FBI-Net: Frequency-Based Image Forgery Localization via Multitask Learning With Self-Attention

A-ROM GU, JU-HYEON NAM, (Graduate Student Member, IEEE), AND SANG-CHUL LEE®, (Member, IEEE)
Department of Computer Science and Engineering, Inha University, Incheon 22211, South Korea
Corresponding author: Sang-Chul Lee (scllee@inha.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) under Grant NRF-2021R1A2C2010893, and in part by Inha University Research Grant.

ABSTRACT Image forgery is easily manufactured for illegal acts such as spreading misleading information, which can have unfortunate consequences for society. In this work, we propose a Discrete Cosine Transformation (DCT) based multi-task learning network named FBI-Net, for forgery localization. Our proposed network adopts a fully convolutional encoder-decoder architecture, consisting of three encoders sharing parameters, a bridge attention module, and two output streams in the decoder. The encoder takes three inputs: RGB images and high-/low- DCT-filtered images. High-frequency components help learn object characteristics that improve CNN accuracy; low-frequency components are essential frequency information to keep most of the energy found in the typical DCT. Subsequently, Dilated Frequency Self-Attention Module, DFSAM in the bridge layer, is incorporated into the network to recalibrate the fused features and enhance the representation. Finally, in the decoder stage, region and edge information of the label are learned through multi-task learning to provide more extensive supervision for forged region localization; the edge stream will give a deeper understanding of features between forged and authentic images and help learn how to predict exquisite representations in images. Simultaneously, the auxiliary features from the pre-trained segmentation model are fused to separate the segmented background and objects, drawing the segmentation result of the dense region obtained. Extensive experiments show that our proposed FBI-Net outperforms existing forgery localization methods on six benchmark splicing and copy-move image datasets, CASIA TIDE v 1.0, CASIA TIDE v 2.0, Carvalho, Columbia, Coverage, IMD2020, achieving the best performance in an average of IoU of 70.99% and F1-score of 76.98% which is 9.79%, 9.82% higher than the previous method, respectively.

INDEX TERMS Deep learning, forgery detection, frequency domain, self-attention, localization.

I. INTRODUCTION

Over the past few years, rapid advances in graphics editing technology have enabled attackers to manipulate, forge, and tamper with multimedia to make it invisible to the human visual system. The forged images have become more epidemic and sophisticated. Especially splicing and copy-move are the most widely used techniques, which process images in almost imperceptible ways to the human eyes. Meanwhile, the malicious spread of fake media created in those ways will cause severe damage to our society, including fake news propagation, transaction fraud, and invasion of individual privacy. Therefore, an effective image forgery detection method for preventing the malicious propagation of forged digital images is a significant problem in modern society.

Most forgery detection research has developed traditional algorithms to detect different features between forgery and authentic multimedia. Traditional forgery detection use hand-crafted features that require prior knowledge, including JPEG compression artifacts [5], local pattern analysis [13], noise variance estimation [19], and Steganalysis [20]. However, since prior knowledge is not always available in the real world, these detection methods cannot be generalized to other datasets. Therefore, recent studies use detection
methods using deep neural networks (DNN) to exclude prior knowledge and generalize a broad range of datasets.

Numerous studies have shown that DNNs can effectively infer qualitative dependencies from high-dimensional inputs [16]. In particular, deep convolutional neural networks (CNNs) with numerous parameters and strong representation power provide excellent generalization ability for computer vision tasks. Due to the high generalization ability and success of CNNs in various tasks and benchmark datasets, recent researchers have been interested in DL-based forgery detection techniques. These methods extract forged traces in the feature space and achieve high accuracy in public datasets [25].

Recently, there have been increasing attempts to combine frequency domain and deep learning (DL) to solve advanced computer vision tasks such as image classification, super-resolution, and forgery detection [6]. In particular, forgery detection with frequency domain tracks manipulation traces by analyzing the frequency of forged images. Some studies use high-pass filters, Gabor filters, and other methods to extract relevant features from high-frequency components to correct and uncover hidden subtleties and detect forgery based on our understanding of forged image features. This is a challenging task due to the limited ability to identify fakes visually. [27] performed a successful detection methodology in frequency distribution shapes to capture frequency domain elements that are not detected in the RGB domain. Nevertheless, state-of-the-art detection methods still suffer from problems and lack of generalization due to high-quality fake images and various forgery methods. Since the other characteristics of copy-move forgery detection (CMFD) and splicing datasets disparity, they are rarely addressed concurrently [36]. Therefore, developing a comprehensive detection method for various high-quality fake images is overly essential and challenging.

Our paper notes that various state-of-the-art methods do not work well for CMFD and Splicing datasets. To address the above problems, we propose a novel deep encoder-decoder network that incorporates the correlational information between frequency-decomposed and RGB images. Our model, FBI-Net, consists of the following main parts: frequency decomposition, a pre-trained parameter sharing encoder, our proposed Dilated Frequency Self-Attention Module, and a segmentation representative auxiliary feature. The image is first decomposed to frequency components using Discrete Cosine Transform (DCT) and filtered out to low and high frequency. Then they are passed into three pre-trained parameter sharing encoders. The RGB features from the shared encoder are skip-connected into a decoder in an up-sampling layer. The self-attention module DFSAM enhances the features extracted from the encoder. In addition, a pre-trained segmentation model is used to extract auxiliary object features, and the features are fused into a decoder to assist in detailed localization. Finally, multi-task learning with the region and edge stream helps utilize more extended supervision for the localization of forged regions.

The rest of this paper consists of the following. Section II introduces contemporary forgery detection techniques, the combination of frequency domain and DL, and research trends in self-attention mechanisms. Section III describes the data preprocessing techniques and the overall structure of FBI-Net. In particular, Section III-C describes the main algorithm of the dilated frequency self-attention module. Section IV describes the CASIA TIDE v2.0, CASIA TIDE v1.0, IMD2020, COLUMBIA, IMD2020, CARVALHO, and COVERAGE datasets used in this paper, the experimental setup, and network architecture in detail. Section V compares and evaluates FBI-Net with other methods on various datasets. The experimental results in Section V verify that FBI-Net has superior generalization ability to other methods. Section V-B performs ablation studies on the model architecture and hyperparameters.

Our main contributions are summarized as follows:

- We propose a novel DL-based model, FBI-Net, for forgery detection with an encoder-decoder model capable of end-to-end learning. DCT filtering is used to extract frequency information, and then DFSAM combines them to emphasize the traces of the forged region.
- We successfully incorporated a pre-trained segmentation model to guide the location of specific forged objects during decoding. The attention between the previous decoder’s feature maps, skip connections, and segmentation features are applied.
- MTL is applied to accurately predict the shape of the forged region by simultaneously predicting the edge and region. An ablation study has demonstrated the efficiency of each component of FBI-Net. Compared to the state-of-the-art networks on six benchmark datasets, we obtain the highest performance, and our model detects forged images, including splicing and copy-move.

II. RELATED WORKS

FBI-Net is a network that transforms the input image into the frequency domain through DCT to extract the forgery traces contained in the spatial domain and localize the forged region of the input image. In this section, we describe traditional techniques and modern CNN-based approaches to detect the forged region of spliced or copy-move images. Furthermore, we describe various attempts to combine DL with the frequency domain and self-attention to improve the object recognition abilities of CNNs.

A. FORGERY DETECTION

Several approaches have been developed to identify image forgery. We will go through several of them and their drawbacks below.

1) TRADITIONAL METHODS

Color Filter Array (CFA) is a feature that detects forged regions due to differences in color processing methods.
between digital cameras. [26] proposed an algorithm to detect duplicated regions by performing the principal component analysis (PCA) on small patches of a single digital image. However, the detection rate varies sensitively depending on the size of the image block and the JPEG quality. [13] use discontinuities introduced by forged regions during CFA interpolation. Meanwhile, CFA-based detection techniques only detect the forged regions made with different camera models. [4] detected duplicated moving images of identical objects through a feature extraction algorithm, but these algorithms are also unable to detect forged regions if the image is from the same camera model. Error Level Analysis (ELA) is a forensic technique that utilizes different compression levels to analyze images. This method is used to identify images that have been digitally modified. Since our method is based on DL, we did not compare it with the algorithm-based approach. In this paper, we propose a CNN-based network that can be used on various datasets beyond the limitations of forgery detection by traditional techniques that require prior knowledge of the dataset.

2) DEEP LEARNING METHODS
Modern DL-based forgery detection techniques require no prior knowledge of datasets and have a high generalization ability to various datasets. In particular, [9] showed that the learning stability of residual-based DL [15] and its success in various tasks provide opportunities to utilize DL. Inspired by the success of [9] respectively propose a CNN with a fixed (SRM filter) or trainable single-layer high-pass filter. [35] proposed a high-level operation of a CNN consisting of two streams with features obtained through SRM filters. However, [30] demonstrated that the limited layers of [28] and [2] are suitable for small-scale networks and datasets. [20] suggests that their forensic approach for identifying camera models based on CNN feature extraction may be used to detect splicing since it can estimate the source of picture fragments.

B. THE FREQUENCY DOMAIN WITH DEEP LEARNING
Recently, various studies introducing the frequency domain into CNNs have proposed new perspectives for DL. [31] demonstrated that high-frequency components have object characteristics that improve CNN accuracy, unlike human visual systems. Inspired by the JPEG compression process, [22] propose a method to train and represent CNNs in the frequency domain. In particular, [14] trained a CNN for image classification using the DCT components of each image patch. At the same time, research on forgery tracking in the frequency domain is being actively carried out. In particular, [27] identifies facial forgery by extracting information in
TABLE 1. Notation table.

| Notation | Description |
|----------|-------------|
| $X_{\text{full}}$ | input RGB domain image |
| $X_{\text{low}}$, $X_{\text{high}}$ | low-/high-frequency decomposed image |
| $\mathcal{D}$ | Discrete Cosine Transformation function |
| $\mathcal{D}^{-1}$ | Inverse Discrete Cosine Transformation function |
| $\hat{X}$ | DCT transformed image |
| $I_{\text{low}}$, $I_{\text{high}}$ | indicator function for mask generation |
| $\alpha$ | hyperparameter for mask generation |
| $\odot$ | element-wise multiplication |
| $\mathcal{S}_i$, $\mathcal{O}_i$ | encoder blocks, $i = 1, 2, 3$ |
| $X_{\text{full}}$, $X_{\text{low}}$, $X_{\text{high}}$ | output of each encoder block, $i = 1, 2, 3$ |
| Q, K, V | concatenated features of last encoder output |
| A | self-attention map |
| $\odot$ | matrix multiplication |
| O | output of DFSAM |
| $\gamma$ | 2D dilated convolutional filter |
| $\sigma$ | dilation rate |
| $d_i$ | decoder blocks, $i = 1, 2, 3$ |
| $P_i$ | output of decoder block, $i = 1, 2, 3$ |
| S | auxiliary segmentation model |
| $S_i$ | auxiliary segmentation feature |
| R | region prediction |
| E | edge prediction |

the frequency domain and examining statistical features using DCT and DFT, respectively. F3-Net [27] obtained SOTA performance on heavily compressed videos, but its cross-dataset assessment performance suffers substantially.

C. SELF-ATTENTION MECHANISM

[21] emphasizes the importance of attention mechanisms in the process of human visual systems. Self-attention leads to global dependencies on input sequences and reliably performs interpretation of long sentences. This self-attention is used in natural language processing and computer vision, including image classification, object detection, instance segmentation, and Generative Adversarial Networks.

III. PROPOSED METHOD

FBI-Net is a convolutional encoder-decoder network with an end-to-end learning architecture. We present a novel MTL network that simultaneously learns mask regions and mask edges with our proposed module dilated frequency self-attention module (DFSAM). The notations and the algorithm of the proposed method are shown in Table 1 and Algorithm 1, respectively.

A. FREQUENCY DECOMPOSITION

In this paper, we design a network to classify forged and authentic image regions with low-/high-frequency components. To decompose the frequency of the input image, we use a 2D DCT transformation function with a computational complexity of $O(N^2 \log_2 N)$. We define $x^{\text{full}} \in \mathbb{R}^{H \times W \times 3}$ as input image where $H$ and $W$ denote height and width of the input image $x^{\text{full}}$. First, we transform RGB domain input to the DCT domain.

$$\hat{x} = \mathcal{D}(x^{\text{full}}),$$

where $\hat{x} \in \mathbb{R}^{H \times W \times 3}$, and $\mathcal{D}$ denote DCT transformed image and 2D DCT transformation function, respectively.

In order to extract the high-frequency components, the low-frequency region of $\hat{x}$ is filtered. We note that high-frequency components help to emphasize the distortion contained in $x^{\text{full}}$. In addition, the low-frequency components are essential frequency information to keep most of the energy found in the typical components. We use these two types of information to exploit image distortion and entire content. It is expected that high-quality information can be learned more effectively than when using a single RGB image as an input image. These assumptions can be confirmed in Table 4. To filter out high-/low-frequency components, we define indicator function $I_{\text{low}}$, $I_{\text{high}} \in \mathbb{R}^{H \times W}$ as follows:

$$I_{\text{low}}(i, j) = \begin{cases} 1 & \text{if } 0 \leq i \leq W - \alpha \text{ and } 0 \leq j \leq H - \alpha, \\ 0 & \text{otherwise}, \end{cases}$$

$$I_{\text{high}} = 1 - I_{\text{low}},$$

where $\alpha$ denote a hyperparameter that determines the range of low-frequency.

The binary mask separates a square matrix’s upper-left and lower-right regions, where $\alpha$ values start at the (0,0) axis in...
the upper-left corner of the mask, forming a complete isosceles triangle. Taking a low-pass filter mask as an example, when \( \alpha \) values are \( H \) and \( W \), a binary filter with a value of 0 in the lower-right corner and 1 in the upper-left corner are created. Also, the high-pass filter mask equals the subtraction of 1 of the low-pass filter mask (See Figure 3). \( \alpha \) related ablation studies and imaging data can be found in Table 4 and Figure 5.

Now, \( x^i \) is decomposed by element-wise multiplication of binary mask \( I^i_{\text{low}} \) and \( I^i_{\text{high}} \). We get \( x^i \), \( x^i_{\text{high}} \in \mathbb{R}^{H \times W \times 3} \) by applying inverse Discrete Cosine Transform (IDCT) as follow:

\[
\begin{align*}
    x^i_{\text{low}} &= D^{-1}(I^i_{\text{low}} \otimes \hat{x}) \quad (4) \\
    x^i_{\text{high}} &= D^{-1}(I^i_{\text{high}} \otimes \hat{x}) \quad (5)
\end{align*}
\]

where \( \otimes \), and \( D^{-1} \) denote element-wise multiplication, and IDCT, respectively.

**B. SHARED ENCODER**

We employ U-Net as the encoder-decoder network with ResNet18, pre-trained with ImageNet as the backbone network. To retain the more dense feature maps and design an efficient network, we remove the last downsampling operation of ResNet. This allows the resolution of the output feature map to be increased to 1/8 of the input image.

Each encoder block \( e_i \) consists of a convolutional layer with a kernel size of \( 3 \times 3 \), the stride of 1, padding of 1, with a batch normalization layer and a ReLU activation function. Each block \( e_i \) shares parameters with three inputs \( x^i_{\text{low}} \), \( x^i_{\text{high}} \), \( x^i_{\text{full}} \), and outputs \( x^i_{\text{low}} \), \( x^i_{\text{high}} \), \( x^i_{\text{full}} \) where \( i = 1, 2, 3 \).

\[
x^i_z = e(x^i_{z-1}) \quad (6)
\]

where \( x^0_z = x^i \) for \( z = \text{low}, \text{high}, \text{or full} \).

By learning images of different frequency types from a single encoder, the encoder learns rich representations in terms of various ranges of inputs. It is shown in Table 4 that the configuration of the corresponding network is adequate.

**C. DILATED FREQUENCY SELF ATTENTION MODULE**

During the encoding phase of the input image, the last layer contains abstract and high-dimensional information; then, the encoded information is passed to an attention-based bridge layer. The RGB information \( x^i_{\text{full}} \) helps to identify abnormal textures in fake images, and the compressed frequency information \( x^i_{\text{low}} \) and \( x^i_{\text{high}} \) improve subtle bias errors. In this paper, we propose a Dilated Frequency Self-Attention Module (DFSAM), which extracts an attention map and highlights essential information.

The structure of DFSAM module is illustrated in Figure 2. To produce a self-attention map \( A \), we first concatenate three output feature maps, \( X = [x^i_{\text{low}}, x^i_{\text{high}}, x^i_{\text{full}}] \in \mathbb{R}^{H' \times W' \times 3C'} \) and reduce channel by applying convolution layer with kernel size of \( 1 \times 1 \) into \( X' \in \mathbb{R}^{H' \times W' \times C'} \). After that, we feed \( X' \) into three individual single convolution layer with kernel size of \( 3 \times 3 \) to generate three new features \( Q, K, \) and \( V \). Then, we reshape them into \( \mathbb{R}^{C' \times N} \) where \( N = H' \times W' \) is the number of pixels in each feature maps. We perform a matrix multiplication, denoted as \( \odot \), between \( Q \) and the transpose of \( K \) and apply softmax operation to calculate a self-attention map \( A \in \mathbb{R}^{C' \times C'} \).

\[
A_{ij} = \frac{\exp \left( \max (Q_i \odot K_j) \right)}{\sum_{i=1}^{N} \exp \left( \max (Q_i \odot K_j) \right)} \quad (7)
\]

where \( A_{ij} \) measures the impact of \( i \)-th channel with respect to \( j \)-th channel. We perform a matrix multiplication between \( A \) and \( V \) to produce an attention mechanism. And then, we apply element-wise summation between \( A \) and scaled feature map to derive output \( O_j \):

\[
O_j = \sum_{i=1}^{N} (V_{ji} \odot A_i) \quad (8)
\]

We reshape \( O_j \) into \( O \in \mathbb{R}^{H' \times W' \times C'} \).

Multi-scale feature fusion detects forged regions at various scales by applying convolutional layers with different dilation rates to increase the field of view of filters. Since we noticed the objects of various sizes in the forged images, we applied multi-scale atrous convolution fusion to the attention feature map. Mathematically, a 2D dilated convolutional filter denoted as \( y \) is defined as follows.:

\[
y(i) = \sum_{k} x(i + rk)w(k) \quad (9)
\]

where \( x \) and \( w \) are the input signal and filter, respectively, and \( r \) is the dilation rate that determines the sampling step size of the input signal. This paper sets the dilation rate \( r = [1, 6, 12, 18] \) and concatenates the resulting feature maps for each dilation rate. Note that if \( r = 1 \), it is the same as the standard convolution operation.

**D. MULTI-TASK LEARNING DECODER**

By sharing representations throughout numerous related tasks, multi-task learning (MTL) can improve the learning efficiency of deep networks. The decoder further processes the feature map, which results in the final pixel-by-pixel prediction.

The decoder block \( d_i \), shown in Figure 1, sequence starts at the bottom and ends with the last decoder block that generates the output for \( i = 1, 2, 3 \). Decoder blocks \( d_i \) are composed of a convolution layer with a kernel size of \( 3 \times 3 \), a stride of 1, padding of 1, a ReLU activation function, nearest neighborhood upsampling layer with rate 2, followed by dropout with a probability of 0.5. However, if feature maps from the previous convolution block are upsampled, the forgery traces extracted by the convolution operation can disappear. To compensate for the loss of feature information caused by upsampling, we perform the skip connections on feature maps of the same spatial size extracted from the corresponding stage encoder block with the decoder block. A residual block (RB) is applied before skip-connecting the feature map from
Algorithm 1 FBI-Net Forward Procedure.

Require: Forged image $x^{\text{full}}$, Low-frequency range parameter $\alpha$

Apply DCT on forged image, $x^d \leftarrow \mathcal{D}(x^{\text{full}})$
Define filter masks, $I_{\text{low}}$, and $I_{\text{high}}$

\[
x_{\text{low}}, x_{\text{high}} \leftarrow \mathcal{D}^{-1}(I_{\text{low}} \otimes \hat{x}), \mathcal{D}^{-1}(I_{\text{high}} \otimes \hat{x})
\]

for $i = 1, 2, 3$ do

for $z = \text{low}, \text{high}, \text{full}$ do

\[
x^z_i \leftarrow e_i(x^z_{i-1}) \quad \triangleright \quad x^z_0 = x^d
\]

$P_i \leftarrow \text{DFSAM}\left(x^z_3, x^z_2, x^z_1\right)$

\[
\text{for } i = 1, 2, 3 \quad s_i \leftarrow \text{Conv}($S(x^{\text{full}})$)
\]

$P_i \leftarrow d_i($FAM$\left(RB(x^{\text{full}}_{i-1}), P_{i-1}, s_i\right))$

$\hat{R} \leftarrow C_{3 \times 3}(P_3; w_{\hat{R}})$

$\hat{E} \leftarrow C_{3 \times 3}(P_3; w_{\hat{E}})$

return Region Prediction $\hat{R}$, Edge Prediction $\hat{E}$

the encoder block to the decoder block. RB is a simplified pre-activation residual block, which removes the batch normalization layer.

The pre-trained segmentation model is designed to assist in training the decoder. The forged image contains a mix of fake and authentic objects. The pre-trained segmentation model trained with a general dataset tends to extract all objects as an auxiliary feature that supports detecting forged objects, but there is a limitation that it tends to detect authentic objects as forged regions, even if they are not forged (See Figure 6). FBI-Net identifies the fake object based on the obtained object location and checks the fake background that the pre-trained segmentation model cannot find. The pre-trained segmentation model trained with a subset of COCO train2017, 21 categories present in the Pascal VOC dataset by DeepLabV3+ with Resnet50 as a backbone. The auxiliary feature from the pre-trained segmentation model denoted as $S$ aims to extract the location information of an object from forged and avoid losing object boundary information during decoding. It helps to locate objects more accurately in the final prediction than other methods (See Column 9 in Figure 5). Through the auxiliary feature containing information about the segmented background and objects, the segmentation result of the dense area can be obtained. After the features have been concatenated, the feature attention module (FAM) emphasizes the channels, which is a module composed of a batch normalization layer, ReLU activation function, and convolution layer with a kernel size of $3 \times 3$ a sigmoid function, bringing utilization feature inter-channel relationships. Channel attention focuses on what is meaningful given an input image since each channel of a feature map is considered a feature detector.

\[
P_i = d_i($FAM$([RB(x^{\text{full}}_{i-1}), P_{i-1}, s_i]))
\]

where $P_{i-1}$ denotes output from the bridge or previous decoder according to the stage of the decoder block. In this case, $s_i$ includes location information of objects in the forged image $x^{\text{full}}$. Before being input to each decoder block, the resolution of the feature map must be reduced for concatenation with other features maps $P_{i-1}$, and $x^{\text{full}}_{i-1}$. In this paper, since the pooling operation distorts the location information of objects, we apply convolution layers with kernel size/stride of 4/4, 2/2, and 1/1 to reduce the resolution of auxiliary segmentation features. Finally, the two outputs, region and edge prediction are derived from two individual $3 \times 3$ convolutional layers. When an image is forged, manual cropping is applied, and the boundaries of a forged image have properties that distinguish them from authentic images. Learning this feature with edge stream will give a deeper understanding of features between forged and authentic images and help learn how to predict exquisite representations in images. Layer disposal details are provided in Table 2.

TABLE 2. Layer disposal details of the FBI-Net. The bridge and decoder details are shown above. Encoder architecture information can be found on [15]. The input/output sizes are in the channel priority shape. Each element of Operation represents (kernel size, stride, and padding). Note that the MTL region and MTL Edge receive features separately from the decoder, and the outputs also proceed separately. Upsstands for nearest interpolation upsampling. The activation function is the nonlinearization function applied after the convolution operation ($\text{BN} = $ Batch Normalization, ReLU = Rectified Linear Unit activation function).

\[
\begin{array}{cccccc}
\text{Layer} & \text{Input Size} & \text{Output Size} & \text{Operation} & \text{Activation} \\
\text{Bridge} & \text{Conv} (256, 32, 32) & (256, 32, 32) & (3, 1, 1) & \text{BN}, \text{ReLU} \\
\text{Decoder} & \text{Conv} (528, 32, 32) & (128, 32, 32) & (3, 1, 1) & \text{ReLU} \\
& \text{Up} (128, 32, 32) & (128, 64, 64) & \text{ratio} = 2 & \text{Null} \\
& \text{Conv} (277, 64, 64) & (64, 64, 64) & (3, 1, 1) & \text{ReLU} \\
& \text{Up} (64, 64, 64) & (64, 128, 128) & \text{ratio} = 2 & \text{Null} \\
& \text{Conv} (149, 128, 128) & (32, 128, 128) & (3, 1, 1) & \text{ReLU} \\
& \text{Up} (32, 128, 128) & (32, 256, 256) & \text{ratio} = 2 & \text{Null} \\
\text{MTL Region} & \text{Conv} (32, 256, 256) & (1, 256, 256) & (3, 1, 1) & \text{Null} \\
\text{MTL Edge} & \text{Conv} (32, 256, 256) & (1, 256, 256) & (3, 1, 1) & \text{Null} \\
\end{array}
\]

where $C_{3 \times 3}$ is a convolution layer with kernel size of $3 \times 3$, stride of 1, and padding of 1. Since the region and edge prediction results from an independent stream, there are two individual convolution layers in the last decoder block. So, $w_{\hat{R}}$ and $w_{\hat{E}}$ are region stream, and edge stream parameter, respectively.

E. LOSS FUNCTION

This section describes a loss function designed for multi-task learning for FBI-Net. Our proposed loss function is constructed for label masks and mask edges predictions.

Whereas forged regions frequently appear as tiny parts. Therefore, the model mainly predicts negative regions due to the severe class imbalance, which can hamper our model accuracy big time. Since this imbalance problem makes it difficult to transmit meaningful information, it decreases learning efficiency. We note that this imbalance problem is exacerbated when predicting edges; hence we employ focal
loss (FL) to reweight the loss of negative and positive samples to tackle the class imbalance problem. Loss of concentration while training reduces the weight of simple classes while increasing attention to challenging classes. We employ FL for the region and edge loss, which is expressed by

\[
L_{FL} = - \sum_{i=1}^{W} \sum_{j=1}^{H} [\alpha(1 - \hat{Y}_{i,j})^Y Y_{i,j} \log(\hat{Y}_{i,j}) + (1 - \alpha)\hat{Y}_{i,j}^Y (1 - Y_{i,j}) \log(1 - \hat{Y}_{i,j})],
\]

where \( Y = [Y_{i,j}] \) and \( \hat{Y} = [\hat{Y}_{i,j}] \) are ground truths, and predictions of mask edge, respectively. \( \alpha \) is a parameter used to balance the positive and negative regions; \( \gamma \) is used to balance the easy and hard classified samples. We set \( \alpha = 0.25, \gamma = 2 \) in our experiments. Edge labels are obtained by sobel gradient operation. \( Z(Y) = Sobel(Y) \) where \( Z(Y) \) signifies the label’s edge.

We employ binary cross-entropy (BCE) loss for region loss, which is expressed by

\[
L_{BCE} = - \sum_{i=1}^{W} \sum_{j=1}^{H} [\hat{Y}_{i,j} \log(Y_{i,j}) + (1 - \hat{Y}_{i,j}) \log(1 - Y_{i,j})]
\]

Dice Loss significantly reduces foreground-background imbalance by optimizing the network based on the dice overlap coefficient between the predicted segmentation result and the correct answer annotation. We employ Dice loss for the region and edge loss, which is expressed by

\[
L_{Dice} = 2 \frac{\sum_{j=1}^{W} \sum_{i=1}^{H} Y_{i,j} \hat{Y}_{i,j}}{\sum_{j=1}^{W} \sum_{i=1}^{H} Y_{i,j}^2 + \sum_{j=1}^{W} \sum_{i=1}^{H} \hat{Y}_{i,j}^2}
\]

Finally, the total loss of the multi-task FBI-Net for forgery localization can be written as:

\[
L_{total} = \lambda_1 L_{region} + \lambda_2 L_{edge}
\]

where \( L_{region} = L_{BCE} + L_{Dice} \) and \( L_{edge} = L_{FL} + L_{Dice} \). The objective of FBI-Net training is to minimize \( L_{total} \), which is equivalent to minimizing \( L_{region} \) and \( L_{edge} \), respectively.

In our experiments, we set \( \lambda_1 = \lambda_2 = 1 \), so that forgery detection ability of our network is not significantly affected by \( \lambda_1 \) and \( \lambda_2 \).

### IV. IMPLEMENTATION DETAILS

#### A. DATASET DESCRIPTION

We evaluate the performance of our proposed method on the following six public forged datasets. The number of images in the dataset is provided in Table 3.

- **CASIA TIDE** [12] CASIA TIDE v1.0 and v2.0 have 461, and 5,105 forged images, respectively, both includes Copy-Move and Splicing images. Also, a popular dataset for image forgery localization includes images from several sources.
- **Carvalho** [10] has 100 Splicing images, DSO-1 contains images of people. Forgeries were created by adding one or more individuals from one image to another with post-processing to increase photorealism.
- **Columbia** [23] has 180 splicing images, Columbia is a historic dataset for forgery detection. Ground truth masks are obtained by distinguishing between authentic and forged images, followed by some post-processing.
- **IMD2020** [24] has 2010 splicing images. It contains both copy-move and splicing images, includes real-life forgery images, and manually created ground truth masks.

For convenience, the images of all datasets are resized to 256 x 256. We split each dataset into 80%, 10%, and 10% of training, validation, and test dataset, respectively.

#### B. EXPERIMENTAL SETUP

Our experiment is performed on Ubuntu 18.04 with TITAN RTX 2080Ti GPU. A multi-task FBI-Net for detecting and locating traces of image forgery has been implemented in Python 3.8 using the DL framework Pytorch 1.8.0.

We employed the Adabound optimizer [18] with weight decay of 0.01, \( \beta_1 \) of 0.9, and \( \beta_2 \) of 0.999 for network training. We set the loss function propagation ratio \( \gamma_1 \) and \( \gamma_2 \) of pre-trained encoder and decoder as same, that is, \( \gamma_1 = \gamma_2 = 1 \). The initial learning rate starts at \( 10^{-3} \) and is reduced to \( 10^{-6} \) at the last epoch with Cosine Annealing Scheduler.

We fix batch size to 32 and train the network for 100 epochs.

We employed data augmentation CutMix [34] with a probability of 0.1. The parameter for mask generation, \( \alpha \), is fixed to 100.

For the quantitative performance comparison of each network, we choose the following three metrics.

- **Intersection over Union (IoU):** IoU is used to measure the intersection between the ground truth and prediction, divided by their union
- **F1-score:** F1-score measures the overlap between the ground truth and the segmented predictions
- **Matthews correlation coefficient (MCC):** MCC is defined by considering all categories of the confusion matrix.

We re-implemented DCUNet, and SENet according to their paper. Besides, the released code of FCN8s, UNet, SegNet, DeepLabV3+, and RRUNet are also used for training.

### TABLE 3. The forgery localization datasets used in our experiments.

| Datasets           | No. of Images |
|--------------------|---------------|
|                    | copy-move | Splicing | Total |
| CASIA TIDE v 1.0   | 459       | 461      | 920   |
| CASIA TIDE v 2.0   | 3274      | 1840     | 5123  |
| Carvalho           | -         | 100      | 100   |
| Columbia           | -         | 180      | 180   |
| Coverage           | 100       | -        | 100   |
| IMD2020            | -         | 2010     | 2010  |
The loss function is set without Dice loss, of training the network with Dice Loss. In this experiment, This experiment is performed to demonstrate the efficiency compared to those of the compared methods. Our proposed FBI-Net can accurately localize almost all the splicing and copy-move images. It out- detailed localization. Our proposed FBI-Net can accurately localize all of the forged regions well but still lacks images but wrongly localizes copy-move images as noises. DCUNet successfully detects splicing small regions of the spliced regions and copy-move images but is not well localized. RRU-Net detect rough regions in splicing images but localize unnecessarily image textures in copy-move images. RRU-Net and Mantranet localize unnecessary regions. Unet and SegNet+ are not well localized. FCN8s, DeepLabV3+ are shown in Figure 4. We can see that FCN8s, DeepLabV3+, and Mantranet localize unnecessary regions. Unet and SegNet detect rough regions in splicing images but localize unnecessary image textures in copy-move images. RRU-Net detects the spliced regions quite well on the large regions, but small regions of the spliced regions and copy-move images are not well localized. DCUNet successfully detects splicing images but wrongly localizes copy-move images as noises. SENet detects all of the forged regions well but still lacks detailed localization. Our proposed FBI-Net can accurately localize almost all the splicing and copy-move images. It outperforms all the compared methods.

### B. ABLATION STUDY

We validate the effectiveness of each component in our proposed network. The splicing localization results are then compared to those of the compared methods.

### C. DICE LOSS

This experiment is performed to demonstrate the efficiency of training the network with Dice Loss. In this experiment, the loss function is set without Dice loss, $L_{\text{Dice}}$ and $L_{\text{FL}}$. The quantitative results are shown in $7^{\text{th}}$ row in Table 4. Achieving IoU, F1-score of 69.39%, 75.69% which is 2.15%, 1.61% lower than our proposed method.

### D. DATA AUGMENTATION

This experiment is performed to demonstrate the efficiency of applying data augmentation with CutMix. In this experiment, the network is trained without Data Augmentation. The quantitative results are shown in $5^{\text{th}}$ row in Table 4. Achieving IoU, F1-score of 69.44%, 74.78% which is 2.1%, 2.52% lower than our proposed method. The qualitative results are shown in $3^{\text{rd}}$ column in Figure 5. It can be observed that the forgery region is not detected at all. The only noise that is completely irrelevant is detected. The reason is that images are arbitrarily synthesized with data augmentation, which helps in learning by alleviating data imbalance. This result proves that the Data Augmentation helps network training when class is imbalanced.

We also experiment with the changes of the parameters, $\lambda$ deciding the size of the blocks generated by CutMix. Shown in Table 5. We sample the bounding box coordinates $B = (rx, ry, rw, rh)$ to indicate the cropping regions on $x_A$ and $x_B$ before sampling the binary mask $M$. The patch clipped from $B$ of $x_B$ is used to fill up the region $B$ in $x_A$. We sample rectangular masks $M$ whose aspect ratio is proportionate to the original picture. The box coordinates are sampled evenly following:

$$
rx \sim \text{Uniform}(0, W), \quad rw = W \sqrt{1 - \lambda},
$$

$$
ry \sim \text{Uniform}(0, H), \quad rh = H \sqrt{1 - \lambda},
$$

where, $\text{Uniform}$ is a uniform distribution, making cropped area ratio $\frac{rw \times rh}{W \times H} = 1 - \lambda$. As a result of the ablation study, we investigate a tendency where $\lambda$ value increases, the network accuracy increases simultaneously by enlarged spliced regions, causing to learn more class balanced environment.

### E. PARAMETER SHARING

This experiment is performed to demonstrate the efficiency of parameter sharing in our proposed method. In this experiment, the network is trained without parameter sharing in the encoder with individual encoders for each inputs, $x_{\text{full}}$, $x_{\text{low}}$, $x_{\text{high}}$. The quantitative results are shown in $6^{\text{th}}$ row in Table 4. Achieving IoU, F1-score of 69.70%, 75.01% which is 1.84%, 2.29% lower than our proposed method.

The qualitative results are shown in $2^{\text{nd}}$ column in Figure 5. It can be observed that it captures the forged region somewhat but only detects a large area. The reason is that the dice loss considers class imbalance while learning. This result proves that adopting Dice loss is effective in training localization tasks.

### V. EXPERIMENTAL RESULT

In this section, first, the forgery localization results of our proposed method are compared with the results of the compared methods. Then the efficiency of each component of our proposed network. The performance of our proposed network is evaluated by calculating the IoU, F1-score, and MCC. Our proposed method is compared with existing forgery localization methods introduced in Table 7, 8. It can be observed that our proposed method achieves an average IoU, F1-score, MCC of 61.20%, 67.16%, 55.90%, which is 9.79%, 1.61%, 2.15% lower than our proposed method. In this experiment, the network is trained without parameter sharing in our proposed method. In this exper-

### TABLE 4. Performance comparison of our proposed method with ablation study in terms of IoU, F1-Score, MCC. Bold means 1st performance.

| Methods               | IoU  | F1-Score | MCC  |
|-----------------------|------|----------|------|
| w/o DCT decomposition | 0.6978 | 0.7496   | 0.5334 |
| Auxiliary Feature     | 0.5836 | 0.6454   | 0.3353 |
| DPSAM                 | 0.6904 | 0.7427   | 0.5216 |
| MTL edge              | 0.6818 | 0.7381   | 0.5136 |
| Data Augmentation     | 0.6944 | 0.7478   | 0.5136 |

### TABLE 5. According to CutMix size sampling ratio $\lambda$, the performance comparison of the proposed method in terms of IoU, F1-score, and MCC. Bold means 1st performance.

| Parameter   | IoU  | F1-Score | MCC  |
|-------------|------|----------|------|
| 0.0001 - 0.0001 | 0.7074 | 0.7641   | 0.5610 |
| 0.001 - 0.0001  | 0.7075 | 0.7619   | 0.5606 |
| 0.01 - 0.001    | 0.7104 | 0.7619   | 0.5572 |
| Ours           | 0.7154 | 0.7730   | 0.5714 |
FIGURE 4. Localization maps of different compared methods. The first row is the forged images; the second to the tenth rows represent the localization maps obtained by FCN8s, Unet, SegNet, DeepLabV3+, RRU-Net, Mantranet, DCUNet, SENet, and our proposed method, respectively; the last row is the corresponding ground-truth mask of the first row.
The qualitative results are shown in 5th column in Figure 5. It can be observed that there are traces of detecting the outer edge of the synthesized area, but it cannot be detected as a complete form as a whole. This result proves that parameter sharing can learn to assimilate features well.

**F. DCT DECOMPOSITION**

This experiment is performed to demonstrate the efficiency of our proposed three-encoder model structure. In this experiment, the encoder is reduced by the number of one, and the input of the encoder is only an RGB domain image, $x_{full}$ without frequency decomposed images, $x_{low}$, $x_{high}$. The quantitative results are shown in the first row in Table 4. Achieving IoU, F1-score of 69.78%, 74.96% which is 1.76%, 2.34% lower than our proposed method. The qualitative results are shown in the sixth column in Figure 5. It can be observed that it detects the vicinity of the forged regions but only shows a recognizable rough shape. This result proves that the frequency-decomposed input image effectively detects forgery while using three encoders.

**G. MTL EDGE**

This experiment is performed to demonstrate the efficiency of Multi-task Learning. In this experiment, the network is trained without Edge stream and $L_{edge}$ as well. The quantitative results are shown in 4th row in Table 4. Achieving IoU, F1-score of 68.18%, 73.81% which is 3.36%, 3.49% lower than our proposed method. The qualitative results are shown in 7th column in Figure 5. It can be observed that it intends to detect the traces of the synthesized area, but it is still insufficient. This result proves that Multi-task learning with an edge guide helps to learn subtle regions.

**H. DFSAM**

This experiment is performed to demonstrate the efficiency of our proposed self-attention method, DFSAM. In this experiment, the network is trained without DFSAM in the bridge, $d_i$ and just applying simple concatenation followed by $1 \times 1$ convolution layer for matching the dimension.
TABLE 7. Performance comparison of our proposed method with baseline methods in terms of IoU, F1-Score, MCC. Bold means 1st performance.

| Methods     | CASIA1 [12] | CASIA2 [12] | COLUMBIA [23] |
|-------------|-------------|-------------|---------------|
|             | IoU         | F1-Score    | MCC           | IoU          | F1-Score    | MCC           | IoU          | F1-Score    | MCC           |
| FCNs [17]   | 0.4516      | 0.4888      | 0.0758        | 0.4811       | 0.5063      | 0.0677        | 0.3769       | 0.4294      | -0.0014       |
| Unet [29]   | 0.5110      | 0.5674      | 0.1953        | 0.5197       | 0.5496      | 0.1464        | 0.7688       | 0.8289      | 0.6972        |
| SegNet [1]  | 0.5268      | 0.5860      | 0.2346        | 0.4775       | 0.4976      | 0.0425        | 0.7947       | 0.8515      | 0.7272        |
| DeepLabV3+ [7] | 0.4342  | 0.5137      | 0.0653        | 0.4824       | 0.5012      | 0.0491        | 0.4483       | 0.5613      | 0.1504        |
| RRU-Net [3] | 0.5736      | 0.6467      | 0.3487        | 0.4957       | 0.5225      | 0.0907        | 0.8015       | 0.8663      | 0.7440        |
| Mantranet [33] | 0.4688     | 0.5203      | 0.1051        | 0.4608       | 0.4952      | 0.0301        | 0.3912       | 0.4565      | 0.0729        |
| DCUNet [11] | 0.3023      | 0.5666      | 0.2163        | 0.6746       | 0.7135      | 0.4728        | 0.7900       | 0.8596      | 0.7472        |
| SENet [36]  | 0.6732      | 0.7360      | 0.5103        | 0.6801       | 0.7245      | 0.4997        | 0.7030       | 0.7689      | 0.5935        |
| Ours        | 0.7154      | 0.7730      | 0.5714        | 0.6923       | 0.7388      | 0.5052        | 0.9311       | 0.9634      | 0.9171        |

TABLE 8. Performance comparison of our proposed method with baseline methods in terms of IoU, F1-Score, MCC. Bold means 1st performance.

| Methods     | IMD2020 [24] | CARVALHO [10] | COVERAGE [32] |
|-------------|--------------|----------------|---------------|
|             | IoU          | F1-Score       | MCC           | IoU          | F1-Score       | MCC           | IoU          | F1-Score       | MCC           |
| FCNs [17]   | 0.4807       | 0.5048        | 0.0672        | 0.4358       | 0.4829        | 0.0401        | 0.4542       | 0.5099        | 0.0291        |
| Unet [29]   | 0.5234       | 0.5658        | 0.1779        | 0.5642       | 0.6351        | 0.3091        | 0.3120       | 0.5630        | 0.1586        |
| SegNet [1]  | 0.5285       | 0.5716        | 0.1855        | 0.4321       | 0.4672        | 0.0290        | 0.4936       | 0.4830        | 0.0204        |
| DeepLabV3+ [7] | 0.5166   | 0.5546        | 0.1383        | 0.4284       | 0.4603        | 0.0000        | 0.4966       | 0.1722        | 0.0027        |
| RRU-Net [3] | 0.5328       | 0.5806        | 0.2105        | 0.5791       | 0.6364        | 0.3004        | 0.4929       | 0.5593        | 0.1370        |
| Mantranet [33] | 0.4610     | 0.5003        | 0.0332        | 0.4295       | 0.4838        | 0.0220        | 0.4610       | 0.5003        | 0.0352        |
| DCUNet [11] | 0.5367      | 0.5883        | 0.2247        | 0.5367       | 0.6203        | 0.2749        | 0.4924       | 0.5419        | 0.1172        |
| SENet [36]  | 0.5508       | 0.5956        | 0.2373        | 0.5394       | 0.6109        | 0.2628        | 0.5256       | 0.5937        | 0.2174        |
| Ours        | 0.6204       | 0.6827        | 0.4066        | 0.6853       | 0.7700        | 0.5521        | 0.6151       | 0.6910        | 0.4018        |

FIGURE 7. Example images of failure cases. The first row is the forged images, the second row represents the corresponding ground-truth mask, and the third row represents the corresponding failure case of the prediction.

The quantitative results are shown in 3rd row in Table 4. Achieving IoU, F1-score of 69.04%, 74.27% which is 2.5%, 3.03% lower than our proposed method. The quantitative results are shown in 8th column in Figure 5. It can be observed that it detects a completely different area except for one example. This result proves that our proposed self-attention method, DFSAM enhances feature learning for the network.

I. AUXILIARY FEATURE

This experiment is performed to demonstrate the efficiency of adding auxiliary features, $s_i$ from the pre-trained segmentation model for training. In this experiment, the auxiliary features are excluded from the decoder. The quantitative results are shown in 2nd row in Table 4. Achieving IoU, F1-score of 58.36%, 64.54% which is 13.18%, 12.76% lower than our proposed method. The qualitative results are shown in 9th column in Figure 5. It can be observed that almost no forged regions are detected. This result proves that our proposed self-attention method, DFSAM enhances feature learning for the network.

J. BACKBONE

In this experiment, we explain why we use ResNet-18 [15] as the backbone of FBI-Net. As shown in Table 6, ResNet-18 not only achieves the best performance but also has the lowest computation complexity so that it can be used in practical applications. Too complex networks are more prone to over-fitting due to a large number of parameters.

VI. CONCLUSION

In this paper, a novel multi-task FBI-Net consisting of DFSAM has been proposed for forgery localization. First, a DCT decomposition is applied to the input image to generate various frequency information. Second, our proposed self-attention method, DFSAM, helps to intensify the assorted features extracted from the feature extractor. Third, the pre-trained segmentation model helps extract auxiliary features to assist the information of the segmented background and objects. Finally, multi-task learning with the region and edge labels can supply abundant supervision on the forged region. Ablation studies have confirmed the efficiency of each network component. Existing methods are specialized for CMFD or splicing datasets. The findings from the ablation study of DCT decomposition are that even though the human visual system cannot detect the forgery, a subtle change in frequency remains. Therefore, the frequency domain can provide valuable clues to detecting forged images.

Our method has high generality compared to other methods as it shows high performance on all six datasets, including CMFD and splicing datasets.

Although our proposed method outperforms existing forgery localization methods, it has certain limitations. Because the purpose of our study is to detect a general forged...
region, the filter size was empirically derived considering the average synthesized region of the dataset. However, when a wide area is forged, the filter size is relatively small compared to the forged region, so it can be seen that the limitation is revealed because the feature cannot be captured effectively (See Figure 7). In a future study, we will apply various filter sizes to one image to transmit information of various frequencies so that forged images of the more diverse region can be detected together.

REFERENCES

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[2] B. Bayar and M. C. Stamm, “A deep learning approach to universal image manipulation detection using a new convolutional layer,” in Proc. 4th ACM Workshop Inf. Hiding Multimedia Secur., Jun. 2016, pp. 5–10.

[3] X. Bi, Y. Wei, B. Xiao, and W. Li, “RRU-Net: The ringed residual U-Net for image splicing forgery detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2019, pp. 30–39.

[4] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei, “Image copy-move forgery detection based on SURF,” in Proc. Int. Conf. Multimedia Inf. Netw. Secur., Nov. 2010, pp. 889–892.

[5] C. Chen, Y. Q. Shi, and W. Su, “A machine learning based scheme for double JPEG compression detection,” in Proc. 19th Int. Conf. Pattern Recognit., Dec. 2008, pp. 1–4.

[6] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich, “JPEG-phase-aware convolutional neural network for steganalysis of JPEG images,” in Proc. 5th ACM Workshop Inf. Hiding Multimedia Secur., Jun. 2017, pp. 75–84.

[7] L. C. Chen, Y. Zhu, G. Papandreou, S. Schröff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 801–818.

[8] S. Chen, X. Tan, B. Wang, and X. Hu, “Reverse attention for salient object detection,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 234–250.

[9] D. Cozzolino, G. Poggi, and L. Verdoliva, “Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection,” in Proc. 5th ACM Workshop Inf. Hiding Multimedia Secur., Jun. 2017, pp. 159–164.

[10] T. J. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. Rocha, “Exposing digital image forgeries by illumination color classification,” IEEE Trans. Inf. Forensics Security, vol. 8, no. 7, pp. 1182–1194, Jul. 2013.

[11] H. Ding, L. Chen, Q. Tao, Z. Fu, L. Dong, and X. Cui, “DCU-Net: A dual-channel U-shaped network for image splicing forgery detection,” Neural Comput. Appl., pp. 1–17, Aug. 2021.

[12] J. Dong, W. Wang, and T. Tan, “CASA image tampering detection evaluation database,” in Proc. IEEE China Summit Int. Conf. Signal Inf. Process., Jul. 2013, pp. 422–426.

[13] P. Ferrara, T. Bianchi, A. D. Rosa, and A. Piva, “Image forgery localization via fine-grained analysis of CFA artifacts,” IEEE Trans. Inf. Forensics Security, vol. 7, no. 5, pp. 1566–1577, Oct. 2012.

[14] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, and J. Yosinski, “Faster neural networks straight from JPEG,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), vol. 31, Dec. 2018, pp. 3933–3944.

[15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[16] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” Neural Comput., vol. 18, no. 7, pp. 1527–1554, 2006.

[17] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 3343–3344.

[18] L. Luo, Y. Xiong, Y. Liu, and X. Sun, “Adaptive gradient methods with dynamic bound of learning rate,” 2019, arXiv:1902.09843.

[19] B. Mahdian and S. Saic, “Using noise inconsistencies for blind image forensics,” Image Vis. Comput., vol. 27, no. 10, pp. 1497–1503, Sep. 2009.

[20] O. Mayer and M. C. Stamm, “Learned forensic source similarity for unknown camera models,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2018, pp. 2012–2016.

[21] V. Minh, N. Heess, and A. Graves, “Recurrent models of visual attention,” in Proc. Adv. Neural Inform. Process. Syst., 2014, pp. 2204–2212.

[22] J.-H. Nam and S.-C. Lee, “Frequency filtering for data augmentation in X-ray image classification,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2021, pp. 81–85.

[23] T. T. Ng, S. F. Chang, and Q. Sun, “A data set of authentic and spliced image blocks,” ADVENT, Columbia Univ., New York, NY, USA, Tech. Rep. 203-2004, 2004.

[24] A. Novozamsky, B. Mahdian, and S. Saic, “IMD2020: A large-scale annotated dataset tailored for detecting manipulated images,” in Proc. IEEE Winter Appl. Comput. Vis. Workshops (WACVW), Mar. 2020, pp. 71–80.

[25] Y. Rodriguez-Ortega, D. M. Ballesteros, and D. Renza, “Copy-move forgery detection (CMFD) using deep learning for image and video forensics,” J. Image, vol. 7, no. 3, p. 59, Mar. 2021.

[26] A. C. Popescu and H. Farid, “Exposing digital forgeries by detecting duplicated image regions,” Dept. Comput. Sci., Dartmouth College, Hanover, NH, USA, Tech. Rep. TR2004-515, 2004.

[27] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, “Thinking in frequency: Face forgery detection by mining frequency-aware clues,” in Proc. Eur. Conf. Comput. Vis., Cham, Switzerland: Springer, Aug. 2020, pp. 86–103.

[28] Y. Rao and J. Ni, “A deep learning approach to detection of splicing and copy-move forgeries in images,” in Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS), Dec. 2016, pp. 1–6.

[29] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Comput-Assist. Intervent., Cham, Switzerland: Springer, Oct. 2015, pp. 234–241.

[30] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, “FaceForensics++: Learning to detect manipulated facial images,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1–11.

[31] H. Wang, X. Wu, Z. Huang, and E. F. Xing, “High-frequency component helps explain the generalization of convolutional neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 8684–8694.

[32] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, “COVERAGE—A novel database for copy-move forgery detection,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2016, pp. 161–165.

[33] Y. Wu, W. Abdalmageed, and P. Natarajan, “ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 9543–9552.

[34] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, “CutMix: Regularization strategy to train strong classifiers with localizable features,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6023–6032.

[35] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, “Learning rich features for image manipulation detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1053–1061.

[36] Y. Zhang, G. Zhu, L. Wu, S. Kwong, H. Zhang, and Y. Zhou, “Multi-task SE-network for image splicing localization,” IEEE Trans. Circuits Syst. Video Technol., early access, 27, 2021, doi: 10.1109/TCSVT.2021.3123829.