Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.042417

ARTICLE

Multi-Branch Deepfake Detection Algorithm Based on Fine-Grained Features

Wenkai Qin[1], Tianliang Lu[1,*], Lu Zhang[2], Shufan Peng[1] and Da Wan[1]

[1]School of Information Network Security, People's Public Security University of China, Beijing, 100038, China

[2]Department of Investigation, Shandong Police College, Jinan, 250200, China

*Corresponding Author: Tianliang Lu. Email: lutianliang@ppsuc.edu.cn

Received: 29 May 2023 Accepted: 18 August 2023 Published: 31 October 2023

ABSTRACT

With the rapid development of deepfake technology, the authenticity of various types of fake synthetic content is increasing rapidly, which brings potential security threats to people's daily life and social stability. Currently, most algorithms define deepfake detection as a binary classification problem, i.e., global features are first extracted using a backbone network and then fed into a binary classifier to discriminate true or false. However, the differences between real and fake samples are often subtle and local, and such global feature-based detection algorithms are not optimal in efficiency and accuracy. To this end, to enhance the extraction of forgery details in deep forgery samples, we propose a multi-branch deepfake detection algorithm based on fine-grained features from the perspective of fine-grained classification. First, to address the critical problem in locating discriminative feature regions in fine-grained classification tasks, we investigate a method for locating multiple different discriminative regions and design a lightweight feature localization module to obtain crucial feature representations by augmenting the most significant parts of the feature map. Second, using information complementation, we introduce a correlation-guided fusion module to enhance the discriminative feature information of different branches. Finally, we use the global attention module in the multi-branch model to improve the cross-dimensional interaction of spatial domain and channel domain information and increase the weights of crucial feature regions and feature channels. We conduct sufficient ablation experiments and comparative experiments. The experimental results show that the algorithm outperforms the detection accuracy and effectiveness on the FaceForensics++ and Celeb-DF-v2 datasets compared with the representative detection algorithms in recent years, which can achieve better detection results.

KEYWORDS

Deepfake detection; fine-grained classification; multi-branch; global attention

1 Introduction

In recent years, with the continuous deep learning research, deepfake techniques have also made rapid development, especially the generative models such as autoencoder and Generative Adversarial Networks (GAN) [1] have been applied to the technology, which makes it easy to achieve high realism generation and replacement of video faces. This technology which uses deep learning methods to generate high-quality forged videos and images is called "deepfake" [2]. People can generate realistic fake face images using deepfake that are difficult to detect with traditional techniques. Compared with conventional forgery methods, deepfake techniques enrich the details of face forgery and greatly enhance the realism of forged faces, bringing a new challenge to detecting and identifying forged content. These forged video images are widely disseminated on the Internet through social media, which not only violate personal privacy and reputation but also mislead social opinion, confuse the public, and even cause political disputes, posing a significant threat to national security [3]. Therefore, to cope with the security risks brought by deepfake technology, researchers have carried out a series of explorations in the academic community, and the design of a general and efficient deepfake detection algorithm has become one of the research hotspots [4–8].

The deep generative model is at the heart of deep face video forgery techniques represented by "deepfake". The deep generative model can be simply described as a model that uses a deep neural network for data generation to generate new data that is not included in the training dataset. Currently, forgery generation algorithms are divided into two categories. One replaces the face to realize the tampering of identity information, and the other replaces only the expression of the face without changing person's identity information. However, these generation techniques also have obvious drawbacks, such as the presence of noticeable visual artifacts in the generated forged video and the difficulty of maintaining frame-to-frame consistency between consecutive video frames. This is because the characteristics of its probabilistic model, the face generation process often a certain degree of randomness, which makes the synthesized face video has a large jitter, and the final forgery effect still needs to be further improved. Therefore, with the continuous improvement of the generation technology, the new forgery method focuses on solving the various shortcomings existing in the existing techniques, and the quality of the fake face produced in the face forgery process is also continuously improved, making the generated image or video more realistic.

Various deepfake detection methods have been proposed by researchers from different research perspectives, some of which can achieve high identification accuracy on open-source forgery datasets. Most of them model deepfake detection as a binary classification problem, and they tend to extract the global features of an image using a backbone network and then feed them into a binary classifier to distinguish between true and false. Depending on the type of feature selection, these detection methods can be classified as temporal feature-based and spatial feature-based. The temporal feature-based detection methods mainly use Recurrent Neural Networks (RNN) to learn the temporal features between forgery video frames. However, these methods rely heavily on the video's preprocessing and the frame sequence's length, resulting in poor generalization for cross-library detection. At the same time, it can not meet the detection needs in the face of highly compressed videos and complex video backgrounds due to less attention to the local forgery details within frames. Therefore, the detection methods based on spatial features have become the mainstream in deepfake detection tasks due to their better robustness and higher accuracy. These methods transform video detection into an image detection problem by extracting spatial features using Convolutional Neural Networks (CNN) after randomly selecting videos frames. However, in real scenarios, the differences between real and fake samples often exist in subtle ways, such as changes in the curvature of the lips of a face, folds in the facial skin texture, etc. These differences can be easily ignored by these in-frame global feature-based detection algorithms. These detection algorithms based on intra-frame global features can easily overlook these differences, resulting in the model ignoring many discriminative features in the feature extraction phase, which induces the model to make wrong judgments about the authenticity of the samples. In deep forgery detection, due to the extensive presence of forgery traces in different regions of the sample, researchers need to pay more attention to the influence of interactions between various features to avoid monolithic feature learning. In addition, we examined that the commonly used channel or spatial attention mechanisms can only effectively capture local information and cannot establish the functional dependency between space and channel, making it difficult to fully utilize the global information across dimensions. Therefore, this paper makes the detection model focus on critical features more effectively by incorporating global attention mechanism into the backbone network.

Therefore, to solve the difficulties and problems in the detection mentioned above, this paper investigates the method of introducing localized fine-grained features in the backbone network inspired by the idea of fine-grained classification. And to enhance the cross-dimensional interactions of attention information, we embed the global attention mechanism in the backbone and design a multi-branch deepfake detection algorithm. Specifically, the improvements in this paper are mainly carried out in the following three aspects:

1. The recent detection algorithms based on intra-frame features directly extract global features as the source of classification. It is often difficult to focus on the subtle differences between authentic and fake samples. Therefore, to mine these fine-grained features hidden in different regions, we design a feature localization module and use it to transform the backbone into a multi-branch detection model, thus locating multiple discriminative regions and obtaining multi-scale fine-grained information.

2. In the training phase, we introduce the global attention module [9] in the second branch to calculate the model attention weights. The global attention module can enhance the cross-dimensional interaction of spatial and channel information while increasing the weights of crucial feature regions and channels to assign more appropriate attention weights to the feature maps and improve the accuracy of localization and detection. And we validate the effectiveness of embedding the global attention module in the backbone through various ablation experiments.

3. To explicitly explore the relationship and calculate the complementary information between the features of different branch regions, we employ the correlation-guided fusion module in the feature integration phase. And then, under its guidance, we fuse the information of different branch features to provide a diverse basis for identifying genuine and forged samples.

4. In the final integrated decision-making phase, to improve the correlation between standard and forged samples, we formulate a multi-branch loss function, which uses cross-entropy loss as an end-to-end loss function in all three branches.

The structure of our treatise is organized as follows. Section 2 combs through the existing work and relevant references in the field of deepfake detection in recent years and introduces the sources and limitations that inspired us to make algorithmic improvements. Section 3 is the basic methodology and implementation details of our approach, including the principles and model construction for the improvement module. Section 4 provides the details of the parameter settings, environment configurations, dataset processing, and various results of our experiments, which validate the effectiveness of our model. Section 5 concludes with a summary of contributions and proposes future research directions to advance the field of deep forgery detection.

2 Related Work

2.1 Deepfake Detection Based on Deep Learning

To cope with the impact of deepfake techniques on public security, researchers have proposed various deepfake detection methods, which can be divided into detection based on video intraframe features and detection based on video interframe features.

2.1.1 Detection Based on Interframe Features

The detection methods based on interframe timing features to focus on mining the timing features between consecutive frames of the video and classifying the video authenticity by detecting the interframe continuity and integrity. Since deepfake techniques can not forge the video as a whole but tamper with specific regions of the face frame by frame, it is difficult to take into account the sequence of previously forged frames when forging the current frame, which leads to discrepancies in the spatiotemporal distribution of successive frames such as differences in the expressions of the characters, inconsistent spatiotemporal states of the tampered regions, visual artifacts and noise inside successive frames. Such discrepancies can be captured by RNN or other methods based on sequence feature extraction and used to detect actual and forged videos.

Agarwal et al. [10] encoded face expressions, head movements, and other physiological signals, then used a support vector machine (SVM) for detection and classification. Lima et al. [11] employed spatiotemporal convolution to detect temporal and spatial artifacts of forged videos as well as utilized multiple layers of coding units based on the self-attention mechanism to enable each frame feature to aggregate other frame information, thus enabling effective detection of current mainstream forgery datasets. Cozzolino et al. [12] trained an ID-Reveal model using a self-supervised learning paradigm to mine interframe temporal features through ID networks and perform adversarial learning using 3DMM generative networks to identify the authenticity of face videos. Sun et al. [13] proposed a robust framework LRNet to detect forged videos by modeling accurate geometric features.

2.1.2 Detection Based on Intraframe Features

The main idea of the detection methods based on intraframe spatial features is to randomly extract image frames from the video stream, input them into CNN for feature extraction after preprocessing, such as face alignment and face cropping, and finally use the feature information for classification. These approaches first transform the detection object from video to image and use classical models such as ResNet [14], Xception [15], and EfficientNet [16] as backbone networks for classification.

Currently, most deepfake detection is carried out based on interframe spatial features. Afchar et al. [17] found that the artifacts of deepfake images are often present in the middle layer semantic information and proposed using MesoNet combined with the Inception module to extract image middle layer features for detection. Nguyen et al. [18] designed to use the VGG16 model to extract features and then employ a capsule network to learn more detailed features of the face. Nirkin et al. [19] used three encoders based on the Xception structure to encode the whole image, face region, and background region after preprocessing. Finally, they combined the three feature vectors to achieve the effect of detection and localization. Shang et al. [20] constructed a pixel-region relationship network (PRRNet) and used the pixel relationship module and the region relationship module to detect spatial association and inconsistent forgery traces in images.

However, as the forged content becomes more and more realistic, the above binary classification models based on global features have poor detection performance. The reason is that the highly distinguishable discrepancies between forgery samples and original samples often exist in subtleties of the image, such as lip curvature changes and skin texture differences, which are often not easily captured by the detection algorithm based on global features, resulting in reduced detection accuracy. Such subtle and located differences are similar to the fine-grained classification. Therefore, to improve the detection based on spatial features, researchers have started to analyze the deepfake detection task expressed as a fine-grained classification problem.

Zhou et al. [21] designed a detection network based on multi-attention and used bilinear pooling to aggregate low-level texture features and high-level semantic features under the guidance of attention graphs. Guarnera et al. [22] utilized an expectation-maximization algorithm to extract convolutional traces from specific local pixels. Chen et al. [23] used a pixelhop++ module to extract features and reduce the dimensionality of local regions separately in multiple regions of face images and finally integrated and classified the information of each region. Liu et al. [24] divided the original image into several blocks of the same size and randomly shuffled pixels within blocks and positions between blocks, forcing the model to extract more discriminative forgery traces. Based on the lack of physiological constraints in the current generative models, Hu et al. [25] proposed to explore whether the number, shape, and relative positions of the reflected glare of forged eyes are consistent for detection. Guo et al. [26] first located the human eyes' oval pupil mask and then used the IoU algorithm to calculate the difference between the fake pupil shape and the real pupil shape to determine the authenticity of the input image.

In recent years, with the rapid development of Transformers [27], Vision Transformer algorithms [28] that can be used in computer vision have been derived. These algorithms can extract the global association relationship of pixels, and the number of operations required to compute the association between two positions does not increase with distance compared with the CNN. Moreover, the internal self-attention mechanism can generate more interpretable models with stronger modeling ability and detection performance, which have been gradually applied in the study of deepfake detection.

2.2 Fine-Grained Classification

Fine-grained classification is a challenging research task in computer vision, which distinguishes different fine-grained categories by capturing local discriminative features [29,30]. Compared with traditional classification tasks, fine-grained classification provides a more detailed classification of images. Since the discrepancies between categories are minor, the essence of this classification task is to locate the target and local regions and perform feature extraction and processing to complete the training and detection of the classifier.

The key to fine-grained classification is how to locate the subtle discrepancies that exist in local regions. The main research directions can be divided into strongly supervised and weakly supervised approaches For different ways of extracting fine-grained features. The strongly supervised approach uses additional information, such as annotation frames and part annotation points, in addition to category labels in classification. Therefore, it requires more manual effort and time consumption. The weakly supervised approach only uses category labels to complete the model's training. Currently, research in this field mainly focuses on locating discriminative regions in a weakly supervised manner [31–34]. In this paper, we define deepfake detection as a particular fine-grained classification problem, and both have the same characteristics in extracting subtle and discriminative features. However, deepfake detection only involves two categories, i.e., real and fake.

2.3 Attention Mechanism

In deep neural networks, the attention mechanism can mainly be divided into channel attention mechanism and hybrid attention mechanism.

The channel attention mechanism focuses on the correlation between different channels of the feature graph and automatically obtains the importance of each feature channel through network learning, and finally assigns different weight coefficients to each channel to reinforce the critical features and suppress the non-important features. The commonly used are the squeeze-and-excitation

(SE) module [35] and the efficient channel attention (ECA) module [36]. The SE first compresses the spatial dimension of the graph by global average pooling operation and then extracts the channel attention information using a fully connected layer. The ECA, on the other hand, utilizes a 1 × 1 convolution instead of the fully connected layer to achieve information interaction across channels based on the SE.

The hybrid attention mechanism is a more integrated attention method that combines the advantages of channel dimension attention and spatial dimension attention. The commonly used are the bottleneck attention module (BAM) [37], the convolutional block attention module (CBAM) [38], and the coordinate attention (CA) module [39]. The BAM and CBAM can extract the attention information of both channel and spatial dimensions and fuse different attention information. The CA extracts channel attention information from the width and height directions, combining the position information and channel attention information. In 2021, Liu et al. [9] proposed the global attention module (GAM), which still uses the channel-space attention arrangement order. But they redesigned the channel attention submodule and the space attention submodule. When the GAM is embedded in the backbone model, the model cannot only consider the attention weight information extraction in both channel dimension and spatial dimension but also ensure the cross-dimensional information interaction to improve the detection accuracy further.

3 Methods

In this paper, we propose a multi-branch deepfake detection algorithm based on fine-grained features, and the algorithm architecture is shown in Fig. 1.

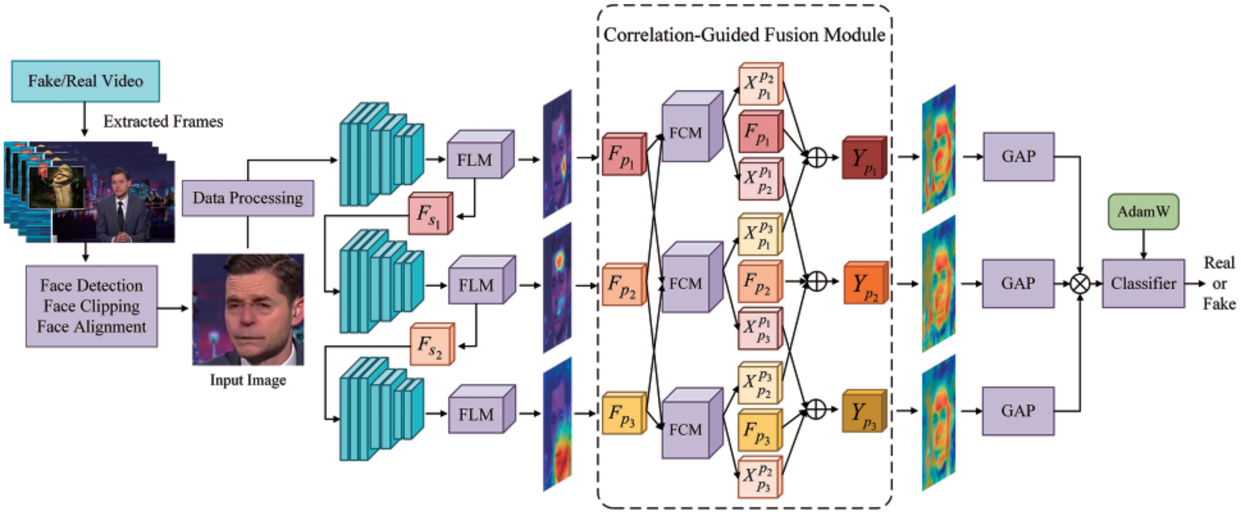

Figure 1: Architecture of multi-branch deepfake detection algorithm based on fine-grained features

As aforementioned, the discrepancies between real and fake faces are usually subtle and occur in local regions, which are not easily captured by the models based on global features. Therefore, to collect local features from different regions more effectively, we design a multi-branch detection framework. In this framework, three main components are embedded in the backbone network: (1) We use the feature localization module (FLM) to locate the subtle discrepancy in different regions. (2) We employ the correlation-guided fusion module (CGFM) to aggregate complementary information from different branches. (3) We introduce the global attention module (GAM) to increase the weights of critical regions and channels and enhance the cross-dimensional interaction. Besides, to learn effectively the fine-grained features of the sample, we specially design a multi-branch loss.

3.1 Feature Localization Module

To locate forgery traces in different regions, we introduce the feature localization module to obtain significant feature representations by mining the most prominent parts of the feature map and then suppressing them to force the model to explore other potential discriminative features in subsequent branches. Moreover, by embedding the feature localization module into different layers of the backbone network, we construct a multi-branch fine-grained feature extraction model, which can obtain feature representations of multiple regions. The specific structure of the feature localization module is shown in Fig. 2.

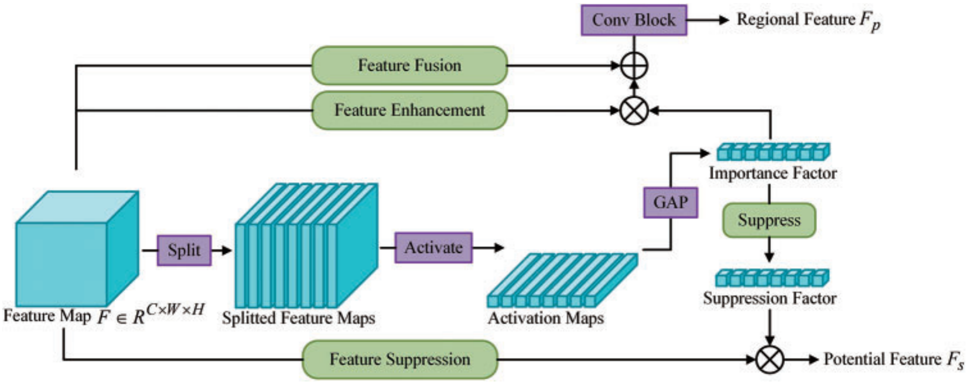

Figure 2: Structure of feature localization module

In the feature localization module, the feature map from a particular layer is first defined as $F \in R^{C\times W\times H}$, where C, W, H denotes the number of channels, width, and height of the feature map, respectively. In the feature map F, it is uniformly partitioned into k striped sections along the width dimension, and each striped section is represented as $F_{(i)} \in R^{C\times(W/k)\times H}, i \in [1, k]$. We define $A_{(i)}$ as a high attention activation map for each part as shown in Eq. (1):

$$A_{(i)} = \mathrm{Re}\,lu\left(\phi\left(F_{(i)}\right)\right) \in R^{1\times(W/k)\times H} \tag{1}$$

where the nonlinear activation function Re*lu* is employed to remove the negative activation parts. ϕ represents the 1×1 convolution, which is used to calculate the importance of each striped section and share the parameters among different parts. In the activation map $A_{(i)}$, to measure the contribution of each striped section to the detection, we use Eq. (2) to gain its average value:

$$b_i = GAP\left(A_{(i)}\right) \in R \tag{2}$$

where GAP denotes the global average pooling. b_i is used as an importance factor, and its value reflects the contribution of the striped sections. If the $b_i$value of a part is larger, it indicates that it is more likely to belong to the critical regional features. The normalization of b_i is performed by the *Soft* max function, which is expressed in Eq. (3):

$$b_i' = Soft\max\left(b_i\right) = \frac{\exp\left(b_i\right)}{\sum\limits_{j\in[1,k]} \exp\left(b_j\right)} \tag{3}$$

where b'_i is the normalized importance factor, which are combined to obtain the feature vector $B' = \left(b'_1, b'_2, \ldots, b'_k\right)^T$. The vector has a global sensory field, which can obtain the statistical information of the feature map F on different striped sections. The most significant region on the feature map F can be represented using the feature vector, and the enhanced feature F_b is obtained by feature fusion. The feature fusion process is shown in Eq. (4):

$$F_b = F + \alpha \times (B' \otimes F) \tag{4}$$

where the magnitude of the weight parameter α reflects the proportion of the most significant region in the feature fusion, and the degree of feature enhancement can be effectively controlled by adjusting α. $\otimes$ denotes element-by-element multiplication. Finally, as shown in Eq. (5), through applying a convolution layer ϕ' on F_b, we can obtain a specific high-concentration feature F_p as follows:

$$F_p = \phi'(F_b) \tag{5}$$

In order for the subsequent modules to continue mining discriminative features in other regions, the suppression factor s_i can be obtained by Eq. (6), which suppress the high attention regions on the current branch and force the subsequent modules of the model to focus more on other regions of the feature map F, and then learn different scale features.

$$s_i = \begin{cases} 1 - \beta, & \mathit{if}\ b'_i = \max(B') \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

where the magnitude of the weight parameter β reflects the degree of suppression of the most significant region in the feature map F. The combination of the suppression factors s_i yields the feature vector $S = (s_1, s_2, \ldots, s_k)^T$. In the current branch, the most significant region of the feature map F is suppressed by S, as shown in Eq. (7), and we can obtain the suppressed potential feature map F_s.

$$F_s = S \otimes F \tag{7}$$

The feature map F_s still contains many discriminative features in other regions except the suppressed high-concentration features. Therefore, to locate these discriminative features, F_s is continued to be fed into the feature localization module of the subsequent branches to capture the multi-scale discriminative features more effectively.

3.2 Correlation-Guided Fusion Module

Although the feature localization module can locate the highly focused part of the sample, the relationship among the features of each branch can be easily ignored because the branch structures are relatively independent of each other. To extract a more robust feature representation, we introduce a correlation-guided fusion module for the feature information learned from different branches. The correlation-guided fusion module consists of several feature complementary modules (FCM), which first explore the complementary information among the features of each branch to avoid ignoring other discriminative features due to the focus on individual local features, and then enhance the feature representation of the current branch through feature fusion guided by the complementary information. The structure of the feature complementation module is shown in Fig. 3.

In the feature complementation module, the regional feature maps F_{p_1} output from any two branches are defined as $F_{p_1} \in R^{C\times W_1\times H_1}$ and $F_{p_2} \in R^{C\times W_2\times H_2}$, where C denotes the number of channels, and $W_1 \times H_1$ and $W_2 \times H_2$ denote the spatial size of each feature map, respectively. To explore the relationship among the features of different branches, we represent the feature vector of each spatial position along the channel dimension of the feature map F as a pixel, and characterizes the correlation among the feature maps by the correlation among pixels, as shown in Eq. (8):

$$pixcel\left(F,i\right)=\left(F_{1,i},F_{2,i},\ldots,F_{C,i}\right)^{T} \tag{8}$$

where $pixcel\left(F,i\right)$ denotes the feature vector of the i^{th} region along the channel dimension on the feature map F, and $F_{j,i}$ denotes the i^{th} region located on the j^{th} channel of the feature map F. After obtaining the feature vectors along the channel dimension for each region on F_{p_1} and F_{p_2}, respectively, we calculate the vector inner product to reflect the similarity among the feature image elements more concisely, as shown in Eqs. (9) and (10):

$$M=f\left(F_{p_1},F_{p_2}\right)=F_{p_1}^{T}F_{p_2} \tag{9}$$

$$M_{i,j}=f\left[pixel\left(F_{p_1},i\right),pixel\left(F_{p_2},j\right)\right]=\left[pixel\left(F_{p_1},i\right)\right]^{T}pixel\left(F_{p_2},j\right) \tag{10}$$

where the matrix M represents the similarity matrix between feature maps F_{p_1} and F_{p_2}, and the element $M_{i,j}$ represents the similarity between the i^{th} pixel of feature map F_{p_1} and the j^{th} pixel of feature map F_{p_2}. If the $M_{i,j}$ between two pixels is smaller, the less similar and more complementary these two pixels are to each other. Therefore, based on the negative correlation between similarity and complementarity, to reflect the complementarity more intuitively, we define the complementarity matrix C as the negative matrix of the similarity matrix M, and normalize the columns and rows of the complementarity matrix through Eqs. (11) and (12) as follows:

$$N_{p1}^{p2}=Soft\max\left(C^{T}\right)\in[0,1]^{W_2H_2\times W_1H_1} \tag{11}$$

$$N_{p2}^{p1}=Soft\max\left(C\right)\in[0,1]^{W_1H_1\times W_2H_2} \tag{12}$$

where N_{p1}^{p2} and N_{p2}^{p1} denote the complementary matrices after normalization by columns and rows, respectively. $W_2H_2\times W_1H_1$ denotes the dimension of N_{p1}^{p2}, and $W_1H_1\times W_2H_2$ denotes the dimension of N_{p1}^{p2}. The respective complementary information can be obtained from the regional features using N_{p1}^{p2} and N_{p2}^{p1}, as shown in Eqs. (13) and (14):

$$X_{p1}^{p2}=F_{p_2}N_{p1}^{p2}\in R^{C\times W_1\times H_1} \tag{13}$$

$$X_{p2}^{p1}=F_{p_1}N_{p2}^{p1}\in R^{C\times W_2\times H_2} \tag{14}$$

where $X_{p_i}^{pj}$ denotes the complementary information extracted from F_{p_j} against F_{p_i}, which is used as the basis for enhancing F_{p_i} in the subsequent aggregation of branch features. To further explore the complementary relationship between the two feature maps, the mining of complementary information operation is transformed as pixel-specific through Eqs. (15) and (16) as follows:

$$pixel\left(X_{p1}^{p2},i\right)=\sum_{j\in\left[1,W_2\times H_2\right]}pixel\left(F_{p_2},j\right)\left(N_{p1}^{p2}\right)_{i,j} \tag{15}$$

$$pixel\left(X_{p2}^{p1},i\right)=\sum_{j\in\left[1,W_1\times H_1\right]}pixel\left(F_{p_1},j\right)\left(N_{p2}^{p1}\right)_{i,j} \tag{16}$$

where each pixel in X_{p1}^{p2} is enhanced by all pixels in F_{p_2}, and each pixel in X_{p2}^{p1} is enhanced by all pixels in F_{p_1}. If the complementarity among the pixels in F_{p_1} and F_{p_2} is stronger, it indicates that the pixels in one of the feature maps contribute more to the pixels in the other feature map. Then each pixel in both can mine semantic complementary information from different regions of the other feature. Therefore, the complementary information of regional features in each branch can be obtained according to Eq. (17):

$$X_{p_i} = \sum_{j \neq i} X_{p_i}^{p_j} \tag{17}$$

where $X_{p_i}^{p_j}$ can be obtained by Eqs. (9), (13) and (14). In addition, as shown in Fig. 3, for the regional feature information F_{p_1} and F_{p_2} in any two branches, the feature complementation module can calculate both the complementary information $X_{p_2}^{p_1}$ and $X_{p_1}^{p_2}$ at the same time. Combining the complementary information from different branches, we can enhance the regional feature information of the current branch through feature fusion to prompt the model focus on different fine-grained features, as Eq. (18) shown in:

$$Y_{p_i} = F_{p_i} + \gamma \times X_{p_i} \tag{18}$$

where Y_{p_i} denotes the enhanced regional features, which contain discriminative information at different scales. The magnitude of the weight parameter γ reflects the proportion of complementary information from different branch features in the feature fusion process.

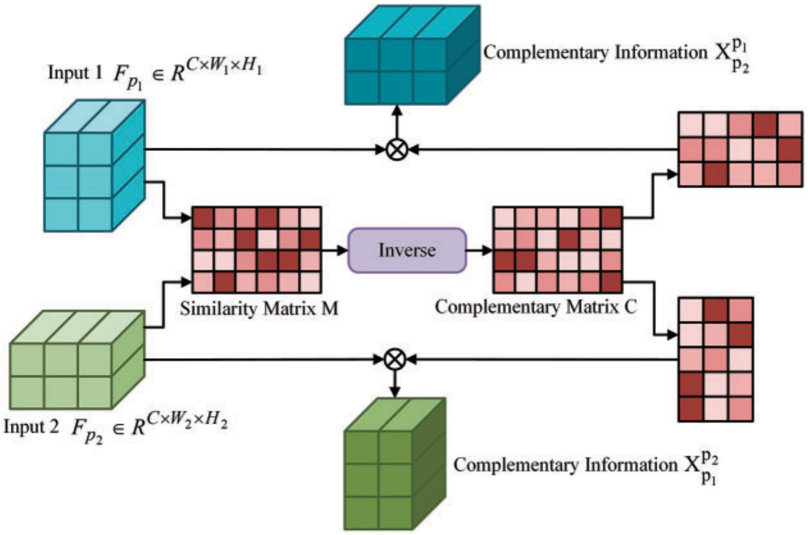
Input 1 $F_{p_1} \in R^{C \times W_1 \times H_1}$
Complementary Information $X_{p_2}^{p_1}$
Inverse
Similarity Matrix M
Complementary Matrix C
Input 2 $F_{p_2} \in R^{C \times W_2 \times H_2}$
Complementary Information $X_{p_1}^{p_2}$

Figure 3: Structure of feature complementary module

3.3 Global Attention Module

In this paper, we choose the Xception as the backbone model, which has a good effect in image classification, and design a multi-branch structure. We find that the suppressed potential feature F_s generated in the previous branch still not only contains a large number of potential discriminative features but also includes some background and noise information. These interferences can impact the localization of subsequent potential discriminative features and eventually lead to a drift between feature localization and the target region.

Therefore, this paper introduces the global attention module in the Middle Flow of Xception to build a more robust network structure by enhancing the model's ability to extract key features of forged images. The global attention module utilizes spatial and channel information at different scales on the feature map, increases the weights of crucial feature regions and feature channels, and enhances the global cross-dimensional interaction of spatial and channel information so that the network can obtain more effective feature representation capability. The structure of GAM is shown in Fig. 4.

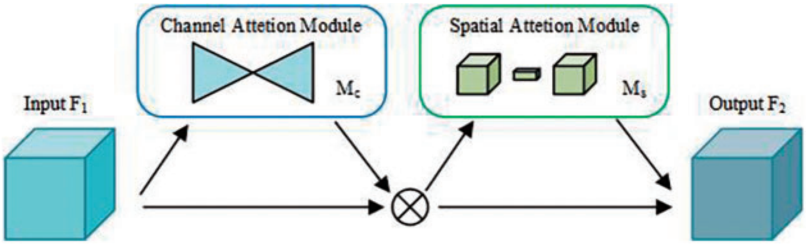

Figure 4: Structure of global attention module

The GAM comprises two submodules: the channel attention module (CAM) and the spatial attention module (SAM). Similar to the CBAM, the GAM belongs to the hybrid attention module that incorporates channel and spatial attention.

In the CAM, the input feature map is given as $F_1 \in R^{C\times W\times H}$, where C, W, H denote the number of channels, width, and height of the feature map, respectively. To obtain global cross-dimensional interaction information, the channel attention module uses a 3D permutation to retain information across three dimensions and get the feature map $F_1' \in R^{W\times H\times C}$. Then F_1' is input to a two-layer multilayer perceptron (MLP), which first compresses the channels as C/r and then reduces them to amplify the cross-dimensional dependence of channel and space. Finally, the feature map dimensions are reduced using another 3D permutation, and the result is element-by-element multiplied by the original feature F_1 after the activation function *sigmiod* to obtain the output $F_2 \in R^{C\times W\times H}$. The calculation process is shown in Eqs. (19) and (20):

$$M_c(F_1) = sigmoid(\phi'(MLP(\phi(F_1)))) = sigmoid(\phi'(W_1(W_0(\phi(F_1))))) \tag{19}$$

$$F_2 = M_c(F_1) \otimes F_1 \tag{20}$$

where *sigmoid* denotes the activation function. The parameters $W_0 \in R^{C\times C/r}$ and $W_1 \in R^{C/r\times C}$ denote the parameter-sharing multilayer perceptron weights. r denotes the compression rate of the hidden layer, and $M_c(F_1)$ denotes the output channel feature vector. The structure of the CAM is shown in Fig. 5.

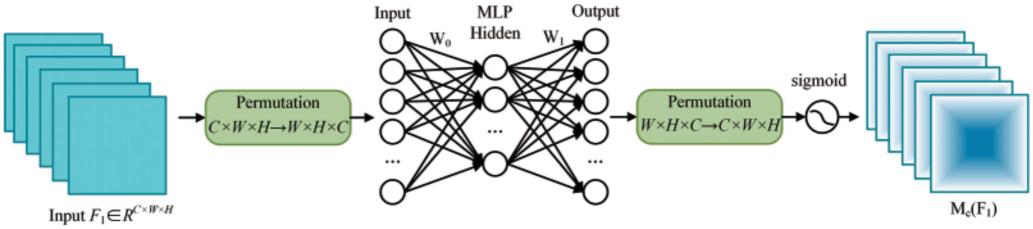
Input
MLP
Hidden
Output
W_0
W_1
Permutation
$C\times W\times H\rightarrow W\times H\times C$
Permutation
$W\times H\times C\rightarrow C\times W\times H$
sigmoid
Input $F_1 \in R^{C\times W\times H}$
$M_c(F_1)$

Figure 5: Structure of channel attention module

In the SAM, the input feature map is the output $F_2 \in R^{C\times W\times H}$ of the channel attention module, where C, W, H denote the number of feature map channels, width, and height, respectively. To focus on spatial information more precisely, the spatial attention module employs two convolutional layers of 7 × 7 for compression of channel dimension and fusion of spatial information. Meanwhile, since max pooling reduces the amount of information, the spatial attention module removes the pooling layer to further preserve the feature information. Besides, the spatial attention module uses grouped convolution on the channel dimension to reduce the negative effect of the significant increase in the number of parameters due to removing of the pooling layer. Finally, the result is element-by-element multiplied with the original feature F_2 after the activation function *sigmoid* to obtain the output $F_3 \in R^{C\times W\times H}$ of the spatial attention module, i.e., the final output of the global attention module. The calculation process is shown in Eqs. (21) and (22):

$$M_s(F_2) = sigmoid\left(Conv_{7\times 7}\left(Conv_{7\times 7}\left(F_2\right)\right)\right) \tag{21}$$

$$F_3 = M_s(F_2) \otimes F_2 \tag{22}$$

where *sigmoid* denotes the activation function. $Conv_{7\times 7}$ denotes the convolutional layer using a convolution kernel of size 7 × 7. The compression rate of the channel between the two convolution layers is represented as r'. $M_s(F_2)$ denotes the output spatial feature vector. The structure of the SAM is shown in Fig. 6.

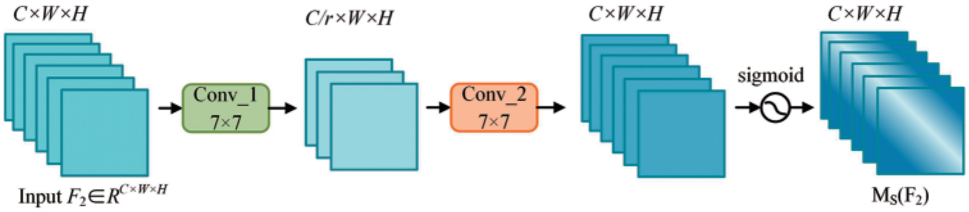

Figure 6: Structure of spatial attention module

In this paper, we embed the GAM into the Block of Xception as a kind of data enhancement module, which makes it possible to modify the Block without affecting the branch structure, as shown in Fig. 7. The model reconstructs the extracted feature maps by embedding the global attention mechanism. It can improve the attention of subsequent branches on essential features and eliminate the interference of background information as much as possible. In addition, the global attention module achieves cross-dimension interaction without reducing dimensionality, which avoids information loss while reducing the number of parameters and computational cost.

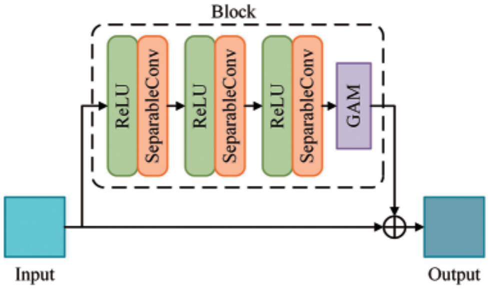

Figure 7: Block structure embedded in GAM

3.4 *Multi-Branch Loss Function*

To assess the validity of the model classification results, this paper designs a multi-branch loss function that uses cross-entropy loss as an end-to-end loss function in all three branches. It is expressed in Eqs. (23)–(25) as:

$$L_i = -y_i \log \left(P_i \left(Y_{p_i}\right)\right) \tag{23}$$

$$P_i \left(Y_{p_i}\right) = soft\max \left(cls_i \left(Y_{p_i}\right)\right) \tag{24}$$

$$L_{total} = L_1 + L_2 + L_3 \tag{25}$$

where L_i denotes the loss of the i^{th} branch. y_i is the true label of the input sample which is represented by the one-hot vector. cls_i denotes the classifier of the i^{th} branch. P_i denotes the classification probability calculated by the i^{th} branch. L_{total} denotes the final loss.

In the training phase, the final loss L_{total} is obtained through calculating the classification loss of the enhanced regional features Y_{p_i} in each branch. We use the AdamW algorithm to optimize the final loss and accelerate the convergence of the model. The three branches together finally make the model learn the various fine-grained features of the forged sample. In the testing phase, the model takes the average prediction scores of all branches as the final prediction result.

4 Experimental Results and Analysis

4.1 *Experiment Environment*

The experimental platform in this paper is a 64-bit Linux operating system with Ubuntu 18.04 LTS, an NVIDIA GeForce RTX 2080Ti graphics card with 11 GB of video memory, an Intel(R) Xeon(R) CPU E5-2678 v3 @ 2.50 GHz, and 62 GB of RAM. The experimental code is implemented in the Pytorch deep learning framework with version 1.11.0, Anaconda version 4.12.0, and Python version 3.8.

4.2 *Introductions of Datasets*

In this paper, we select two mainstream deepfake datasets for experiments, i.e., FaceForensics++ [40] and Celeb-DF [41]. Some samples of datasets are shown in Fig. 8.

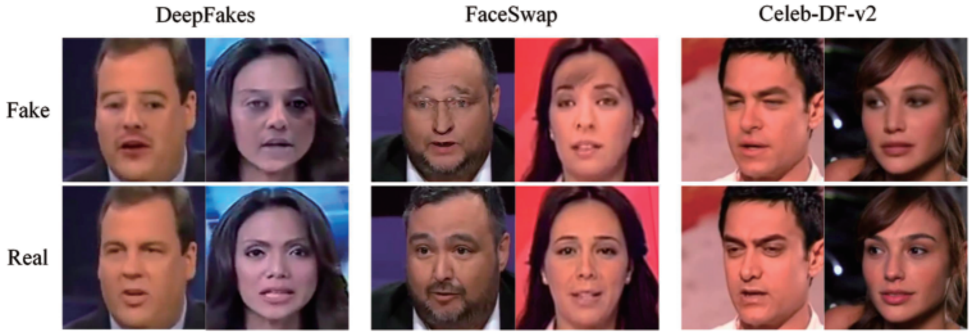

Figure 8: Samples of the three datasets

The FaceForensics++ (FF++) dataset is one of the most significant and widely used datasets. The dataset captures 1000 short videos containing unobstructed faces from YouTube and ensures that the faces are contained in consecutive frames. Then it is manipulated by five forgery techniques

(DeepFakes, Face2Face, FaceSwap, Neural Textures, and FaceShifer) to generate a total of 5000 forged videos with three different compression ratios (lossless compression: Raw, high-quality compression: c23, and low-quality compression: c40). Among them, DeepFakes is based on an autoencoder for face replacement; FaceSwap is a graphics-based approach. To verify the effectiveness of the proposed algorithm for the deepfake detection task, we select the DeepFakes and FaceSwap datasets with a compression rate of c23 in this paper.

The Celeb-DF dataset captures 590 real videos of 59 celebrities from YouTube, which consider people of different genders, ages, and races. It generates 5639 forged videos in MPEG4.0 format with an average length of 13 s using a single DeepFakes approach. Since this dataset can fuse the boundary between the forged and original region by enhancing the face resolution and establishing a color conversion algorithm for the faces in the forged and original videos, it dramatically improves the quality of the forged data. The Celeb-DF dataset is currently recognized as a high-quality deepfake dataset. In this paper, we use the extended Celeb-DF-v2 dataset for our experiments.

4.3 Experiment Setting

Since the forgery regions are primarily concentrated in the facial regions, to make the model focus more on the features of the forgery traces, the experiments in this paper firstly intercept 30 frames at equal intervals for each video in the data preprocessing stage. Second, we use the face detection algorithm RetinaFace to locate the critical points of the five facial features in each frame to determine the facial rectangle. After the face alignment, the images are cropped to 320 × 320 input samples. Then, we employ the sharpening operation and Gaussian blur on the samples to refine the images and reduce image noise. We standardize and normalize the images to enhance the model's generalization ability. Finally, the datasets are divided into train-sets and test-sets in the ratio of 7:3, and the division results are shown in Table 1.

In addition, the specific settings of other hyperparameters are shown in Table 2.

Table 1: Division results of datasets

Datasets	DeepFakes	FaceSwap	Celeb-DF-v2
Train	41936	41945	143192
Test	17979	17976	61368

Table 2: Hyperparameters setting

Parameters	Value
Epoch	40
Batchsize	18
α in FLM	0.5
β in FLM	0.5
γ in CGFM	0.8
Learning rate	1×10^{-3}

4.4 Evaluation Index

In this paper, we use the Accuracy (Acc) and Area under Receiver Operating Characteristic (ROC) Curve (AUC) to evaluate the model effect comprehensively.

Acc is used to describe the classification accuracy of the classifier for genuine and fake samples, and a larger value of Acc indicates a higher correct classification rate of the model. AUC is defined as the probability that the predicted probability value of getting real samples is greater than the probability of fake samples, and a larger value of AUC indicates the better performance of the detection model. The calculation formula of Acc and AUC is shown in (26) and (27):

$$\mathrm{Acc} = \frac{TP + TN}{TP + FP + FN + TN} \tag{26}$$

$$\mathrm{AUC} = \frac{1}{2}\sum_{i=1}^{n-1}\left(\left(\frac{FP}{FP + TN}\right)^{(i+1)} - \left(\frac{FP}{FP + TN}\right)^{(i)}\right) \times \left(\left(\frac{TP}{TP + FN}\right)^{(i+1)} + \left(\frac{TP}{TP + FN}\right)^{(i)}\right) \tag{27}$$

where TP denotes the real face image predicted as true by the model. TN denotes the fake face image predicted as false by the model. FN denotes the real face image predicted as false by the model. FP denotes the fake face image predicted as true by the model. And n denotes the total number of positive and negative samples.

4.5 Analysis of Experimental Results

4.5.1 The Ablation Experiment of Each Improved Strategy in Terms of Performance Gain

In this experiment, we use the Xception as the baseline model and design a set of ablation experiments on the DeepFakes dataset and FaceSwap dataset to verify the performance gains brought by the feature localization module (FLM), correlation-guided fusion module (CGFM), global attention module (GAM) and their combinations to the model detection. The evaluation metrics include Acc value and AUC value. The experimental results are shown in Table 3 and Fig. 9.

Table 3: The classification effect performance gain generated by each improved strategy

Models		DeepFakes		FaceSwap	
		Acc/%	AUC/%	Acc/%	AUC/%
Xception	Baseline	96.81	98.74	97.17	99.10
	+FLM	97.75	99.65	97.84	99.55
	+CGFM	97.47	99.56	97.66	99.60
	+GAM	97.57	99.45	97.62	99.44
	+FLM+CGFM	98.03	99.61	98.23	99.64
	+FLM+GAM	98.11	99.66	98.32	99.61
	+CGFM+GAM	97.98	99.63	97.84	99.54
	+FLM+CGFM+GAM	**98.67**	**99.80**	**98.55**	**99.72**

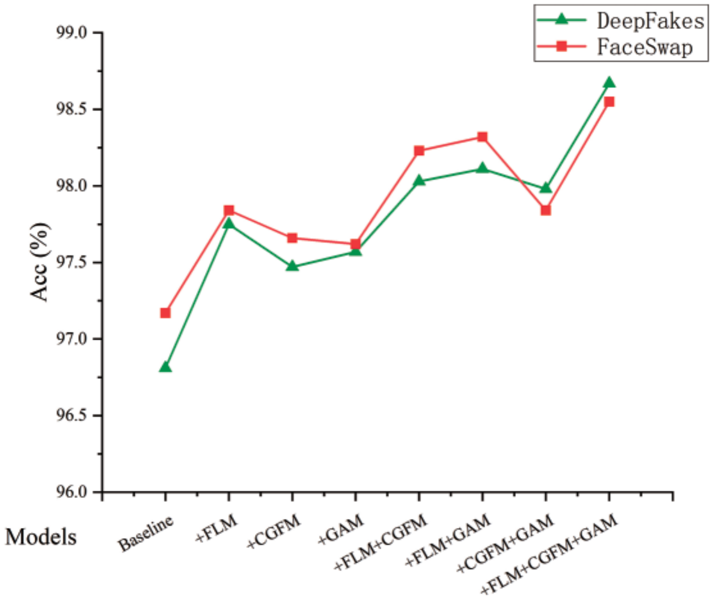

Figure 9: The performance gains brought by different improvement strategies on two datasets

Based on Xception, when the FLM is used only, the detection accuracy of the model is improved by 0.94% and 0.67% on the two datasets, respectively, proving FLM's effectiveness. After embedding the FLM in the baseline model, the Xception is transformed into a multi-branch structure, allowing the model to focus on the subtle discrepancies among different regions of the real and fake images so that more potential feature information can be mined, providing more possible bases for subsequent classification.

When the FLM and CGFM are used together, the detection accuracy of the model is improved by 0.28% and 0.39%, respectively, compared with using the FLM alone. Furthermore, the AUC and loss are also optimized. It indicates that the model's performance can be further improved by using the CGFM and FLM together. The reason is that the submodule FCM in CGFM can effectively explore the complementary information between the features of two branches and enhance the regional features in each branch through feature fusion guided by the complementary information to make them more diverse and discriminative.

After introducing the GAM, compared with the baseline model, using the GAM alone leads to 0.76% and 0.45% improvement in detection accuracy on the two datasets, respectively. Compared with using FLM and CGFM, the simultaneous introduction of the three improvement strategies leads to 0.64% and 0.32% improvement in model detection accuracy, respectively. The experimental comparison results show that the global attention module enables the model to better represent critical information through the global cross-dimensional interaction. At the same time, it further demonstrates that the combination of the three improvement strategies can obtain the best detection results, which verifies the effectiveness of the proposed method in deepfake detection in this paper.

4.5.2 The Ablation Experiment of Global Attention Module Embedding

To explore the best embedding position of the global attention module in the multi-branch model, this paper designs seven types of attention module embedding methods and inserts them into the

Blocks of different branches, respectively. Acc and AUC are used for the experimental indexes. The specific embedding methods and detection results are shown in Table 4 and Fig. 10.

Table 4: The classification effect performance gain generated by each embedding method

Method	Embedding position			DeepFakes		FaceSwap	
	First branch	Second branch	Third branch	Acc/%	AUC/%	Acc/%	AUC/%
(1)	—	—	—	98.03	99.61	98.23	99.64
(2)	√	—	—	97.18	98.87	97.13	98.45
(3)	—	√	—	**98.67**	**99.80**	**98.55**	**99.72**
(4)	—	—	√	98.13	99.60	98.02	99.58
(5)	√	√	—	97.35	99.52	97.58	99.51
(6)	√	—	√	97.63	99.58	97.67	99.62
(7)	—	√	√	97.90	99.67	97.82	99.58
(8)	√	√	√	97.29	99.50	97.32	99.48

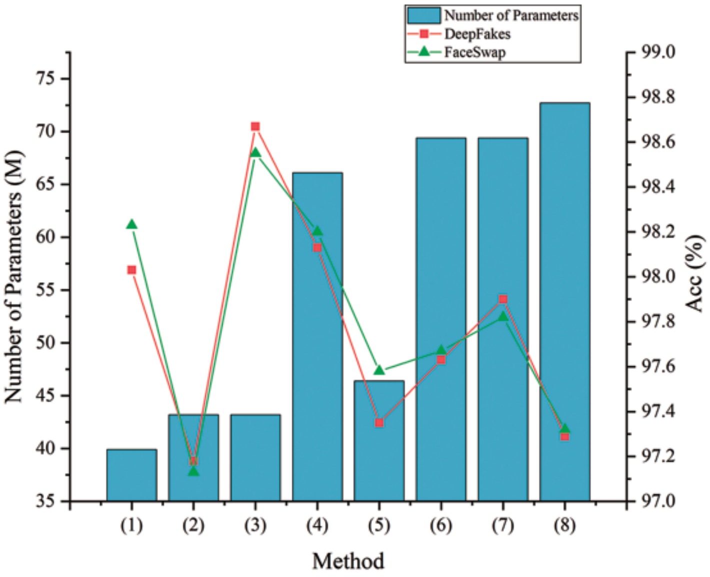

Figure 10: The classification effect and the number of model parameter change on two datasets

According to the experimental setting, the results in the table can be classified as unembedded GAM, single-branch embedded GAM, and multi-branch embedded GAM. Method (1) does not embed GAM. Methods (2) (3) (4) embed GAM in different branches, respectively. Methods (5) (6) (7) (8) belong to multi-branch embedded GAM.

The model achieved the highest Acc and AUC values on both datasets when the GAM is embedded in the second branch only. Compared with method (1), embedding GAM in the second branch increased the Acc values by 0.61% and 0.64%, and embedding GAM in the third branch increased the Acc values by 0.2% and 0.39%. However, there is a slight decrease in the model detection accuracy after embedding the GAM in the first branch. In the first branch, the model still mines the global features of the sample, which contains both forgery information and a lot of background information. Therefore, embedding the GAM in the first branch tends to make the model focus on irrelevant background information, which causes bias in detection.

At the same time, compared with the case of method (1), the model detection accuracy in the case of embedding GAM in multiple branches does not improve significantly with the increase of attention module numbers. Therefore, considering model detection accuracy and the number of parameters brought by the introduction of the attention modules, in this paper, we choose to insert the GAM in the Block of the second branch.

4.5.3 The Comparative Experiment of Different Attention Modules

To further verify the effectiveness of GAM for model performance improvement in the deepfake detection task, we replace the GAM embedded in the backbone network with the SE [29], ECA [30], BAM [31], CBAM [32], and CA [33]. Meanwhile, the improved model with the FLM and CGFM is defined as the baseline to compare the detection performance of the model after embedding different attention modules on two kinds of datasets. Experimental metrics use Acc and AUC. The experimental results are shown in Table 5 and Fig. 11.

According to the comparison results in Table 5, it can be found that the model performance does not improve significantly and even decreases after embedding the SE and the ECA in the branches, respectively. The analysis of the reason for this phenomenon may be related to the lack of spatial information. In extracting channel attention weights, the feature map's spatial dimension is compressed, which makes it easy to ignore the spatial information in the process of feature fusion among channels, leading to a decrease in the model detection accuracy. In addition, although the CA injects the position information into the channel attention, it focuses mainly on the channel information of images and cannot capture spatially long-distance dependencies.

When using the BAM and CBAM, the final results are still unsatisfactory despite the slight improvement in model performance compared with the baseline model. Due to the lack of global cross-dimensional interaction of spatial and channel information, the network cannot obtain a more accurate feature representation capability.

Table 5: The comparison results of different attention modules

Attention module	DeepFakes		FaceSwap	
	Acc/%	AUC/%	Acc/%	AUC/%
Baseline	98.03	99.61	98.23	99.64
SE [29]	96.84	98.82	97.55	99.40
ECA [30]	97.39	99.55	97.79	99.56
BAM [31]	98.13	99.41	97.91	99.62
CBAM [32]	98.32	99.61	98.47	99.68
CA [33]	97.87	99.58	97.59	99.54
GAM [3]	**98.67**	**99.80**	**98.55**	**99.72**

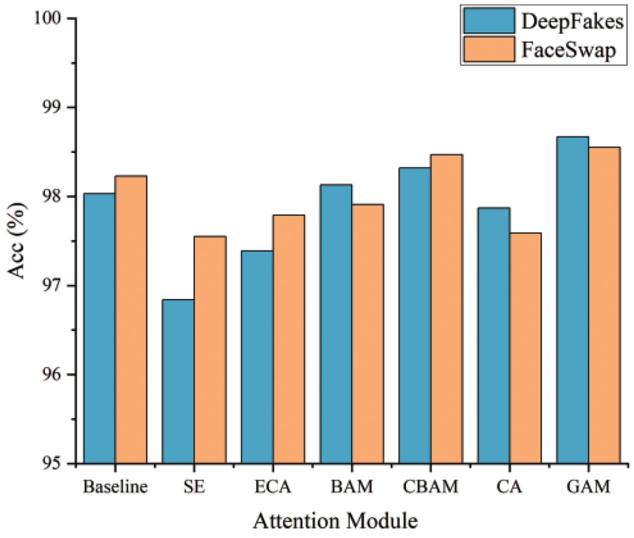

Figure 11: The classification effect of using different attention modules on two datasets

After embedding the GAM, the detection performance of the model is significantly improved, indicating that compared with other mainstream attention modules, the GAM provides more effective attention guidance in deepfake detection tasks, which is conducive to improving model detection accuracy.

4.5.4 The Comparative Experiment with Other Algorithms

To improve the effectiveness of the model in this paper for the deepfake detection task, nine mainstream deepfake detection models are selected for comparison in this experiment, and the evaluation metrics include Acc and AUC. The comparison results are shown in Table 6 and Fig. 12.

Table 6: Comparison with other models

Model	DeepFakes		FaceSwap		Celeb-DF-v2	
	Acc/%	AUC/%	Acc/%	AUC/%	Acc/%	AUC/%
ResNet50 [8]	94.50	95.67	94.83	94.67	92.87	93.43
Xception [9]	96.81	98.74	97.17	99.10	95.82	97.41
EfficientNet [10]	96.63	99.18	97.12	99.13	96.33	97.73
MesoNet [11]	93.40	93.67	91.70	93.40	91.21	91.81
Capsule network [12]	96.50	97.67	95.93	97.31	92.77	92.51
Identity encoding [13]	94.50	96.21	84.50	90.30	91.22	94.34
Muti-attention [15]	97.60	99.29	96.37	98.97	96.90	98.10
Defakehop [17]	97.45	95.95	98.28	97.87	90.56	87.65
Block shuffling [18]	95.30	98.68	97.33	98.13	98.26	98.72
Ours	**98.67**	**99.80**	**98.55**	**99.72**	**98.37**	**99.58**

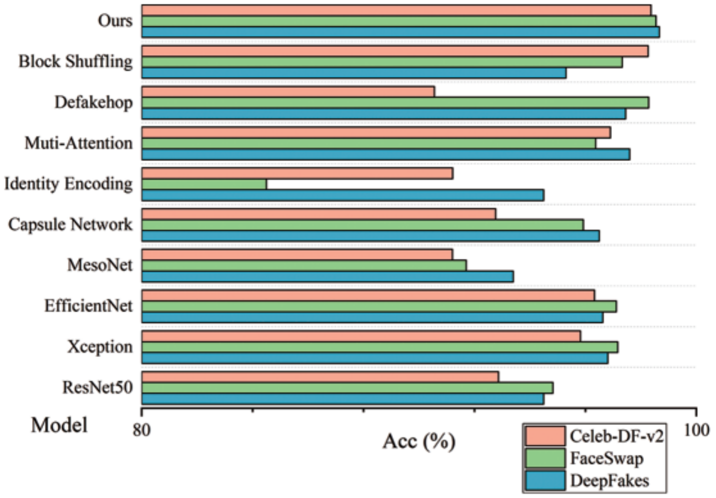

Figure 12: Comparison with other models

Based on the comparison results, it can be seen that the proposed algorithm in this paper has significant improvements in detection performance compared with the mainstream detection models. On the DeepFakes and FaceSwap datasets, the proposed algorithm achieves the best results regarding Acc and AUC values, with 98.67% and 98.55% for Acc and 99.80% and 99.72% for AUC, respectively. Meanwhile, to test the ability of the algorithm to deal with samples with more complex forgery degree, the Celeb-DF-v2 dataset with higher forgery quality is selected experimentally, and the Acc value of 98.37% and the AUC value of 99.58% are obtained, which have better improvements in the detection accuracy and effect than most current algorithms.

In summary, the detection algorithm proposed in this paper is more accurate in detecting forged samples, which is mainly due to the multi-branch structure that enables the model to mine the fine-grained feature information in different regions of the samples and fuses the features among different branches through the correlation-guided fusion module, to learn more discriminative fine-grained features.

4.5.5 Visualization of Results

To represent the gain effect of the improvement module on the multi-branch network more intuitively, we use the Grad-CAM graph [42] to visualize the regions of the detection model in different branches focusing on samples, as shown in Fig. 13.

The Grad-CAM is a deep network visualization method based on gradient localization. Specifically, the activation map is obtained as a cumulative weighted average of activation values across channel dimensions given a feature map. The red part in Fig. 8 represents the region of interest for the model. For the forged samples chosen from the three datasets in the experiment, the activation maps in the first column to the third column correspond to the attention maps drawn from different branches of the model, respectively, and the three rows represent the changes in the activation maps in the case of different improvement methods.

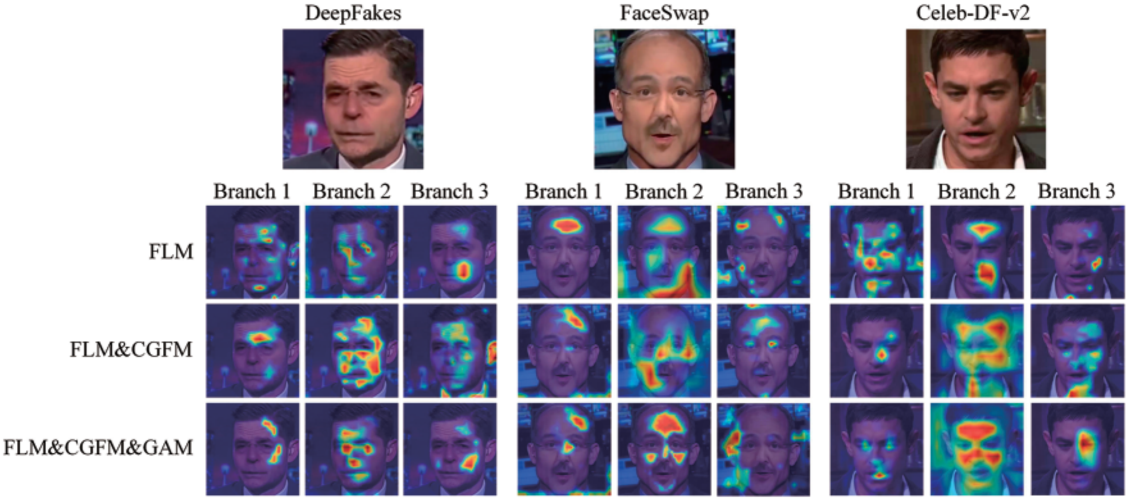

Figure 13: Visualization of the forged region

By comparing the discrepancies among the proposed different improvement methods in the class activation attention graph, it can be found that: (1) After the introduction of FLM, the model can locate different regions of the sample and starts to mine feature information from these fine-grained regions. (2) After the introduction of CGFM, the feature information learned in all branches is focused on the diverse regional information mined by different branches. (3) After the introduction of GAM, different weights are assigned to different channels and regions, improving the feature map's multi-scale feature representation capability. The visualization experiments demonstrate the contribution of FLM, CGFM, and GAM to the model in the deepfake detection task, which gives the model better detection capability.

5 Conclusions

In this paper, we express the deepfake detection task as the fine-grained image classification problem and propose a multi-branch deepfake detection algorithm based on fine-grained features. The algorithm constructs a multi-branch network structure that can focus on the subtleties of different sample regions to learn multi-scale detailed features, which effectively solves the shortage of detection accuracy due to coarse-grained features. We introduce the feature localization module and correlation-guided fusion module to complete the forgery traces' localization and detection. In addition, embedding the global attention module in the backbone network enhances the cross-dimensional interaction of spatial and channel information and reduces the influence of irrelevant background regions in the sample. These improved strategies provide a new idea for deepfake detection.

In terms of detection accuracy, the proposed algorithm in this paper investigates the effectiveness of the improved strategies through a variety of ablation experiments, which has been confirmed to be more comprehensive and accurate in the detection task.

In future work, we can continue to conduct research from the following two aspects: (1) First, we should explore more efficient fine-grained feature extraction methods. (2) Second, we research how to construct richer deepfake samples to improve the cross-library testing capability of fine-grained features for deepfake detection.

Acknowledgement: All the face images used in this paper are from publicly available deepfake datasets, including FaceForensics++ and Celeb-DF-V2.

Funding Statement: This work was supported by the 2023 Open Project of Key Laboratory of Ministry of Public Security for Artificial Intelligence Security (RGZNAQ-2304) and the Fundamental Research Funds for the Central Universities of PPSUC (2023JKF01ZK08).

Author Contributions: Study conception and design, W. K. Qin and T. L. Lu; methodology, W. K. Qin; validation, L. Zhang, S. F. Peng and D. Wan; formal analysis, W. K. Qin and T. L. Lu; investigation, W. K. Qin and S. F. Peng; data collection, W. K. Qin and T. L. Lu; draft manuscript preparation, W. K. Qin; writing review and editing, W. K. Qin and L. Zhang; visualization, W. K. Qin and S. F. Peng; supervision, T. L. Lu and L. Zhang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this paper can be requested from the corresponding author upon request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley *et al.,* “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[2] I. Korshunova, W. Shi, J. Dambre and L. Theis, “Fast face-swap using convolutional neural networks,” in *Proc. of the 2021 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Venice, Italy, pp. 22–29, 2017.

[3] F. Juefei-Xu, R. Wang, Y. H. Huang, Q. Guo, L. Ma *et al.,* “Countering malicious DeepFakes: Survey, battleground, and horizon,” *International Journal of Computer Vision*, vol. 130, no. 7, pp. 1678–1734, 2022.

[4] X. Wang, H. Guo, S. Hu, M. C. Chang and S. W. Lyu, “GAN-generated faces detection: A survey and new perspectives,” *arXiv preprint arXiv:2202.07145*, 2022.

[5] B. J. Chen, X. Liu, Y. H. Zheng, G. Y. Zhao and Y. Q. Shi, “A robust GAN generated face detection method based on dual-color spaces and an improved Xception,” in *Proc. of the 2021 IEEE/CVF Transactions on Circuits and Systems for Video Technology(TCSVT)*, Beijing, vol. 32, pp. 3527–3538, 2022.

[6] H. D. Li, B. Li, S. Q. Tan and J. W. Huang, “Identification of deep network generated images using disparities in color components,” *Signal Processing*, vol. 174, no. 4, pp. 107616–107642, 2022.

[7] Y. Fu, T. F. Sun, X. H. Jiang, K. Xu and P. S. He, “Robust GANs face detection based on dual-channel CNN network,” in *Proc. of the 2019 IEEE/CVF Int. Congress on Image and Signal Processing(ICISP)*, Suzhou, China, pp. 1–5, 2019.

[8] H. Guo, S. Hu, X. Wang, M. C. Chang and S. W. Lyu, “Eyes tell all: Irregular pupil shapes reveal GAN generated faces,” *arXiv preprint arXiv:2109.00162*, 2022.

[9] Y. C. Liu, Z. R. Shao and N. Hoffmann, “Global attention mechanism: Retain information to enhance channel-spatial interactions,” *Computer Systems Science and Engineering*, vol. 43, no. 3, pp. 967–984, 2022.

[10] S. Agarwal, H. Farid, Y. M. Gu, M. M. He, K. Nagano *et al.,* “Protecting world leaders against deep fakes,” in *Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 38–45, 2019.

[11] O. de Lima, S. Franklin, S. Basu, B. Karwoski and A. George, “Deepfake detection using spatiotemporal convolutional networks,” *arXiv preprint arXiv:2006.14749*, 2020.

[12] D. Cozzolino, A. Rössler, J. J. Thies, M. Nießner and L. Verdoliva, "ID-reveal: Identity-aware deepfake video detection," in *Proc. of the 2021 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Montreal, QC, Canada, pp. 15108–15117, 2021.

[13] Z. K. Sun, Y. J. Han, Z. Y. Hua, N. Ruan and W. J. Jia, "Improving the efficiency and robustness of DeepFakes detection through precise geometric features," in *Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, pp. 3609–3618, 2021.

[14] K. M. He, X. Y. Zhang, S. Q. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the 2016 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770–778, 2016.

[15] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 1251–1258, 2017.

[16] M. X. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. of the 2019 PMLR Int. Conf. on Machine Learning (ICML)*, Long Beach, CA, USA, pp. 6105–6114, 2019.

[17] D. Afchar, V. Nozick, J. Yamagishi and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *Proc. of the 2018 IEEE Int. Workshop on Information Forensics and Security (WIFS)*, Hong Kong, China, pp. 1–7, 2018.

[18] H. H. Nguyen, J. Yamagishi and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in *Proc. of the 2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, pp. 2307–2311, 2019.

[19] Y. Nirkin, L. Wolf, Y. Keller and T. Hassner, "Deepfake detection based on discrepancies between faces and their context," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6111–6121, 2021.

[20] Z. H. Shang, H. T. Xie, Z. J. Zha, L. Y. Yu, Y. Li *et al.,* "PRRNet: Pixel-region relation network for face forgery detection," *Pattern Recognition*, vol. 116, no. 4, pp. 107950, 2021.

[21] H. Q. Zhao, W. B. Zhou, D. D. Chen, T. Y. Wei, W. M. Zhang *et al.,* "Multi-attentional deepfake detection," in *Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 2185–2194, 2021.

[22] L. Guarnera, O. Giudice and S. Battiato, "DeepFake detection by analyzing convolutional traces," in *Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 666–667, 2020.

[23] H. S. Chen, M. Rouhsedaghat, H. Ghani, S. W. Hu, S. Y. You *et al.,* "Defakehop: A light-weight high-performance deepfake detector," in *Proc. of the 2021 IEEE Int. Conf. on Multimedia and Expo (ICME)*, Kunming, China, pp. 1–6, 2021.

[24] S. T. Liu, Z. C. Lian, S. Q. Gu and L. Xiao, "Block shuffling learning for deepfake detection," *arXiv preprint arXiv:2202.02819*, 2022.

[25] S. Hu, Y. Z. Li and S. W. Lyu, "Exposing GAN-generated faces using inconsistent corneal specular highlights," in *Proc. of the 2021 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Ontario, Canada, pp. 2500–2504, 2021.

[26] H. Guo, S. Hu, X. Wang, M. C. Chang and S. W. Lyu, "Eyes tell all: Irregular pupil shapes reveal GAN-generated faces," in *Proc. of the 2022 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, pp. 2904–2908, 2022.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017. https://doi.org/10.48550/arXiv.1706.03762

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. H. Zhai *et al.,* "An image is worth 16 × 16 words: Transformers for image recognition at scale," in *Proc. of the 2021 Int. Conf. on Learning Representations (ICLR)*, Wuhan, China, pp. 4092–4114, 2021.

[29] R. Y. Du, D. L. Chang, A. K. Bhunia, J. Y. Xie, Z. Y. Ma *et al.,* "Fine-grained visual classification via progressive multi-granularity training of jigsaw patches," in *Proc. of the 2020 Springer European Conf. on Computer Vision (ECCV)*, Glasgow, USA, pp. 153–168, 2020.

[30] X. S. Wei, Y. Z. Song, O. Mac Aodha, J. X. Wu, Y. X. Peng *et al.,* "Fine-grained image analysis with deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 8927–8948, 2021.

[31] J. L. Fu, H. L. Zheng and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proc. of the 2017 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 4476–4484, 2017.

[32] T. Hu, H. G. Qi, Q. M. Huang and Y. Lu, "See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification," *arXiv preprint arXiv:1901.09891*, 2021.

[33] F. Zhang, M. Li, G. S. Zhai and Y. Z. Liu, "Multi-branch and multi-scale attention learning for fine-grained visual categorization," in *Proc. of the 2021 27th Int. Conf. on MultiMedia Modeling (ICMMM)*, Prague, Czech Republic, pp. 136–147, 2021.

[34] J. W. Song and R. Y. Yang, "Feature boosting, suppression, and diversification for fine-grained visual classification," in *Proc. of the 2021 IEEE Int. Joint Conf. on Neural Networks (IJCNN)*, Shenzhen, China, pp. 1–8, 2021.

[35] J. Hu, L. Shen and G. Sun, "Squeeze-and-excitation networks," in *Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, CA, USA, pp. 7132–7141, 2018.

[36] Q. L. Wang, B. G. Wu, P. F. Zhu, P. H. Li, W. M. Zuo *et al.,* "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 11534–11542, 2020.

[37] J. Park, S. Woo, J. Lee and I. Kweon, "Bam: Bottleneck attention module," *arXiv preprint arXiv:1807.06514*, 2018.

[38] S. Woo, J. Park, J. Lee and I. Kweon, "Cbam: Convolutional block attention module," in *Proc. of the 2018 Springer European Conf. on Computer Vision (ECCV)*, Munich, Germany, pp. 3–19, 2018.

[39] Q. B. Hou, D. Q. Zhou and J. S. Feng, "Coordinate attention for efficient mobile network design," in *Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, pp. 13713–13722, 2021.

[40] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies *et al.,* "FaceForensics++: Learning to detect manipulated facial images," in *Proc. of the 2019 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 1–11, 2019.

[41] Y. Li, X. Yang, P. Sun, H. Qi and S. Lyu, "Celeb-DF: A large-scale challenging dataset for deepfake forensics," in *Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 3204–3213, 2020.

[42] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh *et al.,* "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. of the 2017 IEEE Int. Conf. on Computer Vision (ICCV)*, Venice, Italy, pp. 618–626, 2017.