



OPEN Localization and detection of deepfake videos based on self-blending method

Junfeng Xu[✉], Xintao Liu, Weiguo Lin[✉], Wenqing Shang[✉] & Yuefeng Wang

Deepfake technology, which encompasses various video manipulation techniques implemented through deep learning algorithms—such as face swapping and expression alteration—has advanced to generate fake videos that are increasingly difficult for human observers to detect, posing significant threats to societal security. Existing methods for detecting deepfake videos aim to identify such manipulated content to effectively prevent the spread of misinformation. However, these methods often suffer from limited generalization capabilities, exhibiting poor performance when detecting fake videos outside of their training datasets. Moreover, research on the precise localization of manipulated regions within deepfake videos is limited, primarily due to the absence of datasets with fine-grained annotations that specify which regions have been manipulated. To address these challenges, this paper proposes a novel spatial-based training method that does not require fake samples to detect spatial manipulations in deepfake videos. By employing a technique that combines multi-part local displacement deformation and fusion, we generate more diverse deepfake feature data, enhancing the detection accuracy of specific manipulation methods while producing mixed-region labels to guide manipulation localization. We utilize the Swin-UNET model for manipulation localization detection, incorporating classification loss functions, local difference loss functions, and manipulation localization loss functions to effectively improve the precision of localization and detection. Experimental results demonstrate that the proposed spatial-based training method without fake samples effectively simulates the features present in real datasets. Our method achieves satisfactory detection accuracy on datasets such as FF++, Celeb-DF, and DFDC, while accurately localizing the manipulated regions. These findings indicate the effectiveness of the proposed self-blending method and model in deepfake video detection and manipulation localization.

Forgery primarily relies on deepfake faceswapping technology, which employs deep learning methods to manipulate images or videos. With the continuous advancement of deep learning, deepfake technology is becoming increasingly mature, producing fake images and videos that are more realistic, thus exerting a more profound impact on societal security. For instance, techniques such as Progressive Growing of GANs (PGGAN)¹, Style-based GAN (StyleGAN)², Unified Generative Adversarial Networks (StarGAN)³, High-fidelity GAN (HifiGAN)⁴, Face Swapping GAN (Fsgan)⁵, and FaceShifter⁶ are used for synthesizing full-face images, editing attributes, swapping identities, or using Face2Face⁷ for exchanging facial expressions. Despite their innovative potential, these technologies, when abused, pose severe hazards. They enable the creation of highly realistic fake videos, infringing upon personal privacy and reputation, and undermining social trust. Moreover, they present significant challenges to law enforcement and regulation, making it difficult to identify and punish infringing acts. Additionally, deepfake videos may have far-reaching economic, social, and political impacts, including misleading consumers and compromising electoral integrity. Fortunately, despite the sophistication of these meticulously forged images and videos, they can still be detected using deep learning methods.

To curb the widespread dissemination of deepfake videos on the internet, there is an urgent need for deepfake detection technology to discern forged videos. In pursuit of this goal, Facebook collaborated with prominent companies and universities such as Microsoft and MIT to launch the Deepfake Detection Challenge (DFDC) on the Kaggle competition platform⁸. This competition attracted participation from over 2200 teams, significantly promoting research and development in deepfake detection technology.

However, despite the significant achievements in fostering technological innovation through the DFDC competition, the reliability of deepfake detection technology has not yet reached an ideal level. When using well-trained models to detect datasets with unknown manipulation methods, the detection results are poor,

Communication University of China, School of Computer & Cyber Sciences, Beijing 100024, China. ✉email: junfeng@cuc.edu.cn; linwei@cuc.edu.cn; shangwenqian@cuc.edu.cn

indicating the need for further improvement in the models' generalization. The application of deepfake detection technology in the real world still faces certain challenges. Facebook officially stated after the conclusion of the DFDC competition: "The problem of detecting facial deepfake remains unsolved." Therefore, research on deepfake video detection technology is still needed to enhance its detection accuracy and generalization.

Numerous studies have been conducted on the detection of Deepfake face-swapping videos, typically based on deep learning methods. These studies can be classified into frame-level detection and video-level detection. Frame-level detection focuses on spatially extracting forged features without considering inter-frame relationships. For example, exploring spatial inconsistency achieve high accuracy in detecting manipulations within the dataset^{9–12}. Additionally, some methods that combine frequency domain information fusing high-frequency manipulation features to improve model accuracy^{13–15}. Video-level detection incorporates inter-frame information to primarily explore temporal inconsistencies in videos. Since video manipulations are typically performed frame by frame and then synthesized into a video, forged videos exhibit temporal discontinuities compared to genuine ones. Utilizing temporal or spatiotemporal combination for detection improves model performance and generalization^{20–25}. The Critical Forgery Mining (CFM) framework⁴⁴ pioneered this transition by moving beyond prior knowledge limitations, while Vision Transformer (ViT)-based methods have emerged as powerful solutions, incorporating innovations such as Low-Rank Adaptation⁴⁶ and Mixture-of-Experts modules⁴⁸ to enhance detection performance while maintaining computational efficiency. These developments address critical challenges in generalization and robustness through pixel-inconsistency modeling⁴⁷ and parameter-efficient adaptation methods⁴⁹, while the Forgery-aware Adaptive Vision Transformer (FA-ViT)⁵⁰ further improves cross-database performance. Together, these advancements represent a comprehensive response to the growing security threats posed by deepfakes⁴⁵, establishing more reliable and efficient detection systems for real-world applications.

However, these methods have relatively low accuracy in detecting unseen manipulation methods, indicating insufficient generalization. To enhance the models' generalization, it is necessary to capture the commonalities of manipulations. Face X-ray¹⁶ provides a good idea, focusing on the commonality of Deepfake face-swapping forgery. By proposing an automatic mixer based on traditional synthesis methods and reproducing manipulation vulnerabilities for model learning, the model's generalization is significantly improved. Subsequent methods such as PCL-I2G¹⁸ and SBI¹⁹ also reproduce forgery vulnerabilities through synthesis, achieving state-of-the-art performance and generalization. These training methods, known as learning without forged samples, only utilize real samples from the dataset. These methods achieve high detection accuracy on face-swapping forgery samples and exhibit good generalization.

Meanwhile, since these detection methods are primarily designed for training samples of face-swapping forgery, their accuracy in detecting videos manipulated through expression manipulation or texture editing methods is limited. Furthermore, there has been insufficient research on traceability issues, namely manipulation localization detection. Therefore, this paper will employ a training method without forged samples to realize manipulation localization detection and improve the model's accuracy in detecting various forgery methods, enhancing the model's generalization in detection.

The main contributions of this paper are as follows:

1. A Multi-Part Local Displacement Deformation Self-Blending Method: By employing a training method without forged samples, the model can learn the forged features of deepfake methods such as expression manipulation or texture editing, thereby improving the detection accuracy of these forged samples and enhancing the model's generalization.
2. Adoption of Multi-Task Learning: Designing classification loss functions, local difference loss functions, and manipulation localization loss functions enables the model to focus not only on global features but also on local difference features. This approach not only achieves manipulation localization and detection but also enhances the model's detection accuracy.
3. Experimental Validation: Demonstrating that the proposed self-blending training method effectively simulates the features in real datasets. Detection conducted on datasets such as FF++, Celeb-DF, and DFDC achieves satisfactory accuracy, while accurately localizing the manipulated regions. This indicates the effectiveness of the proposed self-blending method and model in deepfake video detection and manipulation localization.

Relate work

A. Deepfake Detection

Currently, research on detecting Deepfake face-swapping videos predominantly employs methods based on deep learning. These approaches model frame images or video features using deep learning techniques and utilize a binary classification strategy for detection. Specifically, detection methods are categorized into frame-level and video-level approaches. Frame-level detection focuses on spatially extracting forged features without accounting for inter-frame relationships, whereas video-level detection incorporates temporal information between frames, primarily investigating inconsistencies in the temporal sequence of videos.

In 2019, the release of the Faceforensics++ (FF++) dataset greatly propelled the development of Deepfake detection technology. It includes 1000 original videos, 4000 forged videos created using four deepfake methods, and frame-level detection using various convolutional neural network models. Among them, Xception achieves the best detection accuracy within the dataset. However, the detection generalization of models using global feature extraction and binary classification training is poor due to their failure to fully exploit spatially forged features. To enhance the network's focus on various facial regions, Zhao et al¹⁹ proposed a fine-grained detection method, introducing an attention map generation module to guide the generation of attention maps for shallow texture feature generation. This approach directs the network's attention to multiple regions of the face, aiding

in capturing subtle manipulation traces. Wang and Deng¹⁰ removed attention positions calculated during network feedback training and then trained the model to focus on other positions to improve the detection of local manipulation effects. Dong et al¹¹. proposed an ID-unaware Deepfake Detection Model, which guides the network's focus to local areas through the introduction of local loss, thereby enhancing generalization. Additionally, some methods combine frequency domain information for detection. Qian et al¹². use frequency domain transformation and local frequency statistics, employing CNNs for feature extraction and fusion to enhance detection accuracy, particularly for low-resolution image discrimination. Li et al¹³. similarly use frequency domain transformation, employing an adaptive feature generation model to extract features from frequency domain images and train with special loss functions to obtain more discriminative features. Tian et al¹⁴. combine local and global features, spatial and frequency domain features, achieving good performance in both intra-dataset and cross-dataset tests.

A particularly noteworthy training approach is the training without forged samples method, which only uses real data from the dataset during training. Li et al¹⁵. focused on synthetic vulnerabilities, proposing an automatic mixer model that generates mixed data by mixing real samples from the dataset, enhancing the model's learning of mixed features. Zhao et al¹⁶. further enhanced this by applying data augmentation to mixed images, such as compression, adding Gaussian noise, changing brightness contrast, and randomly selecting two mixing methods to increase sample diversity. Chen et al¹⁷. introduced the selection of mixing types and mixing ratios to increase sample diversity, also incorporating multi-task learning to improve model detection generalization. Shiohara and Yamasaki¹⁸ used self-blending, capturing more manipulation vulnerabilities such as facial mismatches, obvious mixing boundaries, color inconsistencies, and frequency inconsistencies. These methods share the common feature of using simple methods to generate datasets with vulnerabilities in manipulation methods for model learning. This strategy helps the model to comprehensively understand various manipulation methods, enhancing generalization and detection capabilities.

In video-level detection, temporal information between frames, i.e., temporal inconsistencies, is explored. Models that extract temporal features, such as RNNs, LSTMs, Transformers, and 3DCNNs, can be used. Masi et al¹⁹. proposed a dual-branch recursive network detection framework that extracts features from frames in the RGB and frequency domains, fusing features from both domains. Finally, the sequence is passed through bidirectional LSTMs to obtain video-level features with temporal information. Gu et al²⁰. proposed a spatiotemporal inconsistency detection framework, with two branches exploring spatial and temporal inconsistencies, achieving good results in intra-dataset or cross-dataset tests. Sun et al²¹. input a feature encoding sequence formed by a series of facial keypoint coordinates into a dual-stream RNN network to extract temporal information. Zheng et al²². proposed a Fully Temporal Convolutional Network (FTCN), a detection method that fully explores temporal information, combining 3DCNNs with Transformers to extract long and short-distance temporal information, achieving good generalization. Gu et al²³. proposed a hierarchical detection method, dividing it into local and global levels. The local level consists of consecutive frames called segments, while the global level integrates several segments to simultaneously focus on short-term and long-term inconsistencies. Guan et al²⁴. combined Vision Transformer modules to extract local temporal information. Firstly, using attention mechanisms to obtain temporal information about the same local regions between frames, then applying attention mechanisms to each local region's temporal information to obtain video-level features, significantly improving generalization and accuracy. These models employ different strategies for mining temporal information, enabling them to better handle Deepfake videos, improving detection performance and generalization.

In addition to spatiotemporal integration methods, there is a category called multimodal detection methods, which combine various information such as images, sounds, and text to detect videos. Haliassos et al²⁵. detect fake videos by discriminating whether lip shapes and speech match. Cheng et al²⁶. detect fake videos by discriminating whether sounds and faces match. Yang et al²⁷. proposed a method to detect audio-visual inconsistency by exploiting inconsistencies between speech and visuals, and fabricated their own dataset DefakeAVMiT, where each modality is manipulated through various synthesis methods. Feng et al²⁸. detect videos by abnormal visual and auditory cues, proposing a method to create audiovisual inconsistencies through time delay. By synthesizing datasets with flaws that may cause audiovisual anomalies for model training, the model learns these flaws. Their datasets FakeAVCeleb and KoDF achieve high accuracy. Zhang et al²⁹. proposed the joint audio-visual deepfake detection model AVA-CL, which fully utilizes the inherent correlation between facial and audio clues to conduct deepfake detection simultaneously in the audio and visual domains, outperforming current leading methods in robustness and generalization. Although audiovisual combined detection methods can achieve high accuracy and generalization, some methods have to fabricate their own datasets due to the lack of large-scale training data suitable for training.

B. Binary Mask Supervision

Among the methods mentioned above, only image-level or video-level labels are available, indicating only the authenticity of images or videos. Such labels cannot directly guide the model to learn localized forged features accurately. Therefore, more fine-grained mosaic labels, which represent manipulated areas with '1' and unaltered areas with '0', can more precisely guide the model to learn local forged features. Mosaic labels play a crucial role in training manipulation localization models.

Bappy et al³⁰. designed a hybrid CNN-LSTM model to capture boundary difference features, achieving manipulation localization functionality. Chen et al³¹. proposed an end-to-end detection and localization framework, introducing the DLFMNet model, which integrates RGB image information and noise image information to achieve both detection and localization. Kong et al³². utilized mosaic labels and noise images to extract high-level semantic information clues and low-level clues effectively from images, achieving high detection accuracy and localization functionality. Chao et al³³. used a dual-stream network to extract features from images and noise images and used mosaics as labels to guide fine-grained similarity comparison, improving cross-dataset testing accuracy. Zhao et al³⁴. utilized the concept of domain generalization and designed a hybrid

domain meta-learning network to achieve manipulation detection and localization functionality, demonstrating good performance on several benchmark datasets.

In contrast to the aforementioned methods, this paper employs a self-blending method to generate more diverse data and produces more accurate mosaics as fine-grained labels for guidance. Additionally, it captures multi-scale local difference features to perform manipulation detection and localization on images.

Method

This section begins by introducing the proposed multi-part local deformation self-blending method, which is utilized to generate negative samples, i.e., forged samples, in the training data. Subsequently, it outlines the manipulation localization detection model employed in this paper along with the training methodology.

A. Multi-part Local Deformation Self-blending Method

In the previous self-blending methods, the operation of blending two faces was utilized to mimic the synthesis of deepfake faces. This approach not only enables the model to learn the local differences in the synthesis of two faces but also provides richer data for model learning. Building upon this, various data augmentation techniques are applied to the images, including local deformation, frequency variation, color transformation, displacement, and scaling. Additionally, operations for synthesizing different parts of the face are introduced. Apart from enhancing the diversity of training samples, this approach allows for the simulation of forged features specific to certain manipulation methods, thereby strengthening the model's learning of these features. The steps involved in the multi-part local deformation self-blending are illustrated in Fig. 1.

Multiple-part local displacement and deformation are introduced by a hybrid input, represented by a genuine face image I . This image is duplicated to yield the source face I_s and the target face I_t . Random data augmentation techniques are applied to both faces, including variations in the RGB color space, sharpening, and downsampling. Concurrently, corresponding mosaics are generated for selected facial regions such as eyes, mouth, nose, and overall face based on the key points of the genuine face.

The self-blended face is generated through a sequential masking process, where different facial regions (eyes, mouth, nose, and overall face) are processed individually with unique masks created from facial landmarks. During the training process, the mosaic generation incorporates extensive random variations: the area coverage of each facial region varies dynamically (eyes: 80 – 120%, mouth: 90 – 130%, nose: 85 – 115%, overall face: 90 – 110% of their original sizes), while mosaic parameters such as cell size (4–16 pixels), grid orientation (± 5 degrees), and opacity (85 – 100%) are randomly determined for each iteration. This comprehensive randomization in both the masking sequence and mosaic generation ensures that each training sample presents unique characteristics, creating a diverse dataset that enhances the model's ability to detect various types of facial manipulations and prevents overfitting through exposure to a wide range of blending scenarios.

Subsequently, localized translation and deformation are applied to the target face to alter its expression or facial contour, simulating potential artifacts of deepfake generation through displacement or scaling operations, resulting in I'_t . Finally, the final blended face O is obtained by preserving the external features of the source face and internal features of the target face based on the mosaics.

The inclusion of localized translation and deformation aims to simulate expression manipulation or texture editing for forged features, thereby enhancing the model's discriminative capability in these aspects. The localized translation and deformation algorithm can infer the displacement of a point within a specific region and the corresponding displacement positions of other points. The color of the displaced point is filled using bilinear interpolation. This enables slight deformations in small areas, such as subtle changes in mouth corners,

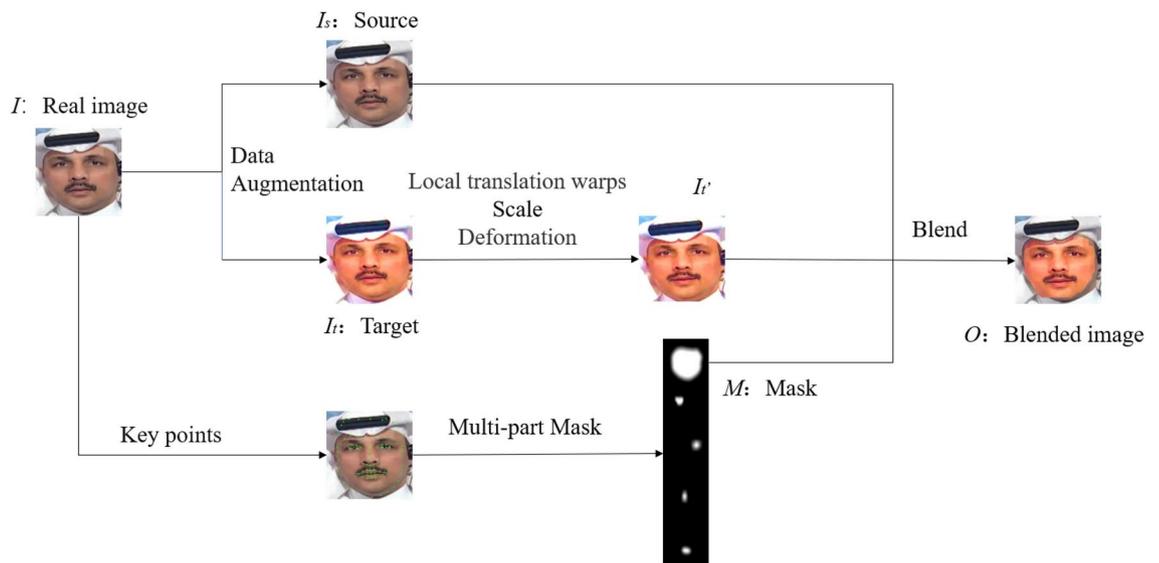


Fig. 1. Multi-part local deformation self-blending method. (The face images shown in this figure are sourced from the FaceForensics++ (FF++) dataset³⁶).

face slimming, eyebrow deformations, drooping or lifting of eye corners, etc. The formula is depicted as follows in Equation (1).

$$\alpha = \left(\frac{r_{\max}^2 - |\vec{x} - \vec{c}|^2}{(r_{\max}^2 - |\vec{x} - \vec{c}|^2) + |\vec{m} - \vec{c}|^2} \right)^2 \tag{1}$$

$$\vec{u} = \vec{x} - \alpha(\vec{m} - \vec{c}) \tag{2}$$

Taking the example of changing the expression to a smile, the first step involves altering the position of the mouth corners. Using the mouth corner as the center, a circle is drawn with a specified radius, forming a circular region as depicted in Fig 2. To lift the corners of the mouth, points near the mouth corners need to be vertically displaced. Assuming the center point C , a certain distance upward leads to point M . Now, to determine any point X within the circular region, it is necessary to ascertain from which position point U is displaced. Equation (2) computes the displacement vector $\vec{m} - \vec{c}$ to obtain the displacement distance of point C , while Equation (1) establishes the displacement ratio for other points. The numerator represents the distance of point X from the circumference, while the denominator is the sum of the distance from X to the circumference and the distance from the circumference to point M . This implies that points closer to the circumference have a smaller proportion and hence a smaller displacement, while points farther away have a larger proportion and a greater displacement. Squaring the ratio ensures that the displacement of points on the circumference from the center is non-linear, thereby ensuring greater displacement for points closer to the center. By iterating through all points within the region, the original position of each point can be determined.

Now that the original position of a point X before displacement has been determined. It is necessary to assign the color information of the pre-displacement point U to point X . Since point X may not align precisely with pixel boundaries, bilinear interpolation is employed to approximate its color information using the color values of the surrounding four points. The calculation of bilinear interpolation is illustrated in Fig. 3.

By utilizing the U -coordinate, we can derive the coordinates of its adjacent points: $a(x_0, y_1), b(x_1, y_1), c(x_0, y_0), d(x_1, y_0)$. As depicted in the figure, this region is partitioned into four areas: A, B, C, and D. Through the coordinates, we can calculate the areas of these four regions separately. Taking A as an example, the formula for the area of A is illustrated in equation (3). Essentially, this equation denotes the weight of color at point d attributed to point U . In other words, the closer U is to a certain pixel point, the greater the weight it holds, and the closer the color approximation. Equation (4) computes the extent to which point U contributes to the color of point d , where the function $f()$ represents the numerical value of the color domain at a certain point. Finally, by synthesizing the extent to which point U influences the color of each point, we obtain the color of point U , denoted as $f(U)$, as shown in equation (5).

$$\omega_d = S_A = (x - x_0)(y_1 - y) \tag{3}$$

$$f(U_d) = \omega_d \cdot f(d) \tag{4}$$

$$f(U) = \omega_a \cdot f(a) + \omega_b \cdot f(b) + \omega_c \cdot f(c) + \omega_d \cdot f(d) \tag{5}$$

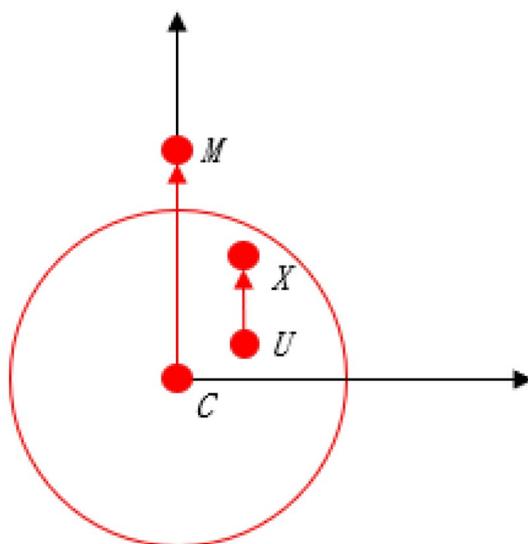


Fig. 2. Local displacement and deformation process.

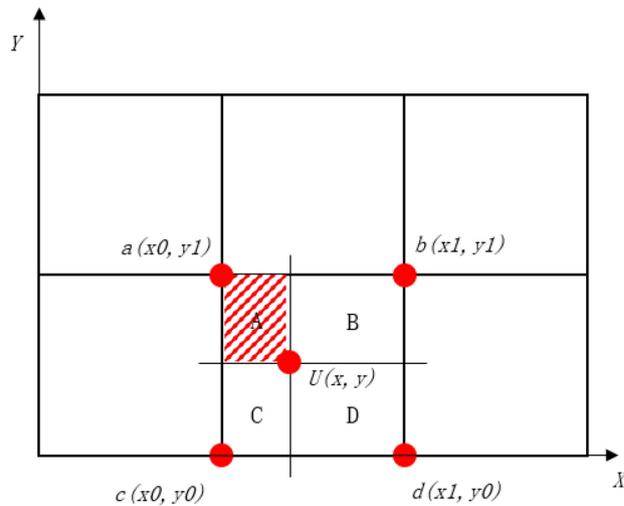


Fig. 3. Bilinear interpolation.

Finally, through the generated mosaic, the source face I_s and the target face I_t' are synthesized. The synthesis formula is represented by equation (6), where \odot denotes element-wise matrix multiplication. This preserves the black regions of the mosaic from the source image while retaining the white regions from the target image.

$$O = I_s \odot M + I_t' \odot (1 - M) \quad (6)$$

By employing self-blending techniques, a multitude of data is synthesized, imbued with characteristics akin to those of fake faces generated through deepfake methods. These characteristics may include certain artifacts and spatial inconsistencies. Additionally, the method employs local displacement deformations to simulate facial expression manipulation, enriching the learned features of the model and enhancing its generalizability. Subsequently, mosaics of different regions are generated to enrich the synthesis region, diversify the dataset, and enable the model to perceive both global and local features.

Next, it will introduce how to train practical models for detection, emphasizing techniques to facilitate the learning of spatial inconsistencies and utilizing self-blending methods for tampering localization.

B. Detection Model

The Swin-Unet model was first proposed in³⁵ and was utilized for medical image classification. It is a Unet model constructed entirely based on the Swin-Transformer architecture. Its front-end serves as the encoder, transforming a medical X-ray image into features, while the back-end functions as the decoder, converting the features back into an image.

This image achieves the classification of various organ types on the X-ray image, represented with different colors. Building upon this foundation, this paper improves the model to enable tampering localization. The model framework is depicted in Fig. 4.

The lower part comprises the encoder section, which shares the same structure as the Swin-transformer model. Each stage consists of layers (2, 2, 6, 2), equivalent to the size of the Swin-t model. Given an input of dimensions (H, W, C) , it passes through four stages of Swin-transformer modules to obtain feature maps of dimensions $\frac{H}{32} \times \frac{W}{32} \times 8C$, as depicted in equation (7).

$$F = \text{Encoder}(I) \quad (7)$$

The feature maps serve two additional purposes besides being fed into the decoder to generate predicted mosaics. Initially, the feature maps are used to generate binary predictions of authenticity. The feature maps F undergo adaptive average pooling to yield a feature vector $F_C(1, 8C)$. Subsequently, a linear layer reduces the dimensionality from $8C$ to 2, producing binary predictions of authenticity, denoted as F_C , as shown in equations (8) and (9). Beyond predicting detection results, the same feature maps are employed to output a low-dimensional mosaic prediction $P_m(\frac{H}{32} \times \frac{W}{32} \times 1)$. This operation involves flattening the features F to obtain $(\frac{H}{32} \times \frac{W}{32})$ vectors of dimensions $8C$ each. Each vector undergoes linear transformation to reduce it to a 1-dimensional value. Subsequently, a sigmoid operation constrains the value to the range (0,1), yielding a low-dimensional mosaic prediction map, as shown in equation (10). This approach aims to compel the model to not only focus on global features but also to attend to local features, thereby generating local features that exhibit more diversity due to spatial differences. This step necessitates the guidance of the corresponding local diversity loss function to regulate the generation of local features, as will be elaborated in subsequent sections.

$$F_C = \text{AdaptiveAvgPooling2d}(F) \quad (8)$$

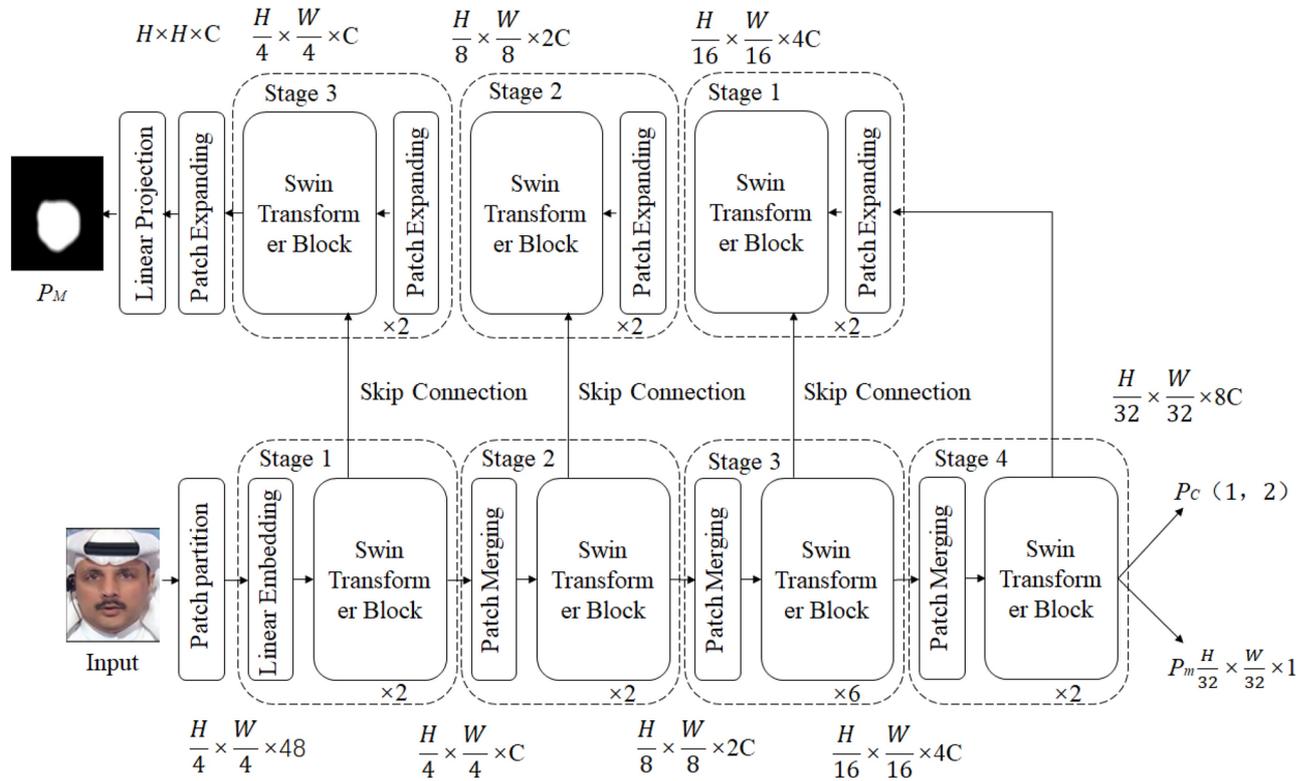


Fig. 4. Deepfake Detection and Localization Model. (The face images shown in this figure are sourced from the FaceForensics++ (FF++) dataset³⁶).

$$P_C = FC(F_C) \tag{9}$$

$$P_m = Sig(Line(Flatten(F))) \tag{10}$$

The upper part comprises the decoder, which represents the inverse process of the encoder. Within the Patch Expansion Module, instead of employing upsampling methods, linear layers are utilized. Taking the first layer as an example, with input dimensions of $\frac{H}{32} \times \frac{W}{32} \times 8C$, the features are expanded dimensionally to $\frac{H}{16} \times \frac{W}{16} \times 16C$, doubling the original dimensions. Subsequently, the shape is adjusted to $\frac{H}{16} \times \frac{W}{16} \times 4C$, as depicted in equation (11), accomplishing the expansion operation. Throughout this process, there exists a skip connection step where the output of the third stage of the encoder is concatenated with the features expanded in the first stage of the decoder. Subsequently, dimensionality is restored through linear transformation, as shown in equation (12). This approach aims to retain shallow texture feature information, facilitating the training and learning process of the model. Finally, after passing through a linear layer, a matrix of dimensions $H \times W \times 1$ is obtained. Each pixel's value is then constrained to the range from 0 to 1 via a sigmoid operation. Consequently, a grayscale image is obtained, as depicted in equation (13).

$$Expend(F) = Rearrange(Line(F)) \tag{11}$$

$$Fd_1 = Line(Concat(F_3, Expend(F))) \tag{12}$$

$$P_M = Sig(Line(Decoder(F))) \tag{13}$$

C. Training and Loss Function

During the training process of tampering localization detection, guidance is needed from labels. However, current deepfake face datasets lack precise mosaic labels to annotate tampered regions. In the FF++ dataset, only datasets generated using the Deepfakes method provide mosaics, but these mosaics only enclose the entire face with a square box, which does not provide clear guidance for training tampering localization detection. This is because these areas may include parts that have not been tampered with, thus affecting the model's feature extraction.

Therefore, this paper proposes the use of self-blending to simulate deepfake methods through simple image processing techniques, synthesizing more synthetic faces with mosaic labels. By training the model using this data and labels, it can effectively locate tampered areas. The training process of the model is illustrated in Fig. 5.

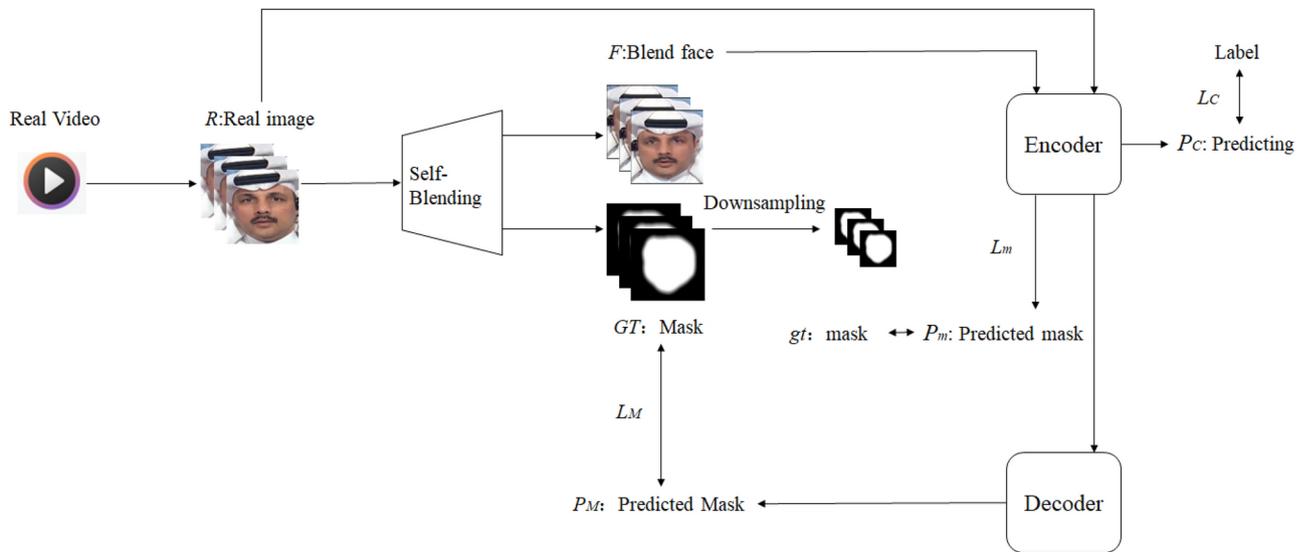


Fig. 5. Training Process. (The face images shown in this figure are sourced from the FaceForensics++ (FF++) dataset³⁶).

In the process, the first stage involves preprocessing, where all frames of the dataset's real videos are extracted into images. Subsequently, the entire blending process is performed in real-time alongside the training process, meaning that random data augmentation and random synthesis are applied to the required data for each training iteration, ensuring data diversity.

Within one training iteration:

1. Eight frames are randomly sampled from each video, and self-blending operations are performed on each image to obtain synthetic faces F and synthetic mosaic ground truth GT .
2. The GT is downsampled to obtain gt , which has the same size as the encoder output P_m and serves as the label for P_m .
3. Both real faces R and synthetic faces F are fed into the tampering detection model.
4. The model's predicted authenticity P_C from the encoder output is compared with the real labels, and the predicted feature map mosaic P_m is obtained.
5. The encoder output is directly fed into the decoder to obtain the predicted blending area P_M , which is then compared with the real synthetic mosaic GT .
6. The next video is selected, and the training continues from the first step until all videos have been sampled, completing one training round. In Fig. 5, three training loss functions are depicted: classification loss function \mathcal{L}_c , local diversity loss function \mathcal{L}_m , and tampering localization loss function \mathcal{L}_M . The classification loss function \mathcal{L}_c treats the task as a binary classification problem, where 0 represents real images and 1 represents fake images. Therefore, the classification loss function employs binary cross-entropy loss:

$$\mathcal{L}_C = -y_{gt} \log(P_C) - (1 - y_{gt}) \log(1 - P_C), \quad (14)$$

y_{gt} represents the true label value. When the predicted label is close to the true label, the loss function tends towards 0. Conversely, when there is a significant difference between the predicted label and the true label, the loss function will be larger. Therefore, through this loss function constraint, as the loss function value decreases, the predicted label becomes closer to the true label.

The local diversity loss function is primarily designed to encourage the model to learn local spatial differences rather than solely focusing on global features. Local diversity refers to the existence of certain differences between the blending area and the regions outside it. The local diversity loss function constrains the local features of these two parts to exhibit certain differences, as illustrated in Fig. 6.

Due to the nature of the Swin-Transformer in generating features, even though the image is partitioned into blocks, and despite the fusion stage, the relative positions of the obtained feature maps do not change compared to the original image. Therefore, the feature vector at each position in the feature map outputted by the encoder is regarded as the local feature at that position, as illustrated by the third image in Fig. 6.

The difference between feature vectors can be quantified using the dot product of vectors, as shown in equation (15). If the dot product of two vectors is larger, it indicates that the difference between the two vectors is smaller; conversely, if the dot product of two vectors is smaller, it suggests that the difference between the two vectors is larger. By calculating the difference value of each vector with every other vector, we obtain $h^2 \times w^2$ difference values. Parallel computation, as depicted in equation (16), yields an $h^2 \times w^2$ matrix containing all

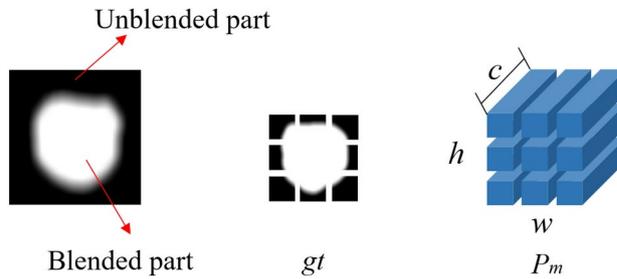


Fig. 6. Local diversity.

difference values. Finally, a sigmoid operation is applied to all difference values to constrain them to the range (0, 1).

$$Sim(i, j) = P_m(i) \cdot P_m(j) \quad (15)$$

$$Sim_{pred} = Sig(P_m \cdot P_m^T) \quad (16)$$

If two feature vectors belong to the same blending area (either inside or outside), they should be made as close to each other as possible. Conversely, if two feature vectors belong to different areas, their difference should be maximized. This aspect is guided by the ground truth gt , where the values range between 0 and 1, with darker areas closer to 1 and lighter areas closer to 0. By taking the absolute difference between the values at two positions in gt , the larger the difference, the greater the disparity between the two positions. Therefore, from equation (17), we obtain the local diversity label Sim_{gt} , where $Flatten$ denotes the flattening operation, transforming gt into a vector of dimensions $h \times w$. By subtracting this vector from its transpose, we obtain an $h^2 \times w^2$ matrix, which stores the difference values between each position and every other position.

$$Sim_{gt} = 1 - |Flatten(gt) - Flatten(gt)^T| \quad (17)$$

The final local diversity loss function is represented as equation (18). The predicted difference matrix Sim_{pred} is subtracted from the ground truth difference matrix Sim_{gt} at corresponding positions to obtain their difference. Then, the average of these differences is calculated to serve as the loss function value.

$$\mathcal{L}_m = Mean(|Sim_{gt} - Sim_{pred}|) \quad (18)$$

The final loss function is the tampering localization loss function LM . Since P_M obtained from the decoder is a matrix of shape $(H, W, 1)$, with each value ranging from 0 to 1, it directly corresponds to the generated mosaic ground truth GT . By flattening both P_M and GT , the cross-entropy is calculated at each position, and then the average is taken to obtain the difference between the predicted P_M and the ground truth GT , as shown in equation (19).

$$\mathcal{L}_M = -\frac{1}{HW} \sum_i^{HW} (GT[i] \log(P_M[i]) + (1 - GT[i]) \log(1 - P_M[i])) \quad (19)$$

The final composite loss function \mathcal{L} used during training is derived by combining the three individual loss functions. It is represented by equation (20).

$$\mathcal{L} = \alpha \mathcal{L}_C + \beta \mathcal{L}_M + \gamma \mathcal{L}_m \quad (20)$$

Experiments

A. Datasets

The primary training dataset used in this experiment is the FaceForensics++ (FF++) dataset³⁶, which consists of 1000 publicly available original videos, each lasting several seconds. It encompasses four different deepfake methods: FaceSwap, Deepfakes, Face2Face, and NeuralTextures, with 1000 videos forged using each method, totaling 4000 forged videos. FaceSwap and Deepfakes are face swapping methods, while Face2Face and NeuralTextures are facial expression replacement methods. The dataset also includes three different compression qualities: C0, C23, and C40. C0 represents the lossless original version, C23 represents the high-quality HQ version, and C40 represents the low-quality LQ version. Due to the large quantity, high quality, diverse forging methods, and clear labels of this dataset, as well as the different compression versions, it has been widely used in deepfake video detection. This experiment also employs this dataset for model training and evaluation. During training, the ratio of training set to validation set to test set is 8:1:1. As the training method uses non-forged samples, the model is trained entirely using real videos, with fake images obtained through self-blending. During testing, both real and fake videos from the dataset are used.

Furthermore, to demonstrate the model's generalization ability, cross-database testing is also conducted. In addition to the FF++ dataset, several other high-quality open-source datasets are utilized for testing and evaluating the model's generalization performance. These datasets include DFDC and DFDCP, as well as the Celeb-DF³⁸ dataset. The DFDC is a significant event initiated by Facebook, in collaboration with Microsoft, the Massachusetts Institute of Technology (MIT), and several other renowned companies and universities. This event launched a highly appealing facial forgery detection challenge on the Kaggle platform. The DFDC dataset is immense, encompassing over 20,000 real videos and more than 100,000 forged videos. Unfortunately, these videos lack clear forgery classifications and generally suffer from poor quality, significantly increasing the difficulty of the detection task. DFDCP represents the high-quality videos within the DFDC, comprising 1,141 authentic videos and 4,073 forged videos. Meanwhile, Celeb-DF is a dataset focused on high-quality forged videos, employing advanced deepfake techniques to produce 590 high-quality samples and expanding to include 5,639 high-quality forged videos.

B. Experimental Setup and Parameters

Preprocessing: Firstly, all real videos from the FF++ dataset are converted into frames using a frame extraction tool. The Dlib library³⁸ is utilized for detecting facial key points, locating facial regions, and obtaining key points coordinates. Subsequently, based on the facial key points coordinates, the images are cropped to retain only the facial region. Finally, the size of the images is resized to (224, 224).

Training: The tampering localization detection model, based on the Swin-UNET architecture with modifications, is trained. The minimum model has encoder stages with 2, 2, 6, 2 layers, and decoder stages with 2, 2, 2 layers. Additionally, experiments involve enlarging the model by adjusting the encoder stages to have 2, 2, 18, 2 layers. The weights of the encoder part are initialized using pretrained weights of Swin-transformer on ImageNet³⁹. The training optimizer is SAM⁴⁰ with a batch size of 16 and a learning rate of 0.001 for 100 epochs. During training, data augmentation operations are applied to both real and self-blended images using methods from the Albumentations library⁴¹, such as color changes, saturation changes, sharpening, scaling, cropping, downsampling, etc.

Detection: When detecting the authenticity of a video, 32 frames are sampled from each video, and each frame is individually evaluated to provide a judgment. The authenticity of the video is determined by averaging the results of these 32 frames.

All experiments are conducted on an NVIDIA 3090 GPU with 16GB of memory.

C. Performance Testing and Analysis

Firstly, the effectiveness of the multi-part local deformation self-blending algorithm is demonstrated. The proposed multi-part local deformation self-blending algorithm generates training data by blending real faces with themselves, resulting in data with forged features resembling those in real datasets, along with corresponding synthetic mosaic regions. The images processed by the multi-part local deformation self-blending algorithm are shown in Fig. 7. In this figure, the source face represents the original face, and all self-blending methods are

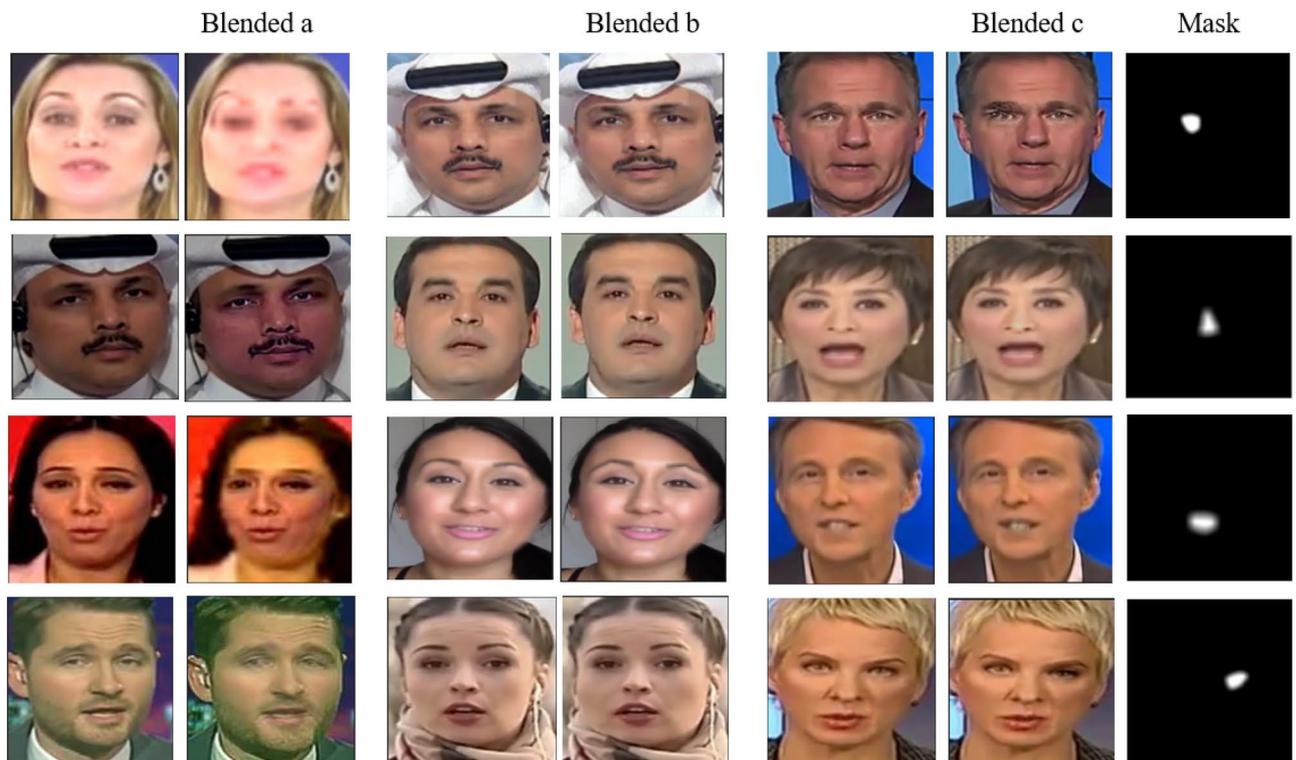


Fig. 7. Demonstration of Multi-part Local Deformation Self-blending. (The face images shown in this figure are sourced from the FaceForensics++ (FF++) dataset³⁶).

Method	Training Set		Test Set AUC (%)			
	Real	Fake	FF++DF	FF++F2F	FF++FS	FF++NT
Xception ³⁶	✓	✓	96.1	92.1	93.8	85.7
F3Net ¹²	✓	✓	98.0	95.2	96.0	88.4
Two-branch ¹⁹	✓	✓	98.8	96.0	97.4	88.6
LipForensics ²⁵	✓	✓	99.7	90.1	99.7	99.1
Face X-Ray ¹⁵	✓		99.17	98.57	98.21	98.13
Ours	✓		99.95	98.82	99.88	99.13

Table 1. In-dataset Evaluation on FF++.

Method	Training Set		Test Set AUC (%)		
	Real	Fake	CDF	DFDCP	DFDC
Xception ³⁶	✓	✓	65.5	-	59.4
F3Net ¹²	✓	✓	67.8	-	57.9
Two-branch ¹⁹	✓	✓	73.4	-	60.8
FTCN ²²	✓	✓	86.9	-	74
LipForensics ²⁵	✓	✓	82.4	73.5	-
Face X-Ray ¹⁵	✓		80.58	71.15	-
Ours	✓		87.52	72.25	80.34

Table 2. Cross-dataset Evaluation.

applied based on this source. Column “Self-blend a” mainly demonstrates self-blending applied to the entire face, showing clear differences inside and outside the blending region, thus reflecting the spatial inconsistency mentioned in Chapter 2. Column “Self-blend b” primarily showcases local deformations applied to specific parts of the face, such as facial contours, mouth, and eyes, simulating the forged features in expression manipulation and texture editing forgery methods. Column “Self-blend c” illustrates self-blending applied to different parts of the face, such as eyes, nose, and mouth. The resulting mosaics will vary correspondingly based on the different blending regions.

Subsequently, comparative experiments are conducted to verify the model’s discrimination ability against deepfake videos. Table 1 presents the in-set detection result. The model proposed in this chapter is trained on the lossless training set of FF++, utilizing only real data and employing the multi-part local deformation self-blending algorithm to generate fake data for training. Then, testing is performed on the test sets of FF++ with four different forgery methods. Among the comparison algorithms, Face X-ray adopts a training method similar to that of this paper, utilizing only non-forged samples for training, while other comparison algorithms use both real and forged data from FF++ for training. The model proposed in this chapter performs excellently on the DF and FS datasets but relatively poorly on the F2F and NT datasets. This is mainly because the data features generated by the multi-part local deformation self-blending method proposed in this chapter are more similar to those in these two datasets.

Then, cross-dataset testing is conducted as shown in Table 2, where the experiment involved training on the FF++ dataset and testing on other datasets. As shown in the table, the model exhibits a certain level of detection capability in datasets beyond those used for training. It performs exceptionally well in CDF and DFDC datasets, indicating that the self-blending method contributes to the model’s generalization ability, and the model can learn forged features effectively.

To comprehensively evaluate the model’s performance, we conducted comparative experiments by replacing the Encoder component of the Swin-UNet architecture while maintaining its original Decoder structure. We implemented and tested three variants: the original Swin-UNet, and two modified versions using ResNet50 and Vision Transformer Small (ViT-S) as feature extractors. The experiments were conducted across four different manipulation scenarios from the FaceForensics++ dataset: DeepFakes (FF++DF), Face2Face (FF++F2F), FaceSwap (FF++FS), and NeuralTextures (FF++NT). As shown in Table 3, the original Swin-UNet consistently demonstrated superior performance, achieving the highest AUC scores across all four manipulation types. Specifically, it achieved AUC scores of 99.95% for FF++DF, 98.82% for FF++F2F, 99.88% for FF++FS, and 99.13% for FF++NT, outperforming both the ResNet50 and ViT-S variants. This superior performance can be attributed to Swin-UNet’s hierarchical feature extraction capability and its efficient handling of both local and global dependencies in the image, making it particularly effective for detecting various types of facial manipulations.

To evaluate the effectiveness of each step, ablation experiments were conducted separately on the four forgery datasets of FF++, as shown in Table 4. It can be observed that adding these two steps helps improve the model’s detection accuracy. Moreover, after incorporating local displacement deformation, the detection accuracy is enhanced, particularly for the F2F and NT forgery datasets, where the forgery methods involve expression

Method	Test Set AUC (%)			
	FF++DF	FF++F2F	FF++FS	FF++NT
Resnet50 ⁴²	96.1	94.2	95.8	94.2
Vit-s ⁴³	97.0	95.3	96.2	95.4
Ours	99.95	98.82	99.88	99.13

Table 3. Evaluation on FF++ for Different Backbone.

Method		Test Set AUC (%)			
Multi-Part	Local Displacement Deformation	FF++DF	FF++F2F	FF++FS	FF++NT
		96.1	94.2	95.8	94.2
✓		97.0	95.3	96.2	95.4
✓	✓	99.95	98.82	99.88	99.13

Table 4. Ablation on FF++.

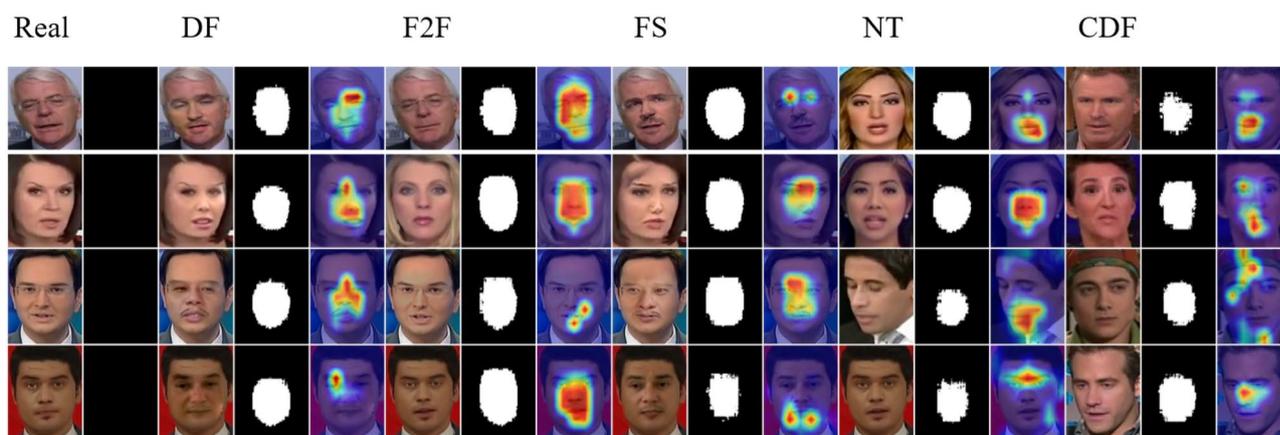


Fig. 8. Tampering Localization and Heatmap Visualization. (In this figure, the face images in the last column are sourced from the CDF dataset³⁷, whereas all other face images are from the FaceForensics++ (FF++) dataset³⁶).

manipulation and texture editing. This indicates the effectiveness of the proposed method in detecting these forgery techniques.

Finally, tampering localization results and visualization experiments were conducted. The data used in this experiment were entirely generated through self-blending methods, simulating spatial inconsistencies using synthetic synthesis. By learning such spatial inconsistencies, the model achieves tampering localization. When applying the trained model to real datasets, it can identify tampered regions, as shown in Fig. 8. The first four rows represent data from the four forgery methods in FF++, and the last row represents data from CDF. In the first column, genuine images are shown, where the model detects tampered regions as pure black, indicating no tampering detected. In the third column, fake images are displayed, with the predicted tampered regions concentrated on the inner side of the face. This demonstrates that the model trained on data generated through multi-part local deformation self-blending can effectively detect tampered regions in real datasets, affirming the effectiveness of the training method proposed in this paper.

Visualization will utilize the GradCAM++ method⁴¹, applied to the tampering localization detection model proposed in this paper, to reveal the regions the model focuses on when making judgments. The heatmap in the last column of Figure 7 shows the areas of attention of the model when detecting the four different forgery methods in FF++. It can be observed that the model's attention is primarily concentrated on the tampered regions of the face, affirming the effectiveness of the model in tampering localization detection.

Conclusion

This paper aims to address the threats posed by deepfake technology to society and content security, with a particular focus on improving methods for detecting and localizing deepfake videos. Currently, deep learning methods suffer from insufficient generalization and the lack of fine-grained labels, which affects localization. To tackle these issues, this paper proposes a multi-location local displacement self-mixing method to generate more diverse deepfake feature data and simultaneously generate mixed region labels to more accurately guide the

model in learning fake regions. Furthermore, by employing the Swin-Unet model, the paper achieves tampering localization detection, introducing classification loss functions, local difference loss functions, and tampering localization loss functions to effectively improve the precision of localization and detection. Experimental results demonstrate that this self-mixing method can effectively simulate fake features in real datasets and achieve good detection accuracy on actual datasets, while also enabling fake region prediction, indicating the effectiveness of this training method. Ultimately, an integrated approach for detecting and localizing deepfake videos is realized, exhibiting high detection accuracy and generalization capability.

While extensive experiments were conducted on mainstream datasets, it is important to acknowledge that these datasets do not cover the entire breadth of deepfake techniques. Consequently, the achieved results, although promising, still present certain limitations, primarily manifested in the method's generalization capability. This limitation reflects one of the inherent challenges of deep learning-based methods—namely, that training and testing on a finite range of data may not fully capture the variations encountered in real-world scenarios. Looking ahead, expanding the range of datasets and exploring more diverse fake content will be a central focus of our future work to enhance the model's robustness and adaptability. We aim to further refine the proposed self-mixing approach with advanced data augmentation, domain adaptation strategies, and additional architectures capable of handling novel and highly sophisticated types of deepfake attacks.

Data availability

The following statement contains information about the datasets and pre-trained models used in this paper: 1. FaceForensics++ are available at the following URL: <https://github.com/ondyari/FaceForensics> 2. Celeb-DF is available at the following URL: <https://github.com/yuezunli/celeb-deepfakeforensics> 3. The pre-trained model trained in this paper which is named “swin_small_patch4_window7_224.pth” is available at the following URL: https://pan.baidu.com/s/1nHD5QUYF_2MLlGbhGZv52w?pwd=1234

Received: 18 September 2024; Accepted: 28 January 2025

Published online: 31 January 2025

References

1. T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” *arXiv preprint[SPACE]arXiv:1710.10196*, 2017.
2. T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.
3. Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.
4. Y. Gao, F. Wei, J. Bao, S. Gu, D. Chen, F. Wen, and Z. Lian, “High-fidelity and arbitrary face editing,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 115–16 124.
5. Y. Nirkin, Y. Keller, and T. Hassner, “Fsgan: Subject agnostic face swapping and reenactment,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7184–7193.
6. L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, “Advancing high fidelity identity swapping for forgery detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5074–5083.
7. J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, “Face2face: Real-time face capture and reenactment of rgb videos,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2387–2395.
8. B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, “The deepfake detection challenge (dfdc) dataset,” *arXiv preprint[SPACE]arXiv:2006.07397*, 2020.
9. H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, “Multi-attentional deepfake detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2185–2194.
10. C. Wang and W. Deng, “Representative forgery mining for fake face detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 923–14 932.
11. S. Dong, J. Wang, R. Ji, J. Liang, H. Fan, and Z. Ge, “Implicit identity leakage: The stumbling block to improving deepfake detection generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3994–4004.
12. Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, “Thinking in frequency: Face forgery detection by mining frequency-aware clues,” in *European conference on computer vision*. Springer, 2020, pp. 86–103.
13. J. Li, H. Xie, J. Li, Z. Wang, and Y. Zhang, “Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6458–6467.
14. C. Tian, Z. Luo, G. Shi, and S. Li, “Frequency-aware attentional feature fusion for deepfake detection,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
15. L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, “Face x-ray for more general face forgery detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5001–5010.
16. T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia, “Learning self-consistency for deepfake detection,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 023–15 033.
17. L. Chen, Y. Zhang, Y. Song, L. Liu, and J. Wang, “Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 710–18 719.
18. K. Shiohara and T. Yamasaki, “Detecting deepfakes with self-blended images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 720–18 729.
19. Masi, I. et al. 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16. Springer 2020, 667–684 (2020).
20. Z. Gu, Y. Chen, T. Yao, S. Ding, J. Li, F. Huang, and L. Ma, “Spatiotemporal inconsistency learning for deepfake video detection,” in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 3473–3481.
21. Z. Sun, Y. Han, Z. Hua, N. Ruan, and W. Jia, “Improving the efficiency and robustness of deepfakes detection through precise geometric features,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3609–3618.
22. Y. Zheng, J. Bao, D. Chen, M. Zeng, and F. Wen, “Exploring temporal coherence for more general video face forgery detection,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 044–15 054.
23. Z. Gu, T. Yao, Y. Chen, S. Ding, and L. Ma, “Hierarchical contrastive inconsistency learning for deepfake video detection,” in *European Conference on Computer Vision*. Springer, 2022, pp. 596–613.

24. Guan, J. et al. Delving into sequential patches for deepfake detection. *Advances in Neural Information Processing Systems* **35**, 4517–4530 (2022).
25. A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, “Lips don’t lie: A generalisable and robust approach to face forgery detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5039–5049.
26. H. Cheng, Y. Guo, T. Wang, Q. Li, X. Chang, and L. Nie, “Voice-face homogeneity tells deepfake,” *arXiv preprint[SPACE]arXiv:2203.02195*, 2022.
27. Yang, W. et al. Avoid-df: Audio-visual joint learning for detecting deepfake. *IEEE Transactions on Information Forensics and Security* **18**, 2015–2029 (2023).
28. C. Feng, Z. Chen, and A. Owens, “Self-supervised video forensics by audio-visual anomaly detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 491–10 503.
29. Zhang, Y., Lin, W. & Xu, J. Joint audio-visual attention with contrastive learning for more general deepfake detection. *ACM Transactions on Multimedia Computing, Communications and Applications* **20**(5), 1–23 (2024).
30. J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. Manjunath, “Exploiting spatial structure for localizing manipulated image regions,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4970–4979.
31. P. Chen, J. Liu, T. Liang, C. Yu, S. Zou, J. Dai, and J. Han, “Dlmmnet: End-to-end detection and localization of face manipulation using multi-domain features,” in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.
32. Kong, C. et al. Detect and locate: Exposing face manipulation by semantic-and noise-level telltales. *IEEE Transactions on Information Forensics and Security* **17**, 1741–1756 (2022).
33. C. Shuai, J. Zhong, S. Wu, F. Lin, Z. Wang, Z. Ba, Z. Liu, L. Cavallaro, and K. Ren, “Locate and verify: A two-stream network for improved deepfake detection,” in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 7131–7142.
34. H. Zhao, B. Liu, Y. Hu, J. Li, and C.-T. Li, “Hybrid domain meta-learning network for face forgery detection and localization in deepfakes,” in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8.
35. H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, “Swin-unet: Unet-like pure transformer for medical image segmentation,” in *European conference on computer vision*. Springer, 2022, pp. 205–218.
36. A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “Faceforensics++: Learning to detect manipulated facial images,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1–11.
37. Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, “Celeb-df: A large-scale challenging dataset for deepfake forensics,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3207–3216.
38. King, D. E. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research* **10**, 1755–1758 (2009).
39. Deng, J. et al. IEEE conference on computer vision and pattern recognition. Ieee 2009, 248–255 (2009).
40. Buslaev, A. et al. Albumentations: fast and flexible image augmentations. *Information* **11**(2), 125 (2020).
41. Chattopadhyay, A., Sarkar, A., Howlader, P., Balasubramanian, V. N. “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks,” in *IEEE winter conference on applications of computer vision (WACV)*. IEEE 2018, 839–847 (2018).
42. K. He, X. Zhang, S. Ren, et al. “Deep residual learning for image recognition,” *Proceedings of the IEEE conference on computer vision and pattern recognition* 2016: 770–778.
43. A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint[SPACE]arXiv:2010.11929*, 2020.
44. Luo, A. et al. “Beyond the Prior Forgery Knowledge: Mining Critical Clues for General Face Forgery Detection.,” *IEEE Transactions on Information Forensics and Security* **19**, 1168–1182 (2024).
45. Chenqi Kong, Shiqi Wang, Haoliang Li, Digital and Physical Face Attacks: Reviewing and One Step Further, [arXiv:2209.14692](https://arxiv.org/abs/2209.14692), 2022.
46. C. Kong, H. Li and S. Wang, “Enhancing General Face Forgery Detection via Vision Transformer with Low-Rank Adaptation,” *2023 IEEE 6th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, Singapore, pp. 102–107, 2023.
47. Chenqi Kong, Anwei Luo, Shiqi Wang, Haoliang Li, Anderson Rocha, Alex C. Kot, Pixel-inconsistency modeling for image manipulation localization, [arXiv:2310.00234](https://arxiv.org/abs/2310.00234), 2023.
48. Chenqi Kong, Anwei Luo, Peijun Bao, Yi Yu, Haoliang Li, Zengwei Zheng, Shiqi Wang, Alex C. Kot, MoE-FFD: Mixture of Experts for Generalized and Parameter-Efficient Face Forgery Detection, [arXiv:2404.08452](https://arxiv.org/abs/2404.08452), 2024.
49. Chenqi Kong, Anwei Luo, Peijun Bao, Haoliang Li, Renjie Wan, Zengwei Zheng, Anderson Rocha, Alex C. Kot, Open-Set Deepfake Detection: A Parameter-Efficient Adaptation Method with Forgery Style Mixture, [arXiv:2408.12791](https://arxiv.org/abs/2408.12791), 2024.
50. Anwei Luo, Rizhao Cai, Chenqi Kong, Yakun Ju, Xiangui Kang, Jiwu Huang, Alex C. Kot, Forgery-aware Adaptive Vision Transformer for Face Forgery Detection, [arXiv:2309.11092v1](https://arxiv.org/abs/2309.11092v1), 2023.

Acknowledgements

The facial images used in this paper are from two publicly available datasets. For Figures 1, 5, 7, and the first three columns of Figure 8, the images are sourced from FaceForensics++ Dataset (Rössler et al., 2019)³⁶, which is available at <https://github.com/ondyari/FaceForensics>. For the last column of Figure 8, the images are sourced from Celeb-DF Dataset (Li et al., 2020)³⁷, which is available at <https://github.com/yuezunli/celeb-deepfakeforensics>. Both datasets are widely used in the deepfake detection research community and permit the use of images in academic publications with proper attribution.

Author contributions

Wenqing Shang and Weiguo Lin conceived the experiments, Junfeng Xu, Xintao Liu and Yuefeng Wang conducted the experiments, Junfeng Xu and Xintao Liu analysed the results. All authors reviewed the manuscript.

Funding

This work is supported in part by the National Science Foundation of China under Grants U2436208, Fundamental Research Funds for the Central Universities (ID: CUC22GZ034), Public Computing Cloud.

Additional information

Correspondence and requests for materials should be addressed to J.X., W.L. or W.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025