLAM vertices and faces limits the representational power of the model, and therefore, FLAME mesh surface almost ignores the middle and fine scale details (see Figure 2). Managing to present dynamic expression-related information and static identity-specific person-specific details and expression-

Figure 3: The midfrequency geometry details on FLAME's surfaces with help of displacement maps.

dependent details for manipulation detection, we conduct the detail reconstruction [39] to produce displacement maps augmenting low-frequency geometry FLAME with higher frequency details. The pixel values of the displacement map control the signed distance from the point on base mesh to its corresponding point. We locate the surface points of FLAME mesh corresponding to the pixels in the displacement map, then inverse-project the points to the raw mesh along normal direction to find its corresponding points. The detail shape in Figures 2 and 3 shows the effect of our generated displacement maps. Though not perfect and lacking some high frequency information, displacement

map representing person-specific and dynamic expression-dependent details is still helpful. So, we regard the displacement maps as "midfrequency" geometry details to help with DeepFake detections.

The displacement map in UV coordinate augments the coarse FLAME mesh with facial geometry details by shifting shape points of FLAME mesh. Similar to the coarse reconstruction, another 128-dimension detail code δ is regressed from image I using an encoder E_d the same architecture as E_c. We then concatenate the jaw pose parameters θ_{jaw}, FLAME's expression parameters ψ with latent code δ, and use detail decoder F_d to generate D:

$$D = F_d\left(\theta_{jaw}, \psi, \delta\right), \tag{5}$$

where the jaw pose parameters $\theta_{jaw} \in \mathbb{R}^3$ and FLAME expression $\psi \in \mathbb{R}^{50}$ indicate the dynamic expression-related information, meanwhile $\delta \in \mathbb{R}^{128}$ is the latent code representing the static identity-specific information. We then convert the D into a normal map for rendering.

The geometry displacement map makes it possible to reconstruct 3D face with midfrequency information. Converted into UV space, $M_{uv} \in \mathbb{R}^{d\times d\times 3}$ and its surface normal $N_{uv} \in \mathbb{R}^{d\times d\times 3}$ are combined with D to augment the mesh with geometry details as

$$M'_{uv} = M_{uv} + D \odot N_{uv}. \tag{6}$$

Applying normal map N' from M', the synthesis image I'_r is rendered as

$$I'_r = \mathscr{R}\left(M, B\left(\alpha, \mathrm{I}, N'\right), c\right). \tag{7}$$

Comparing the rendered detailed image with the real image, the decoder F_d is forced to model detailed geometric information, with the help of the coarse reconstruction on VGGFace2 [40], BUPT-Balancedface [41], and Vox-Celeb2 [42]. As shown in Figure 2, midfrequency details in rendered images I'_r, including both dynamic expression-related information and static identity-specific information, are inferred mainly from the displacement map, which is exactly what we need for forged faces detection. We call the encoder E_c, encoder E_d and decoder E_d together as facial detail generator G_d to produce midfrequency facial geometry details.

4. Methodology

In the following sections, we propose the Forgery-Detection-with-Facial-Detail Transformer (FD^{2}Former), including backbone to extract image features, the introduced transformer architecture and the fine-grained manifold distillation strategy.

4.1. Backbone. We employ face recognition [43] DL libraries to detect and crop faces frame by frame. And the facial geometry detail generator G_d is used to generate middle and fine scale details for face images. The existing methods using an alignment only centralizes the face without considering whether the face is frontolized, which easily leads to facial information loss. With the facial detail displacement map converted to UV space, the facial geometry details for all the faces can be located in the same spatial space. Since the pixels of UV displacement maps corresponding to full 3D face mesh, there is no information loss. Aligned face images can also provide pose, eyes blink and mouth movement information that cannot be perceived in the detail displacement map, so we use both face image and its detail displacement map to extract the face manipulation clues. In order to learn more facial movement information and facial geometry details, we choose convert inputs into much more informative high level image features rather than image patches directly. The aligned face images and the displacement map are concatenated and then fed into CNN backbone to extract high level image features. We employ a ResNet as the high-level image features extractor.

4.2. Forgery-Detection-with-Facial-Detail Transformer. Figure 4 illustrates the architecture of the proposed Forgery-Detection-with-Facial-Detail Transformer for DeepFake detection. From the perspective of transformers introduction [12], many works have paid great attention to attention and focused on designing various attention-based token mixer components to achieve good performance. However, the good performance comes at a high computational cost, and these works pay little attention to the general architecture.

Considering what makes it effect for the success of transformers, we use "MetaFormer" concept for our work. MetaFormer is a general architecture abstracted from numerous transformers [12], where the most components remain the same as transformers, but the token mixer is not specified. MetaFormer first apply input embedding to the input I, such as patch embedding for ViTs [31]:

$$X = \mathrm{InputEmb}\left(I\right), \tag{8}$$

where $X \in \mathbb{R}^{N\times C}$ denotes the embedding tokens with sequence length N and embedding dimension C.

Then, embedding tokens are fed into several MetaFormer blocks, each of which consists of two residual subblocks. Specifically, the first subblock with a token mixer is usually designed to mix token information and can be expressed as

$$Y = \mathrm{TokenMixer}\left(\mathrm{Norm}\left(X\right)\right) + X, \tag{9}$$

where $\mathrm{Norm}\left(\cdot\right)$ means the normalization such as Layer Normalization [1]; $TokenMixer\left(\cdot\right)$ denotes a module mainly working for communicating information among tokens. In recently works, vision transformer models [31, 44, 45] and spatial MLP in MLP-like models [46, 47] have implemented various kinds of attention mechanism, which aims at mainly propagating token information and some mixing channels, like attention.

The second subblock includes a two-layered MLP with nonlinear activation function,

$$Z = \sigma\left(\mathrm{Norm}\left(Y\right)W_1\right)W_2 + Y, \tag{10}$$

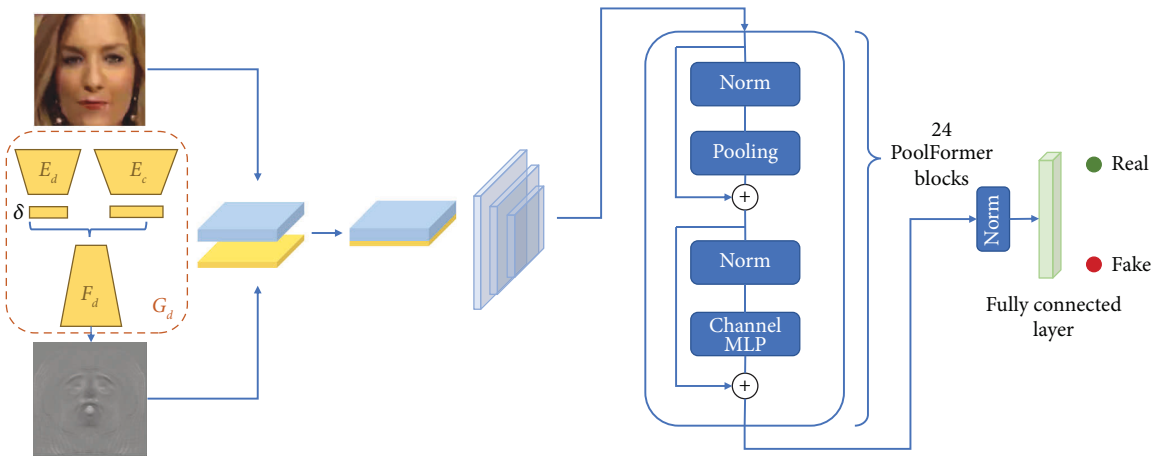


FIGURE 4: The architecture of the proposed forgery-detection-with-facial-detail transformer, including the cropped face image and their corresponding displacement map as input, convolutional networks as backbone for image feature extraction and 24 transformer blocks for feature learning.

where $W_1 \in \mathbb{R}^{C\times rC}$ and $W_2 \in \mathbb{R}^{C\times rC}$ are learnable parameters with MLP expansion ratio r; $\sigma(\cdot)$ is a nonlinear activation function, such as GELU [48] or ReLU [49].

We believe that such a general architecture contributes mostly to the success of the recent transformer and MLP-like models. To decrease the number of learnable parameters and save computational costs, we employ the simple operator, pooling, as the token mixer for Forgery-Detection-with-Facial-Detail Transformer.

Assuming the input T is in channel-first format, $T \in \mathbb{R}^{C\times H\times W}$, the pooling operator can be expressed as

$$T'_{:,i,j} = \frac{1}{K\times K}\sum_{p,q=1}^{K} T_{:,i+p-\frac{K+1}{2},\, i+q-\frac{K+1}{2}} - T_{:,i,j}, \quad (11)$$

where K means the pooling size. For the consistency with the residual connection in FD^{2}Former block, subtraction of the input itself is added in (11). The PyTorch-like code of the pooling is shown in Algorithm 1.

Unlike self-attention and spatial MLP that have computational complexity quadratic to the number of tokens, the pooling operation acquires a computational complexity linear to the sequence length without any learnable parameters. The overall framework of the FD^{2}Former transformer part has 4 stages with $H/4\times W/4$, $H/8\times W/8$, $H/16\times W/16$, and $H/32\times W/32$ tokens, respectively, where H and W represent the height and width of the image features. The model has embedding dimensions of 64, 128, 320, and 512 responding to the four stages. And FD^{2}Former has 24 blocks in total, where stages 1, 2, 3, and 4 contain 4, 4, 12, and 4 FD^{2}Former blocks, respectively. The MLP expansion ratio is set as 4. Not surprisingly, the FD^{2}Former of "MetaFormer" architecture with 24 blocks have fewer parameters (29M) than the same one of "Swin-S" architecture [50] with 24 blocks.

```
import torch.nn as nn
  class Pooling(nn.Module):
    def __init__(self, pool_size = 3):
      super().__init__()
        """
        Padding size is set as half of pool size.
        """
        self.pool = nn.AvgPool2d(
          pool_size, stride = 1
          padding = pool_size//2,
          count_include_pad = False
        )
      def forward(self, x):
        """
        [B, C, H, W] = x.shape
        Subtraction of the input feature is added,
      considering the residual connection of the transformer
      blocks.
        """
        return self.pool(x) – x
```

4.3. Fine-Grained Manifold Distillation. To fully excavate the strong capacity, the fine-grained manifold distillation strategy is used to train the proposed FD^{2}Former with the pretrained same FD^{2}Former but of swin transformer block as the teacher. Since FD^{2}Former of pooling operation can get better limitation during training, with a teacher network having more complex structure (see Figure 5).

For an appropriate constraint during model training, knowledge distillation should not only focus on distilling the output logic [14] but also consider the intermediate features images and their relationship. A natural thought about transferring feature maps may be a workable way, but its harsh conditions for the selection of teacher model cannot be ignored, which requires student and teacher models to have the same feature embedding dimension. Besides, it does not make use of interpatch information.

We use the fine-grained manifold distillation to lift the limitation about the number of heads and dimension of embedding features during distillation. Specifically, given a

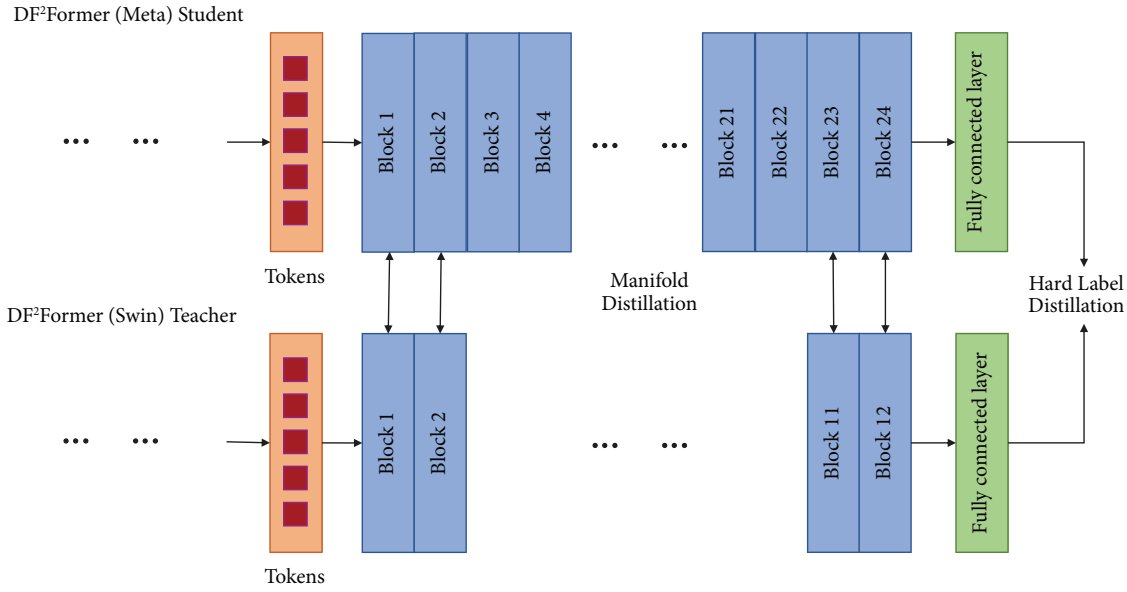


Figure 5: The schematic illustration of the distillation outline. The manifold distillation loss is plugged in the selected blocks of student and teacher DF^{2}Former.

batch of images that described as $(x^1, y^1), \ldots, (x^n, y^n)$, a single image can be divided into p patches. c_S and c_T indicates the patch embedding dimension for student and teacher DF^{2}Former. And feature maps between student or teacher transformer blocks can be expressed as $F_T \in \mathbb{R}^{n\times p\times c_T}$ and $F_S \in \mathbb{R}^{n\times p\times c_S}$. For the student DF^{2}Former, the patch-level manifold of feature maps F_S is produced through the calculation of relative distance as:

$$\mathscr{M}(F_S) = \gamma(F_S')\gamma(F_S')^T, \tag{12}$$

where γ is the conversion which reshape $\mathbb{R}^{n,p,c} \longrightarrow \mathbb{R}^{np,c}$ and $F_S'[i,j,:] = F_S[i,j,:]/F_S[i,j,:]_2$ means the embedding normalization. But should not be ignored is the unbearable load of computation for such a kind of calculation, if we compute the patch-level manifold gap between teacher and student as above. Since the batch size, number of patches and the embedding dimension of each patch lead to a good many calculations. Thus, the patch embedding manifold should be further simplified, which decomposed into two relations sample terms and one random sample error correction term. The manifold loss is simplified as:

$$L_{manifol\ d-sp} = \alpha L_{cp} + \beta L_{rs} + L_{ci}, \tag{13}$$

where L_{cp} means cross-patch manifold loss, L_{ci} indicates cross-image loss and L_{rs} means random-selected loss. α and β are weights that balance the contribution of these terms. Three terms are expressed as:

$$\begin{aligned} L_{ci} &= \sum_{k=0}^{p} \frac{\mathscr{M}(F_T[:,k,:]) - \mathscr{M}(F_S[:,k,:])_F^2}{p} \\ L_{cp} &= \sum_{s=0}^{n} \frac{\mathscr{M}(F_T[s,:,:]) - \mathscr{M}(F_S[s,:,:])_F^2}{n}, \\ L_{rs} &= \mathscr{M}(F_T^r) - \mathscr{M}(F_S^r)_F^2 \end{aligned} \tag{14}$$

where F_T^r and F_S^r are random selection from F_T and F_S, whose dimensions are (k, c_S) and (k, c_T). The number of randomly selected patches is controlled by k. Figure 6 illustrates the meaning of terms L_{ci}, L_{cp} and L_{rs}.

Considering the effectiveness of the hard-label distillation, the simplified patch embedding manifold loss is combined with the former to properly limit the student FD^{2}Fomer during training:

$$\begin{aligned} L_{total} = \sum_{l} L_{manifol\ d-sp} &+ \frac{1}{2}H(\psi(f_s(X)), y) \\ &+ \frac{1}{2}H(\psi(f_s(X)), y_t), \end{aligned} \tag{15}$$

where l means the selected blocks for the insertion of simplified manifold loss.

5. Experiments

In this section, we first introduce benchmark datasets and details about implementations. Also, we conduct a set of ablation studies, and compare our method with previous works. We will describe the datasets and implementation details in Section 5.1. The ablation studies and the analysis of the manifold learning strategy are described in Section 5.2 and Section 5.3. We will analyze the experimental results compared with the previous work in Section 5.4.

5.1. Datasets and Implementation Details

5.1.1. Training Dataset. Faceforensics++ (FF++) [5] is a dataset released to standardize the evaluation of face forgery detection methods, which includes 1000 original videos and other 4000 manipulated videos. The manipulated videos are generated using four typical face swapping and reenactment methods, i.e., DeepFakes (DF) [51], FaceSwap (FS) [52],

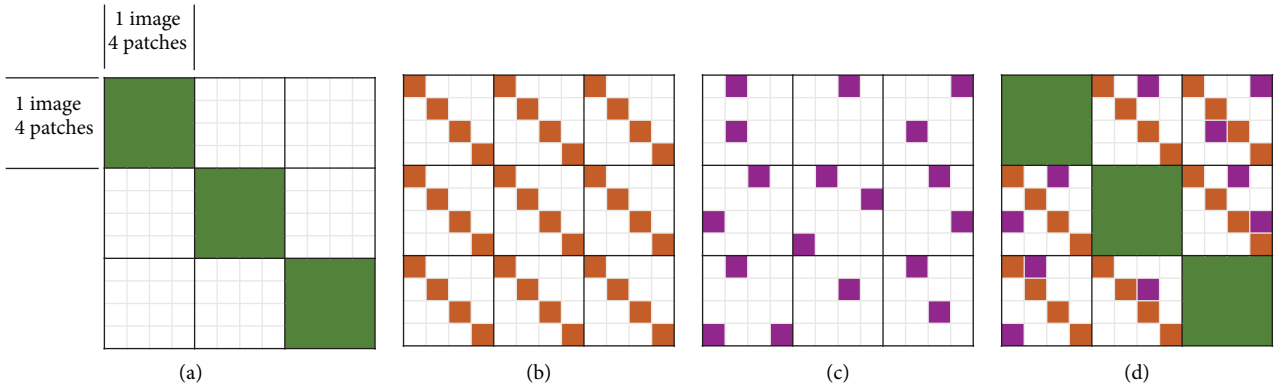


FIGURE 6: The diagram for computing simplified manifold loss. For the convenience of display, three feature maps are involved and each feature map is divided into four patches. (a) Cross-patch. (b) Cross-image. (c) Random-selected. (d) Patch embedding.

Face2Face (F2F) [20], and NeuralTextures (NT) [53]. Besides, there are three versions of FF++ in terms of compression level, i.e., raw, lightly compressed (HQ), and heavily compressed (LQ). Higher the compression level, the harder it is to distinguish the forgery traces. Since the uploaded manipulated videos always have a limited quality, we use the LQ versions in most experiments. We sample 270 frames for each train video and 100 frames for each test one.

5.1.2. Test Datasets. The following datasets are adopted for evaluation. (1) the testing set of FF++ as described. (2) DeepFakes Detection Challenge (DFDC) dataset [7] containing a total of 123,546 videos with the help of paid actors. Each video lasts about 10 seconds and consists of 300 frames. (3) The Celeb-DF dataset [6] containing 408 real videos and 795 synthesized video sequences with reduced visual artifacts, released for the advance of research on manipulated face detection. The examples of data are shown in Figure 7.

5.1.3. Implementation Details. For the facial geometry detail generation, we acquire the detail displacement map using facial geometry detail generator G_d. For the neural network, we train the FD^{2}Former of swin transformer block rather than pooling operation as the teacher. Then, we use the simplified patch embedding manifold loss to teach FD^{2}Foremer of "MetaFormer" architecture with pretrained network. The Adamw optimizer is utilized for training with the initial learning rate of 5×10^{-4} and the warm-up learning rate is 5×10^{-7}. The batch size is set to 64 and weight decay equals to 0.05, $\beta 1 = 0.9$, $\beta 2 = 0.999$. We use a cosine decay learning rate scheduler and 5 epochs of linear warm-up. In our implementation, the total epoch is 30.

5.2. Ablation Studies. For the analysis of the Forgery-Detection-with-Facial-Detail Transformer, facial geometry details and face image are regarded as complementary inputs and a transformer is utilized for the forged face detection. To evaluate each input, we quantitatively evaluate FD^{2}Former with different inputs: a transformer with facial geometry details as input only, a transformer with cropped face image only, and a transformer with both original images and facial geometry details as input. The results are listed in Table 1.

First, the transformer only detects geometry details or cropped face achieves similar results. However, the model performs better with both face image and facial geometry details as input. Besides, the "MetaFormer" architecture achieves the similar and competitive results compared with the swin transformer architecture, but it has less learnable parameters, which saves training resources. Shown as Figure 8, FD^{2}Former(swin) almost saves half learnable params of CViT, while the accuracy only decreases by 0.53%.

We are also interested in the performance of the manifold learning strategy, compared with the hard-label distillation method. As shown in Table 1, the manifold learning strategy is more efficient than the hard-label distillation strategy. The distilled FD^{2}Former using our method outperform the model using hard-label distillation by 1.45% on Celeb-DF. The potential negative impact of the manifold learning strategy may be the increased consumption of computation resources and energy.

5.3. Analysis of the Fine-Grained Manifold Distillation. Aimed at patch-level knowledge distillation, we can insert the simplified manifold loss into any blocks under the condition of same patch numbers at corresponding blocks. As shown in Table 2, the experiments are conducted to find the better insert location of the manifold loss. Results show that applying manifold loss at both the last stage and first stage improves the performance of student network better. We think it is because such a kind of insertion constrains the student network properly, while the student's capacity could be limited if losses inserted in middle of the model.

We test different values of hyper-parameters α and β, and presents results in Tables 3 and 4. The experiments show that assigning 1 and 0.2 to α and β improves the performance relatively. As Figures 9 and 10 shown, a small β could be more efficient because L_{rs} may partially coincide with L_{cp}, L_{ci} terms, meanwhile the uncertainty of sample mechanism should be controlled.

We also test the hyper-parameter k, which means the number of randomly selected patches for calculation of the

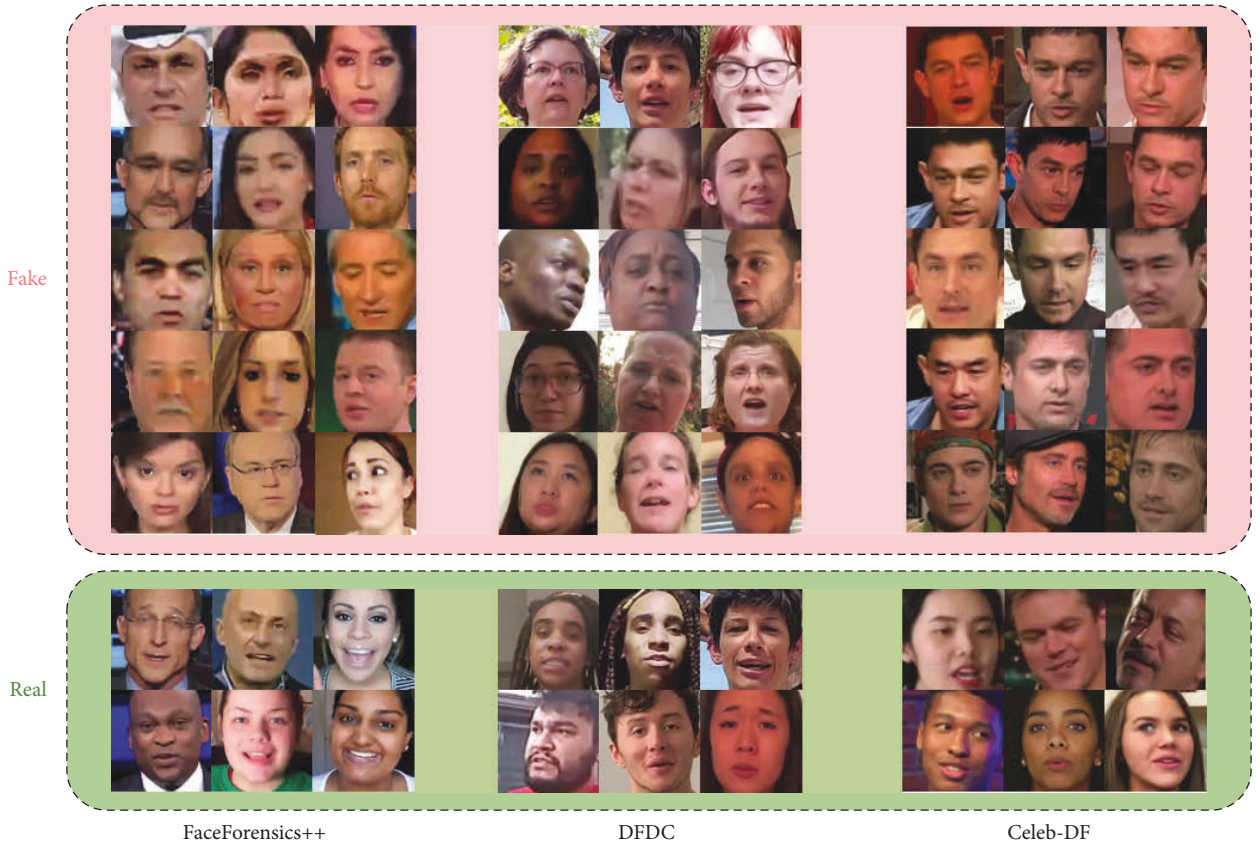


Figure 7: Examples of data from FaceForensics++, DFDC, and Celeb-DF datasets.

Table 1: Test results (%) of the FD^{2}Former and its variants on FF++(LQ), DFDC and Celeb-DF. The "image" indicates the model with cropped face images as input only. The "detail" indicates the model with facial geometry details as input only. The "swin" indicates the FD^{2}Former of the swin transformer backbone. The "meta" is the FD^{2}Former of the MetaFormer backbone. The "manifold" and the "hard" mean the manifold distillation and the hard-label distillation respectively. The metric on FF++(LQ), DFDC and Celeb-DF dataset is ACC.

Structure	FF++(LQ)	DFDC	Celeb-DF
Xception	80.32	85.60	61.25
Image(swin)	81.14	86.32	78.13
Detail(swin)	78.06	80.68	76.94
Img + detail(swin)	**83.23**	**87.97**	**83.51**
Img + detail + manifold(meta)	82.73	86.72	81.36
Img + detail + hard(meta)	81.67	86.03	79.91

The metric on FF++(LQ), DFDC and Celeb-DF dataset is ACC. Best results are shown in bold.

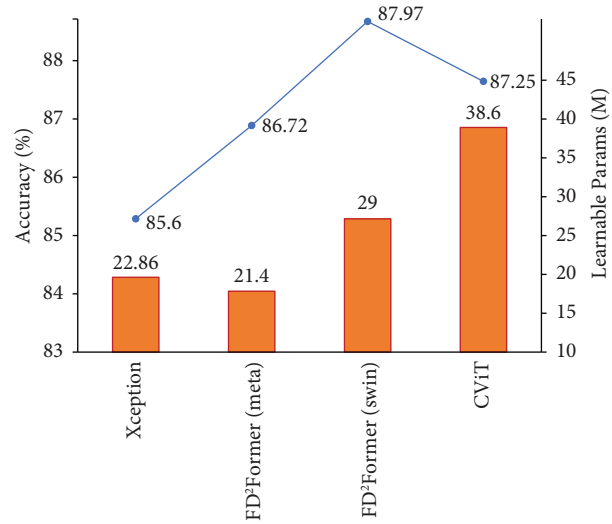


Figure 8: Comparison of the detection accuracy (%) and learnable params (M) with previous methods, Xception and CViT. The "FD^{2}Former(swin)" is the FD^{2}Former of the swin transformer backbone. The "FD^{2}Former(meta)" is the FD^{2}Former of the MetaFormer backbone. The experiment is conducted on DFDC dataset.

sampling correction, random sampler loss L_{rs}. As Table 5 shows, five experiments are launched to search for an appropriate value of k, and we finally assign 256 to k for a better performance. As indicated in Figure 11, a bigger k usually leads to a better performance.

5.4. Comparison with Other Methods. Some previous works have shown the generalization performance problem if confronted with the unseen manipulation methods. In this section, the proposed method is compared with previous ones, Xception [54] and Face X-ray [55], to explore the performance during detecting unseen manipulation methods or datasets.

Table 2: Results (%) of distillation for different manifold computing location. All trained with the image and the displacement map as inputs on low-quality FF++.

Teacher	Student	Teacher blocks extracted	Student blocks extracted	ACC
DF^{2}Former(swin)	DF^{2}Former(meta)	{11, 12}	{23, 24}	82.31
DF^{2}Former(swin)	DF^{2}Former(meta)	{1, 2}	{1, 2}	82.15
DF^{2}Former(swin)	DF^{2}Former(meta)	**{1, 2, 11, 12}**	**{1, 2, 23, 24}**	**82.73**
DF^{2}Former(swin)	DF^{2}Former(meta)	{1, 2, 3, 4, 11, 12}	{1, 2, 5, 6, 23, 24}	81.39

All trained with the image and the displacement map as inputs on low-quality FF++. Best results are shown in bold.

Table 3: Results (%) of distillation for a sequence of selected α. All trained with the image and the displacement map as inputs on low quality FF++. The metric is ACC.

Teacher	Student	α	Testing data (ACC)	
			FF++(LQ)	DFDC
DF^{2}Former(swin)	DF^{2}Former(meta)	0.5	81.76	86.17
DF^{2}Former(swin)	DF^{2}Former(meta)	**1**	**82.73**	**86.72**
DF^{2}Former(swin)	DF^{2}Former(meta)	1.5	82.42	86.47
DF^{2}Former(swin)	DF^{2}Former(meta)	2	82.19	85.74
DF^{2}Former(swin)	DF^{2}Former(meta)	2.5	81.71	85.86

The metric is ACC. Best results are shown in bold.

Table 4: Results (%) of distillation for a sequence of selected β. All trained with the image and the displacement map as inputs on low-quality FF++. The metric is ACC.

Teacher	Student	β	Testing data (ACC)	
			FF++(LQ)	DFDC
DF^{2}Former(swin)	DF^{2}Former(meta)	0	81.85	85.43
DF^{2}Former(swin)	DF^{2}Former(meta)	**0.2**	82.73	**86.72**
DF^{2}Former(swin)	DF^{2}Former(meta)	**0.4**	**82.86**	85.77
DF^{2}Former(swin)	DF^{2}Former(meta)	0.6	82.83	86.58
DF^{2}Former(swin)	DF^{2}Former(meta)	0.8	82.51	86.27
DF^{2}Former(swin)	DF^{2}Former(meta)	1	82.37	86.08

The metric is ACC. Best results are shown in bold.

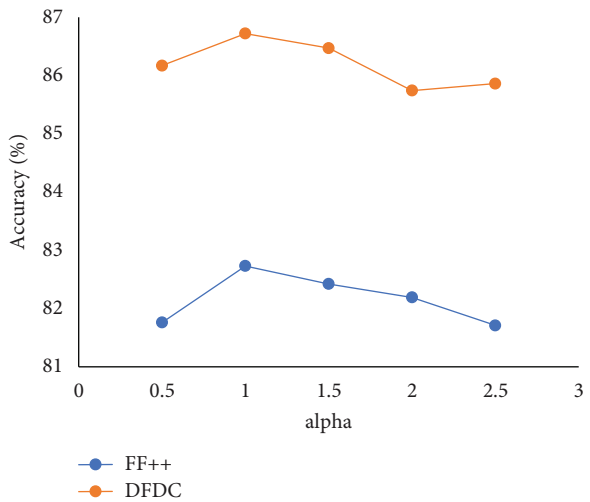


Figure 9: Accuracy (%) of the DF^{2}Former(meta) when the hyperparameter α varies. "(LQ)" refers to the heavily compressed FaceForensics++ dataset.

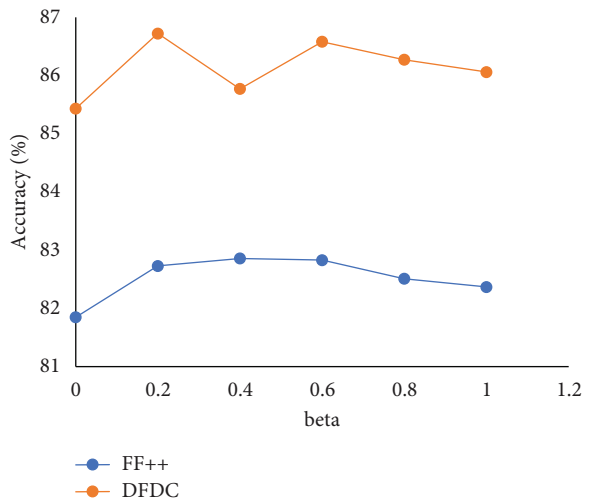


Figure 10: Accuracy (%) of the DF^{2}Former(meta) when the hyperparameter β varies. "(LQ)" refers to the heavily compressed FaceForensics++ dataset.

Following Luo et al. [56], the evaluations on different unseen manipulation methods are conducted on the FF++(HQ) [5] database, and we compare the performance with previous methods. The proposed method is trained on F2F, DF, FS, and NT separately and tested on the remaining methods, taking the AUC as the evaluation metric. Table 6

Table 5: Results (%) of distillation for a sequence of selected k. All trained with the image and the displacement map as inputs on low quality FF++. The metric is ACC.

Teacher	Student	k	Testing data (ACC)	
			FF++(LQ)	DFDC
DF^{2}Former(swin)	DF^{2}Former(meta)	0	81.85	85.43
DF^{2}Former(swin)	DF^{2}Former(meta)	64	82.41	85.71
DF^{2}Former(swin)	DF^{2}Former(meta)	128	82.27	86.54
DF^{2}Former(swin)	DF^{2}Former(meta)	192	82.68	86.37
DF^{2}Former(swin)	DF^{2}Former(meta)	**256**	**82.73**	**86.72**

The metric is ACC. Best results are shown in bold.

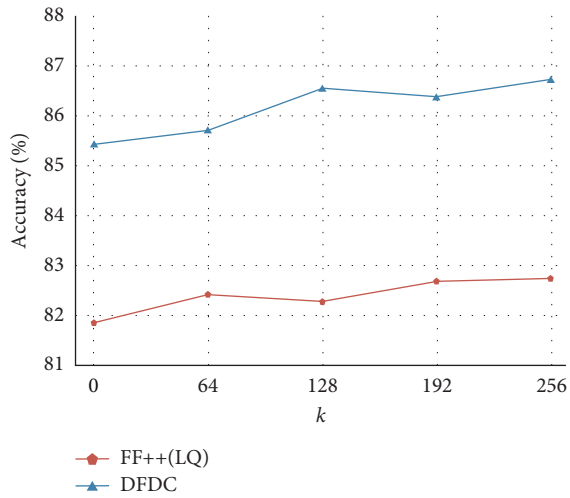


Figure 11: Accuracy (%) of the DF^{2}Former(meta) when the hyper-parameter k varies. "(LQ)" refers to the heavily compressed FaceForensics++ dataset.

Table 6: Generalization evaluation (%) with previous methods on HQ (c23) FF++. AUC is used as metric of performance on the unseen manipulation technique. The highlighted are best results.

Training data	Model	Testing data (AUC)			
		DF	F2F	FS	NT
DF	Xception [54]	**99.3**	**73.6**	49.0	73.6
	Face X-ray [55]	98.7	63.3	**60.0**	69.8
	FD^{2}Former	98.94	69.17	59.58	**77.39**
F2F	Xception [54]	80.3	**99.4**	76.2	69.6
	Face X-ray [55]	63.0	98.4	**93.8**	**94.5**
	FD^{2}Former	**81.78**	98.12	77.31	89.52
FS	Xception [54]	66.4	88.8	**99.4**	71.3
	Face X-ray [55]	45.8	96.1	98.1	**95.7**
	FD^{2}Former	**70.21**	**97.83**	99.27	93.12
NT	Xception [54]	79.9	81.3	73.1	**99.1**
	Face X-ray [55]	70.5	**91.7**	91.0	92.5
	FD^{2}Former	**80.43**	90.82	**92.53**	94.62

AUC is used as metric of performance on the unseen manipulation technique. Best results are shown in bold.

presents the experimental results. Compared to the classic method XceptionNet, the proposed FD^{2}Former achieves a significant improvement most of time, since the former overly relies on the texture patterns in RGB space without thinking unseen details. Face X-ray [55], achieves a better generalization performance benefitting from the blending evidence detection. The FD^{2}Former leverages both textures and highlighted clues extracted from the facial geometry details, which probably why it generalizes better from one method to another.

Following [57] Khodabakhsh et al. [58], the generalization performance is analyzed quantitatively on unseen data and compared with other methods, including the classic Xception [54], the ensemble of EfficientNet's variants [59], and the Face X-ray [55]. The models are trained on FF++ (HQ) [5] and evaluated on Celeb-DF [6], DFDC [7],

Table 7: Generalization evaluation (%) on the unseen dataset, Celeb-DF, and DFDC. The highlighted are best results.

Model	Training	Testing AUC	
		Celeb-DF	DFDC
EfficientNetB4Ensemble [59]	FF++	55.8	63.0
Xception [54]		59.4	67.9
Face X-ray [55]		74.2	70.0
FD^{2}Former		**79.2**	**73.6**

Generalization evaluation (%) on the unseen dataset, Celeb-DF, and DFDC. Best results are shown in bold.

Figure 12: Obtained result of some tested face images for DeepFake detection with the estimated category and the probabilities of each one. (a) FaceForensics++. (b) DFDC. (c) Celeb-DF.

respectively. Such a generalization problem may confront more challenges than the experiments within FF++. We can see from Table 7 that our method achieves apparent improvements over EfficientNetB4 Ensemble [59] and Xception [54], indicating that our model learns more robust representations than the previous methods. Figure 12 displays the obtained result of some tested face images.

6. Conclusion

In this paper, we propose a novel approach for the detection of face manipulation by reconstructing 3D face. We find the facial geometry details through the 3D coarse reconstruction and detail reconstruction. We use the displacement map to amplify the complex artifact patterns. The clues in the geometry details and the cropped face images are fed into FD^{2}Former to classify whether the input face is real or not. Meanwhile, for the capabilities of many edge devices such as smartphones and IoTs, we use MetaFormer architecture to build a light neural network and introduce a manifold learning strategy to improve the performance of our method. The comprehensive experiments on FaceForencis++, Celeb-DF, and DFDC exhibit the effectiveness and generalization of our FD^{2}Former. On the whole, our work presents a novel direction to detect the face manipulation clues through 3D face reconstruction.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been supported by the Key R&D projects in Hubei Province under Grant No. 2022BAA041 and No. 2021BCA124 and the Open Foundation of Engineering Research Center of Cyberspace under Grant No. KJAQ202112002.

References

[1] S 3805-Malicious Deep, "Fake prohibition act of," *Ben Sasse*, vol. 1, 2018.

[2] P. Korshunov and S. Marcel, "Vulnerability assessment and detection of deepfake videos[C]," in *Proceedings of the 2019 International Conference on Biometrics (ICB)*, pp. 1–6, IEEE, Piscataway, NJ, June 2019.

[3] "Fake videos could be the next big problem in the 2020 elections," *Grace Shao*, Oct 2019, https://www.cnbc.com/2019/10/15/deepfakes-could-be-problem-for-the-2020-election.html.

[4] Deepfakes and the new disinformation war, "The coming age of post-truth geopolitics," *Robert Chesney and Danielle Citron*, Jan/Feb 2019, https://www.foreignaffairs.com/articles/world/2018-12-11/deepfak%20es-and-new-disinformation-war.

[5] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: learning to detect manipulated facial images[C]," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1–11, Nashville, TN, USA, June 2019.

[6] Y. Li, X. Yang, P. Sun, Q. Honggang, and L. Siwei, "Celeb-DF (v2): a new dataset for DeepFake Forensics[J]," 2019, http://arXiv.org/abs/1909.12962.

[7] B. Dolhansky, J. Bitton, B. Pflaum et al., "The deepfake detection challenge (dfdc) dataset[J]," 2020, http://arXiv.org/abs/2006.07397.

[8] C. Feichtenhofer, H. Fan, J. Malik, and H. Kaiming, "Slowfast networks for video recognition[C]," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6202–6211, Nashville, TN, USA, July 2019.

[9] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-attentional deepfake detection[C]," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2185–2194, Nashville, TN, USA, June 2021.

[10] T. Vetter and V. Blanz, "Estimating coloured 3D face models from single images: an example based approach[C]," in *Proceedings of the European conference on computer vision*, pp. 499–513, Berlin, Heidelberg, 1998.

[11] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 1–17, 2017.

[12] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need[J]," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[13] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R. Girshick, "Early convolutions help transformers see better [J]," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30392–30400, 2021.

[14] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the International Conference on Machine Learning*, pp. 10347–10357, PMLR, Las Vegas, July 2021.

[15] W. Wang, E. Xie, X. Li et al., "Pyramid vision transformer: a versatile backbone for dense prediction without convolutions [C]," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 568–578, Nashville, TN, USA, June 2021.

[16] W. Yu, M. Luo, P. Zhou, W. Xinchao, F. Jiashi, and Y. Shuicheng, "Metaformer is actually what you need for vision," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10819–10829, Nashville, TN, USA, June 2022.

[17] D. Jia, K. Han, Y. Wang et al., "Efficient vision transformers via fine-grained manifold distillation[J]," 2021, http://arXiv.org/abs/2107.01378.

[18] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: learning lip sync from audio[J]," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–13, 2017.

[19] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt, "Real-time expression transfer for facial reenactment," *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 1–14, 2015.

[20] J. Thies, M. Zollhofer, M. Stamminger, T. Christian, and N. Matthias, "Face2face: real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2387–2395, Las Vegas, June 2016.

[21] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: a Survey of face manipulation and fake detection," *Information Fusion*, vol. 64, pp. 131–148, 2020.

[22] H. Farid, "Image forgery detection[J]," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16–25, 2009.

[23] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, "Vision of the unseen: current trends and challenges in digital image and video forensics[J]," *ACM Computing Surveys*, vol. 43, no. 4, pp. 1–42, 2011.

[24] L. Nataraj, T. M. Mohammed, B. S. Manjunath, and B. Tondi, "Detecting GAN generated fake images using co-occurrence matrices[J]," *Electronic Imaging*, vol. 2019, no. 5, pp. 532–541, 2019.

[25] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-stream neural networks for tampered face detection[C]," in *Proceedings of the 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 1831–1839, IEEE, Honolulu, HI, USA, July 2017.

[26] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses[C]," in *Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8261–8265, IEEE, Brighton, UK, May 2019.

[27] T. J. De Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A de Rezende Rocha, "Exposing digital image forgeries by illumination color classification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1182–1194, 2013.

[28] X. Zhu, H. Wang, H. Fei, Z. Lei, and S. Li, "Face forgery detection by 3d ecomposition[," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2929–2939, Nashville, TN, USA, July 2021.

[29] T. Brown, B. Mann, N. Ryder et al., "Language models are few-shot learners[J]," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[30] J. Devlin, M. W. Chang, K. Lee, and T. Kristina, "Bert: pre-training of deep bidirectional transformers for language understanding[J]," 2018, http://arXiv.org/abs/1810.04805.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is worth 16x16 words: transformers for image recognition at scale[J]," 2020, http://arXiv.org/abs/2010.11929.

[32] N. Carion, F. Massa, G. Synnaeve, U. Nicolas, K. Alexander, and Z. Sergey, "End-to-end object detection with transformers[C]," in *Proceedings of the European conference on computer vision*, pp. 213–229, Springer, Cham, June 2020.

[33] J. Lu, D. Batra, D. Parikh, and L. Stefan, "Vilbert: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks[J]," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[34] M. Chen, A. Radford, R. Child, and H. Jun, "Generative pretraining from pixels," in *Proceedings of the International conference on machine learning*, pp. 1691–1703, PMLR, venue City, September 2020.

[35] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust single-view geometry and motion reconstruction," *ACM Transactions on Graphics*, vol. 28, no. 5, pp. 1–10, 2009.

[36] K. He, X. Zhang, S. Ren, and S. Jian, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, San Juan, PR, USA, June 2016.

[37] P. Paysan, R. Knothe, B. Amberg, Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Proceedings of the 2009 sixth IEEE international conference on advanced video and signal based surveillance*, pp. 296–301, IEEE, Genova, Italy, September 2009.

[38] R. Ramamoorthi and P. Hanrahan, "An efficient representation for irradiance environment maps," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 497–500, New York, NY, United States, August 2001.

[39] Y. Feng, H. Feng, M. J. Black, and T. Bolkart, "Learning an animatable detailed 3D face model from in-the-wild images," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–13, 2021.

[40] Q. Cao, L. Shen, W. Xie, and Z. Andrew, "Vggface2: a dataset for recognising faces across pose and age," in *Proceedings of the 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pp. 67–74, IEEE, Xi'an, China, May 2018.

[41] M. Wang, W. Deng, J. Hu, and H. Yaohai, "Racial faces in the wild: reducing racial bias by information maximization adaptation network," in *Proceedings of the ieee/cvf international conference on computer vision*, pp. 692–702, Montreal, BC, Canada, September 2019.

[42] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: deep speaker recognition," 2018, https://arxiv.org/abs/1806.05622.

[43] g. Adam, *The world's simplest facial recognition api for Python and the command line*, China, 2020, https://github.com/ageitgey/face_recognition.

[44] L. Yuan, Y. Chen, T. Wang et al., "Tokens-to-token vit: training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 558–567, Montreal, BC, Canad, July 2021.

[45] D. Zhou, Y. Shi, B. Kang et al., "Refiner: refining self-attention for vision transformers[J]," 2021, https://arxiv.org/abs/2106.03714.

[46] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov et al., "Mlp-mixer: an all-mlp architecture for vision[J]," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24261–24272, 2021.

[47] H. Touvron, P. Bojanowski, M. Caron et al., "Resmlp: feedforward networks for image classification with data-efficient training[J]," 2021, http://arXiv.org/abs/2105.03404.

[48] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2016, https://arxiv.org/abs/1606.08415.

[49] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines[C]," *Icml*, 2010.

[50] Z. Liu, Y. Lin, Y. Cao et al., "Swin transformer: hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, Tamara Berg, October 2021.

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

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

[53] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: image synthesis using neural textures[J]," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–12, 2019.

[54] F. Chollet, "Xception: deep learning with depthwise separable convolutions[C]," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, Honolulu, HI, USA, July 2017.

[55] L. Li, J. Bao, T. Zhang et al., "Face x-ray for more general face forgery detection[C]," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5001–5010, Nashville, TN, USA, July 2020.

[56] Y. Luo, Y. Zhang, J. Yan, and L. Wei, "Generalizing face forgery detection with high-frequency features," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16317–16326, Nashville, TN, USA, June 2021.

[57] Y. Qian, G. Yin, L. Sheng, C. Zixuan, and S. Jing, "Thinking in frequency: face forgery detection by mining frequency-aware clues[J]," 2020, https://arxiv.org/abs/2007.09355.

[58] A. Khodabakhsh, R. Ramachandra, K. Raja, P. Wasnik, and C. Busch, "Fake face detection methods: can they be generalized?[C]," in *Proceedings of the 2018 international conference of the biometrics special interest group (BIOSIG)*, pp. 1–6, IEEE, Darmstadt, Germany, September 2018.

[59] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video face manipulation detection through ensemble of cnns[C," in *Proceedings of the 2020 25th international conference on pattern recognition (ICPR)*, pp. 5012–5019, IEEE, Milan, Italy, May 2021.