Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-17914-1

FP-Net: frequency-perception network with adversarial training for image manipulation localization

Jintong Gao[1] · Yongping Huang[2]

Received: 15 October 2022 / Revised: 8 September 2023 / Accepted: 17 December 2023

Abstract

Mining the forged regions of digitally tampered images is one of the key research tasks for visual recognition. Although there are many algorithms investigating image manipulation localization, most approaches focus only on the semantic information of the spatial domain and ignore the frequency inconsistency between authentic and tampered regions. In addition, the generality and robustness of the models are severely affected by the different noise distributions of the training and test sets. To address these issues, we propose the frequency-perception network with adversarial training for image manipulation localization. Our method not only captures representation information for boundary artifact identification in the spatial domain but also separates low and high-frequency information in the frequency domain to acquire tampered cues. Specifically, the frequency separation sensing module enriches the local sensing range by separating multi-scale frequency domain features. It accurately identifies high-frequency noise features in the manipulated region and distinguishes low-frequency information. The global frequency attention module uses multiple sampling and convolution operations to interactively learn multi-scale feature information and integrate dual-domain frequency content to identify tampered physical locations. Adversarial training is employed to construct hard training adversarial samples based on adversarial attacks to avoid interference from unevenly distributed redundant noise information. Extensive experimental results show that our proposed method performs significantly better than the mainstream approach on five common standard datasets.

Keywords Image manipulation localization · Frequency perception · Attention mechanism · Adversarial learning

✉ Yongping Huang
hyp@jlu.edu.cn

Jintong Gao
gaojt20@mails.jlu.edu.cn

[1] College of Artificial Intelligence, Jilin University, 2699 Qianjin Road, Changchun 130012, China

[2] College of Computer Science and Technology, Jilin University, 2699 Qianjin Road, Changchun 130012, China

Published online: 08 January 2024

1 Introduction

The advent of multimedia content editing technology has made it extremely difficult to distinguish between authentic and tampered regions, and unscrupulous individuals use software to easily modify important electronic images, such as nucleic acid test reports, news media, and official contract seals. This poses multiple risks to society and the public. The rapid and effective identification of areas where images have been tampered with is therefore one of the most pressing issues that need to be addressed. Common image tampering techniques include splicing (copying and pasting elements from one image to another), copy-move (copying and pasting elements from an image to other areas of the same image), and removal (removing elements from an image). Figure 1 presents authentic, tampered images and ground-truth on four standard datasets. It is apparent that the subject of the manipulation of these images is generally difficult to determine directly with the naked eye, and the tampered content is highly consistent with the background content. This exacerbates the serious challenges to which deep learning is subjected.

Image manipulation localization differs from the common semantic segmentation algorithms that deal with computer vision tasks. Whereas semantic segmentation aims to classify all pixels in an image into meaningful classes of objects based on their semantic content, image manipulation localization is more concerned with the traces left by tampering operations to determine the orientation of a single region. Effective image tampering localization cannot be achieved by using only the semantic information in the spatial domain to distinguish multiple objects. Therefore, learning inconsistencies between tampered and real regions in the frequency domain is imminent. Zhou et al. [2] proposed a dual-stream faster RGB convolutional network (RGB-N) that combined end-to-end training to identify artificially tampered boundaries in RGB images, and noise features from the filtering layer of a model rich in steganalysis to identify noise inconsistencies. RGB-stream mainly provided the distinction between object contours and image brightness, while the steganalysis-rich model (SRM) extracted a large amount of high-frequency information to enhance the perception of tam-

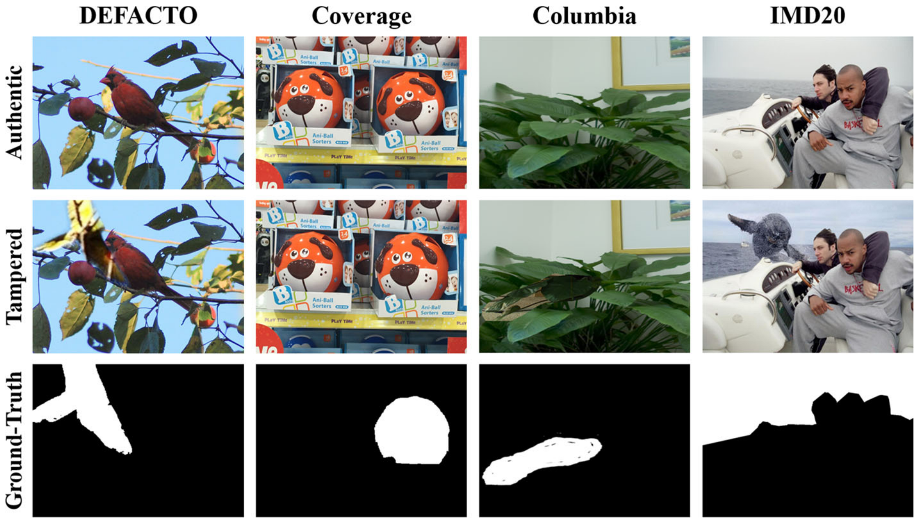


Fig. 1 Instances from four common standard image manipulation datasets. Authentic images (top) with their corresponding tampered images (medium) and ground-truth masks (bottom)

pered edges and noise. Inspired by the approach, our method proposes a frequency-separation sensing module to distinguish low and high-frequency information under multi-scale features and combines SRM to efficiently extract high-frequency noise content.

In addition, to capture the relationships between contextual information in the frequency domain, most networks use attention mechanisms to integrate the details of multi-scale features. TransForensics [11] contained a dense self-focusing encoder and a dense correction module. The former was used to model all pairwise interactions between the global context and local regions at different scales, while the latter was used to improve the transparency of the hidden layers. Our method presents a global frequency attention module based on this interaction learning idea to detect commonalities in manipulated regions from double branches and to predict finer manipulation masks.

The different noise distributions of the training and test instances can seriously affect model performance. To address this issue, models typically increase the number of training samples or perform data enhancement on the images. However, these operations undoubtedly increase the burden on academics to process the data and often have little to do with the network structure. In this paper, adversarial training is used as the primary training method. Adversarial samples adjust the amount of uniform interference added to the image according to the training process of the network structure. It reduces the reliance of the model on certain attributes and thus improves the generalization ability of the model. In summary, the main contributions of our work are described below:

1. We propose a frequency-perception network (FP-Net) that integrates features from the spatial and frequency domains at the multi-scale for image manipulation localization through adversarial training.

2. The frequency-separation sensing module (FSSM) separates the frequency information to enrich the local perceptual area. At the same time, to integrate the dual-frequency content, the global frequency attention module (GFAM) combines the high-frequency content and the low-frequency content, significantly enhancing the extraction of contextual information. Finally, adversarial learning enables the model to obtain more in-depth training to distinguish manipulated regions.

3. Extensive experimental results on five public standard datasets demonstrate that our proposed method can robustly localize the manipulated regions and significantly outperforms current state-of-the-art methods.

2 Related work

2.1 Image manipulation localization

Learning rich manipulation information semantically is an effective solution for image manipulation localization [2, 4, 11, 13, 32], which can be roughly divided into spatial domain and frequency domain. We intuitively summarize and compare FP-Net(Ours) with the state-of-the-art (SOTA) image manipulation localization methods with different techniques in recent years. The spatial domain aims to capture spatial and representation information for boundary artifact identification. For example, the Hybrid CNN-LSTM Model (J-LSTM) [1] and the Hybrid LSTM and Encoder-decoder Model (H-LSTM) [7] were proposed to joint pixel-wise classification and manipulated region segmentation from the spatial domain. The Multi-task Fully Convolutional Network (MFCN) [9] used two output branches to learn the surface labels and boundary contour lines of the stitched regions, respectively. The Spatial Pyramid Attention Network (SPAN) [16] was designed that efficiently compare patches through the local self-attention block on multiple scales. The frequency domain is another way of manipulating localization to leverage the noise features or high-frequency to discover the frequency inconsistency between authentic and tampered regions. Therefore, some recent works employed both the spatial domain and frequency domain and showed success in image manipulation localization. The Learning-rich features network (RGB-N) explored dual modalities to focus on RGB manipulation artifacts and local noise feature inconsistencies. The Manipulation Tracing Network (ManTra-Net) [8] learned the manipulation tracing feature with the long short-term network (LSTM) for both image manipulation classification and forgery localization. The Multi-View Multi-Scale Supervision(MVSS) [12] used the multi-scale edge-supervised branch and the noise-sensitive branch to learn semantic-agnostic features for manipulation detection. In addition, adversarial learning has been integrated with image manipulation localization. For example, the Generate, Segment, and Refine Network (GSR-Net) [10] proposed a manipulated image generation process for creating examples and identifying boundary artifacts with adversarial training.

Inspired by these methods, our method integrates features from the spatial and frequency domains at the multi-scale for image manipulation localization with adversarial training to enhance sensing range and identify forged cues (Table 1).

2.2 Frequency domain attention mechanism

Frequency domain attention mechanisms are widely employed in face manipulation detection. Convolution block attention module and feature fusion attention module [27] were introduced in the face forgery detection network with dual attention mechanism and feature fusion to refine and recombine these high semantic frequency features. Luo et al. [33] proposed a residual-guided spatial and frequency attention module to pay more attention to forgery traces from a new perspective. The cross-modal frequency attention module captured the correlation between two complementary modalities and facilitated learning of each of their characteristics. Inspired by these methods, the global frequency attention module (GFAM) in our method absorbs their essence while learning the features in the spatial domain and

Table 1 The comparative summary of recent methods for image manipulation detection

Method	Technique		
	Spatial Domain	Frequency Domain	Adversarial learning
J-LSTM [1] (ICCV 2017)	√		
RGB-N [2] (CVPR 2018)	√	√	
MFCN [9] (J.Vis. 2018)	√		
H-LSTM [7] (IEEE 2019)	√		
ManTra-Net [8] (CVPR 2019)	√	√	
GSR-Net [10] (AAAI 2020)	√		√
SPAN [16] (ECCV 2020)	√		
MVSS-Net [12] (IEEE 2023)	√	√	
FP-Net(Ours)	√	√	√

√ indicates that the technology was used frequency domain and consolidating their context information. It enhances the sensitivity of FP-Net to tampered images.

2.3 Adversarial learning

Adversarial Learning aims to use adversarial images to strengthen the training sample, usually based on adversarial attacks. Adversarial attacks [14, 15] reinforced the training of the model, while subtle interference provides the possibility of the robustness of the model. Therefore, we choose adversarial training to train the model rather than simply using data enhancement. This brings a new dynamic to the localization of image tampering.

3 The proposed method

In this section, we introduce a frequency-perception network with adversarial training for image manipulation localization (FP-Net). The general overview of FP-Net is shown in Fig. 2. A given tampered image or an adversarial image after an adversarial attack is fed into the ConvNeXt network [3] to obtain multi-scale feature maps F_i, $i \in \{1, 2, 3, 4\}$. The feature map F_i is then routed into the frequency-separation sensing module (in Section 3.1) to generate both high-frequency perceptual features H_i and low-frequency perceptual features L_i. The low-frequency stream merges the feature map F_i and the low-frequency feature map L_i, while the high-frequency stream merges the steganographic analysis feature map S_i after the steganalysis rich model (SRM) and the high-frequency feature map H_i. The global frequency attention mechanism (in Section 3.2) integrates contextual content of high-frequency information and low-frequency information to mine global frequency-aware forgery cues. Finally, the cross-frequency attention feature F_{output} is passed through the sigmoid function to derive the prediction mask. The whole model is enhanced by the adversarial training method to enhance the training ability of the model (in Section 3.3).

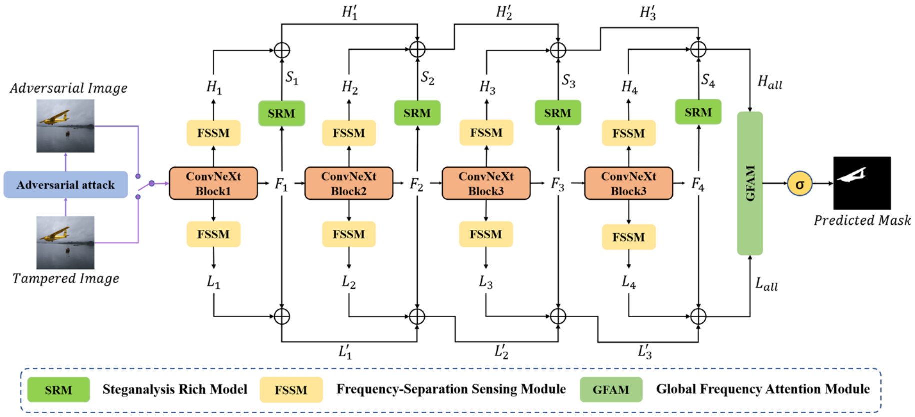


Fig. 2 Overview of the FP-Net. The proposed architecture consists primarily of frequency-separation sensing modules (FSSM), the global frequency attention module (GFAM), and adversarial training

3.1 Frequency-separation sensing module

For the complete identification of high-frequency noise, previous models have typically focused on SRM filters to sense the frequency decomposition of the image. Although SRM can extract high-frequency information, it has difficulty in covering the complete frequency domain as it can only employ fixed weights. In addition, pure spatial-domain information is not sufficient for image manipulation detection. Therefore, we propose the frequency separation sensing module (FSSM), which adaptively separates the input features in the frequency domain based on a group of learnable filters. The decomposed frequency components can in turn be applied to the spatial domain, resulting in a dual frequency-aware image component. Specifically, the FSSM consists of the Sliding Window Discrete Cosine Transform(SWDCT) and the Band Separation Filter(BSF) in Fig. 3. We describe them in detail below.

The feature map F_i, $i \in \{1, 2, 3, 4\}$ is passed into the 32×3×3 convolution layer to decrease the number of channels, thereby reducing the computational cost and facilitating frequency separation. The feature is subsequently transformed by the Sliding Window Discrete Cosine Transform (SWDCT). The computational processes of interactive frequency spectrums f_{base} are expressed as:

$$f_{base} = DCT\left(reshape\left(unfold\left(conv\left(F_i\right)\right)\right)\right), i \in \{1, 2, 3, 4\} \tag{1}$$

where DCT is the discrete cosine transform, and $reshape$ denotes the sliding window cutting operation. $unfold$ extracts sliding windows from the batched input tensor F_i. Naturally, we divide the frequency spectrum f_{base} into low-frequency and high-frequency bands to choose frequencies of interest beyond the fixed-base filters using the base filters. According to [27] applies the base filters to divide the spectrum into N bands with roughly equal energy, from low frequency to high frequency. We specify that the low-frequency band f^1_{base} is the first 1/8 and the high-frequency band f^2_{base} is the last 7/8 of the entire spectrum f_{base} with $N = 2$. These learnable filters $F_i, i \in \{1, 2, 3, 4\}$ are then added to the Band Separation Filter(BSF) to confuse these base filters and learnable filters. The process of decomposing frequency features is represented as follows:

$$H_i, L_i = split\left\{\log_{10}\left|f^j_{base} + f_w\right|\right\}, j \in \{1, 2\} \tag{2}$$

where $split$ indicates the function of the splitting feature. $\log_{10}$ is applied to balance the magnitude in the frequency band. We can acquire the $f^i_w = \frac{1-e^{F_i}}{1+e^{F_i}}$ to constrict these learnable filters F_i between -1 and 1. From the spectrum, we extract high-frequency features and low-frequency features, from which we can calculate the difference between tampered and non-tampered regions in the frequency domain.

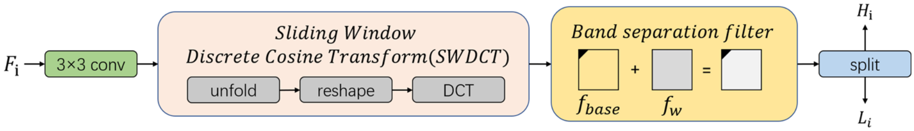


Fig. 3 The proposed frequency-separation sensing module is composed of sliding window discrete cosine transform (SWDCT) and band separation filter

3.2 Global frequency attention module

Both the low-frequency and the high-frequency stream contain a large number of tampering cues, and the global frequency attention mechanism integrates them in Fig. 4. We first use 1×1 dilated convolution and 2×2 dilated convolution to increase the field of perception, so that each convolution output contains a large range of information. Subsequently, influenced by U-Net [34], we believe that appropriate up-sampling and down-sampling will enhance the effective acquisition of contextual information and enable the use of jump-join connectivity features. Afterward, the 3×3 convolutional blocks accompanied by group normalization and rectified linear units continue to normalize the current stream features. The main reason for choosing group normalization is that the channels used to represent object features are not completely independent and there may be multiple channels representing the same feature. Therefore, within this set of feature channels, these feature values have the property of being identically distributed. In particular, it makes sense to apply group normalization to the features within this group when the frequency distribution is so clearly governed. H_{all} and L_{all} learn each other's weighting parameters via the sigmoid function. The specific calculation steps are expressed as follows:

$$L_f = \sigma\left[conv(down(conv(H_{all})) \oplus H_{all})\right] \otimes L_{all} \oplus L_{all} \tag{3}$$

$$H_f = \sigma\left[conv(down(conv(L_{all})) \oplus L_{all})\right] \otimes H_{all} \oplus H_{all} \tag{4}$$

where ⊗ and ⊕ denote dot product and addition, respectively. $conv$ represents the convolution operation with group normalization and the rectified linear unit. $down$ is the downsample, and σ is the sigmoid function. The computation of the feature fusion F_{output} can be presented as:

$$F_{output} = up(concat(H_f, L_f)) \tag{5}$$

where up is upsample, $concat$ means to concatenate two features.

The global frequency attention mechanism extracts features from dual-frequency streams and adaptively trains the model structure using their learnable weights to screen for manipu-

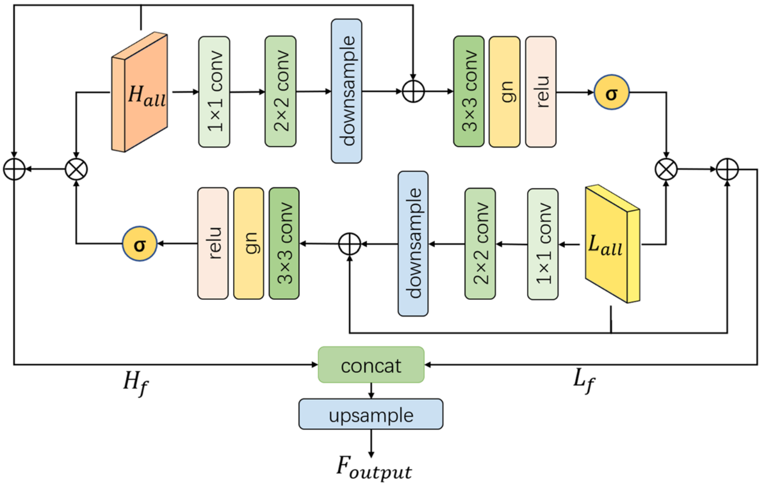


Fig. 4 The proposed global frequency attention module

lation regions. The final prediction mask is generated by a sigmoid function with a channel of 1.

3.3 Adversarial training

Data enhancement is used to improve the generalizability of models, but most of these tools (such as flipping, cropping, and panning) require a lot of manual time to process the data. If the images are processed uniformly using a program, it is not possible to select the appropriate enhancement for each image. We focus on adversarial training, which adds a moderate amount of interference I_{inter} to the input image I to generate the adversarial image I_{adv}. to force the model to learn deeper levels of feature attention. The adversarial image I_{adv} consists of the adversarial image I_{FGSM} generated by the fast gradient sign method (FGSM) and the adversarial image I_{pgd} generated by projected gradient descent (PGD). Figure 5(a) illustrates the generation of adversarial examples.

Fast Gradient Sign Method (FGSM). FGSM is a famous adversarial attack method that increases image interference by attacking the gradient of our model [14]. It is computed as follows:

$$I_{FGSM} = I + \epsilon \cdot sign(\bigtriangledown_I \mathcal{L}(\theta, I, y_{gt})) \tag{6}$$

where I denotes the source image. ϵ takes a random number in the range of (0, 0.01] at each iteration to increase randomness. $\bigtriangledown_I$ represents the model gradient obtained when the input image is I, θ denotes the model parameters, and y_{gt} is the ground-truth mask of the image. $\mathcal{L}$ denotes the loss function.

Projected Gradient Descent (PGD). FGSM is a one-time attack, adding gradients to a graph increases the gradient only once. However, when the attack model is complex, such an approach may not always succeed. We consider PGD to be a multi-step variant of FGSM, one small step at a time, with each iteration projecting the perturbation to a defined range, which is formulated as follows [15]:

$$\begin{cases} I_{PGD}^{0} = I \\ I_{PGD}^{n+1} = Clip(I_{PGD}^{n} + \alpha \cdot sign(\bigtriangledown_I \mathcal{L}(\theta, I_{PGD}^{n}, y_{gt})) \end{cases} \tag{7}$$

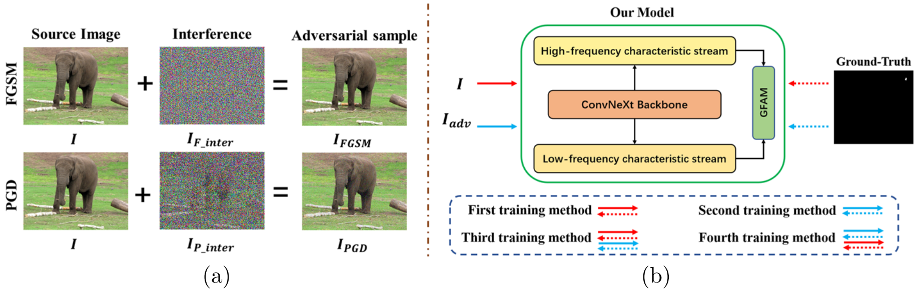


Fig. 5 Illustration of the training of our proposed FP-Net. Figure (a) displays the process of adversarial image generation. A moderate amount of interference is added to the tampered image to generate an adversarial image forcing the model to learn a deeper level of feature attention. Figure (b) shows the training process of our proposed model on four different training methods. The network trains the model with tampered images (red solid arrows) or adversarial images (blue solid arrows) and performs a loss calculation of the predicted values against the ground-truth (red and blue dashed arrows)

Table 2 The number of datasets involved in the experiment and their tampering types

Dataset	Splicing	Copy-move	Removal	Total
DEFACTO	34233	12877	17607	64717
Coverage	0	100	0	100
Columbia	180	0	0	180
NIST16	292	64	208	564
IMD20	–	–	–	2010
Wild	201	0	0	201

where $Clip$ crops the image. α takes a random number in the range of $(0, 0.01]$ at each iteration to increase randomness. Small-step attacks via PGD make it more urgent for the model to learn robust defensive capabilities to avoid misidentifying tampered regions.

In addition, to observe the effectiveness of the adversarial example, we present four training methods in Fig. 5(b). The first training method is to input a tampered source image, with the dashed line indicating the loss of the computed ground-truth and the prediction mask after passing the model. Note that the ground-truth remains constant regardless of changes in the input image. The second training method is to input an adversarial image to train the model. We also experimented with sequential training means, which are the third and fourth methods, respectively. The former inputs the tampered images first for training to obtain a model file, and then inputs the adversarial examples onto the saved model file for a new round of training, while the latter is trained in exactly the opposite order to the former. We specifically state their qualitative results in the experiment (in Section 4.2.2).

4 Experiments

4.1 Experimental setup

Datasets In order to accurately evaluate the model capability, we conducts experiments on six common benchmark datasets, including DEFACTO [6], Coverage [19], Columbia [18], NIST16 [22], IMD20 [20], and Wild [21]. The number of images in the dataset and the different types of tampering are summarized in Table 2. DEFACTO [6] generated forged images by splicing, copy-moving, and removing a significant number of genuine images from MS-COCO [5]. We selected 64, 717 photos for pre-training. Coverage [19] provides 100 copy-move tampered images. The model faces a significant number of challenges since the copied object and the pasted object are so similar. There are 180 splicing images in Columbia [18]. Contrary to most datasets, which only work on objects, it concentrates on random regions that lack semantic information and the accompanying border mask, even if it only employs splicing in a tiny number of places. NIST16 [22] is made up of 564 manipulated images that have been spliced, copy-moved, and removed. IMD20 [20] produces natural pictures and categorized images of uniform regions that are beneficial for analyzing sensor noise to aid depth networks in learning to distinguish features. For the greatest aesthetic result, Wild [21] can seamlessly blend the spliced pieces into the genuine background.

Implementation details FP-Net is implemented in PyTorch [23] and optimized by Adam [17]. The experiments were all trained on an NVIDIA GeForce RTX 3090. The backbone of the model is composed of ConvNeXt-T [3]. Our backbone is initialized by ImageNet pre-trained parameters. The size of the input image is 320×320. We have adopted three training

Table 3 Training details in the experiment

Parameter	Value	Parameter	Value
Image size	320×320	Backbone	ConvNeXt-Tiny
Batch size	10	Epochs	100
Optimizer	Adam	Weight decay	5e-4
Betas	(0.9, 0.999)	Initial LR	1e-3

methods for model training, namely pre-training, fine-tuning, and benchmark-training. The first two training methods are used for comparative experiments, and the last one is used for ablation experiments. Pre-training means that FP-Net is trained on DEFACTO and tested on the images of the remaining five datasets. Fine-tuning means that FP-Net will train again on Coverage, Columbia, and NIST16 based on the pre-training results. Benchmark-training means that FP-Net trains directly on Coverage, Columbia, and NIST16. These training/testing ratios are 75:25, 126:54, and 404:160, respectively. Due to memory limitations, we randomly select 8,000 images from each epoch in DEFACTO for training. The batch size in an epoch is 10, and the initial learning rate is fixed at 1e-3. If the verification loss recorded in each epoch does not decrease within 8 epochs, the learning rate is divided by 10 until 1e-8 is reached. The training details are summarized in Table 3.

Metrics For pixel-level manipulation detection, pixel-level AUC (the Area Under the Receiver Operating Characteristic Curve) and F1 are employed for experiments. F1 is calculated from pixel-level precision and recall:

$$F1 = \frac{2TP}{2TP + FP + FN} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

where $Precision$ and $Recall$ are calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

According to [2], We acquire the best F1 scores obtained by adjusting the prediction thresholds.

Loss Function In the training process, binary cross-entropy loss is used as the training loss function. The loss function of FP-Net is formulated as follows:

$$Loss = L_{bce}(y_{gt}, y_{pred}) \tag{11}$$

where L_{bce} denotes binary cross-entropy loss, y_{gt} denotes ground-truth, y_{pred} denotes prediction mask.

4.2 Ablation study

Quantitative research is the process of collecting and analyzing numerical quantitative data based on numbers. The results are often reported in tables. The remainder of the paper is organized as follows. In this section, we provide the quantitative results of backbone, components, and training methods in Sections 4.2.1, 4.2.2, and 4.2.3, respectively.

Table 4 Quantitative pixel-level AUC/F1 comparison results of backbone on three public datasets

Backbone	Param	Columbia	Coverage	NIST16	Avg
ConvNeXt-B	89M	0.914 / 0.793	0.803 / 0.413	0.941 / 0.609	0.886 / 0.605
ConvNeXt-S	50M	0.920 / 0.817	0.828 / 0.420	0.951 / 0.616	0.900 / 0.618
ConvNeXt-T	29M	0.977 / 0.886	0.842 / 0.485	0.977 / 0.739	0.932 / 0.703

4.2.1 Quantitative results of backbone

To effectively explore the ability of ConvNeXt [3] as the backbone, we evaluated the performance of three models of ConvNeXt in benchmark-training quantitatively. Please note that we have only changed the backbone of FP-Net, and other components have not been increased or decreased. As shown in Table 4, we evaluate the quantitative pixel-level AUC/F1 results of backbone on three public datasets.

Although the results of these models on the dataset are very high, the quantitative results of ConvNeXt-T are better when the parameter quantity is the smallest. ConvNeXt-T optimizes the average AUC and F1 of ConvNeXt-S by 3.2% and 8.5%, respectively. These variants differ only in the number of channels C and the number of blocks B in each stage. The number of channels and blocks of ConvNeXt-T is less than that of ConvNeXt-S and ConvNeXt-B. We believe that the increase in the number of channels and blocks will significantly reduce the performance of the frequency separation extraction module. The frequency separation extraction module separates high-frequency information and low-frequency information at the same time through several low channels. If there are too many channels and blocks, the frequency information will lose a lot of frequency information in the process of reducing the number of channels, so it is difficult to distinguish between high-frequency information and low-frequency information.

4.2.2 Quantitative results of components

To reveal the impact of individual components in the network, we evaluated the performance of the proposed model in the benchmark training of gradually adding components. Table 5 shows the contribution of different components to quantitative results on three common datasets. It mainly includes high-frequency filters (SRM), frequency separation extraction modules (FSSM), and the global frequency attention module (GFAM).

Table 5 Quantitative pixel-level AUC/F1 comparison results of components on three public datasets

Component	Columbia	Coverage	NIST16	Avg
ConvNeXt-T	0.929 / 0.816	0.752 / 0.397	0.861 / 0.533	0.847 / 0.582
ConvNeXt-T+SRM	0.939 / 0.827	0.772 / 0.411	0.948 / 0.699	0.886 / 0.645
ConvNeXt-T+SRM+FSSM	0.954 / 0.850	0.807 / 0.414	0.976 / 0.717	0.912 / 0.660
ConvNeXt-T+SRM+FSSM+GFAM(Normal)	0.956 / 0.852	0.831 / 0.452	0.974 / 0.723	0.920 / 0.676

Table 4 indicates the result of ConvNeXt-T is the best. The perception of high-frequency information increases significantly after adding high-frequency filters. AUC and F1 on nist16 increased by 8.7% and 16.6%. The frequency separation extraction module strengthens the texture information. This is helpful to distinguish between the source region and the target region in the copy-move image. For example, AUC on coverage increased by 3.5% and F1 increased by 0.3%. AUC on nist16 increased by 2.8% and F1 increased by 1.8%. The global frequency attention module learns the global frequency information, which means that the simple low-frequency texture information and high-frequency information are difficult to achieve high-precision tamper edge recognition. The model with a global frequency attention module has been significantly improved on the three datasets.

4.2.3 Quantitative results of training methods

We simultaneously explore the impact that adversarial examples and training methods bring to the model in Table 6.

We generate adversarial images as model input by FGSM and PGD respectively. The models enhanced with PGD data performed significantly better than the other methods. Furthermore, inspired by SAT [4], we have a keen interest in the training sequence. Initially, we thought that training twice should give better results than training only once. However, it turned out that the third training method and the fourth training method failed to achieve this goal. This is possible because the two input images were disturbed by the adversarial attack resulting in inconsistent frequencies, thus altering the capability of the model to separate low-frequency features from high-frequency features and making misjudgments about the disturbed regions. Therefore, only the adversarial example input after the PGD attack is more beneficial to the model's management of the global information.

4.3 Comparison with state-of-the-art methods

To make a fair comparison with the most advanced models, we selected two training methods and evaluated the models on three standard datasets (COVERAGE [19], Columbia [18], and NIST16 [22]). Compared detection methods include classical unsupervised methods (ELA [24], NOI1 [25], CFA1 [26]) and deep learning models J-LSTM [1], H-LSTM [7], RGB-N [2], GSR-Net [10], SPAN [16], ManTra-Net* [8] and MVSS-Net [12]). ManTra-Net* indicates that we have retrained on DEFACTO using the common source code.

Table 6 Quantitative pixel-level AUC/F1 comparison results of training methods on three public datasets

Training	Columbia	Coverage	NIST16	Avg
First training method (Normal)	0.956 / 0.852	0.831 / 0.452	0.974 / 0.723	0.920 / 0.676
Second training method (+FGSM)	0.948 / 0.839	0.829 / 0.459	0.973 / 0.727	0.917 / 0.675
Second training method (+PGD)(FP-Net)	0.977 / 0.886	0.842 / 0.485	0.977 / 0.739	0.932 / 0.703
Third training method	0.947 / 0.829	0.726 / 0.429	0.970 / 0.703	0.881 / 0.653
Fourth training method	0.960 / 0.856	0.808 / 0.448	0.976 / 0.738	0.915 / 0.681

Table 7 Pixel-level AUC comparison results of the pre-training methods on five benchmark datasets

Method	Coverage	Columbia	NIST16	IMD20	Wild	Avg
MVSS-Net [12]	0.668	0.719	0.635	0.668	0.648	0.668
Mantra-Net* [8]	0.642	0.668	0.683	0.719	0.663	0.675
FP-Net	0.703	0.693	0.742	0.721	0.696	0.711

4.3.1 Pre-training comparison

To evaluate the generalization of the pre-training FP-Net, we chose ManTra-Net* and MVSS-Net as typical general tamper detection methods for comparison. Table 7 shows pixel-level AUC detection for MVSS-Net, ManTra-Net*, and FP-Net on five complete standard datasets. It can be seen that although FP-Net on Columbia is not as good as MVSS-Net, it produces the best positioning results in terms of Coverage, NIST16, IMD20, and Wild, indicating the advantages of FP-Net in generalization. In fact, since the edges of the tampered area in Columbia are very obvious, MVSS-Net containing a large number of edge extraction blocks will be more dominant in this dataset. In addition, the most significant performance improvement on FP-Net was observed when processing real manipulation images collected on IMD20. This demonstrates that FP-Net may effectively expand to real scenes without the need for extra adaptive data.

4.3.2 Fine-tuning comparison

The pixel-AUC/F1 comparison between our approach and other benchmarks on the fine-tuning model is illustrated in Table 8. The supervised architecture model is superior to the typical unsupervised method. Compared with the supervised deep learning method, FP-Net achieves the best performance on Columbia, NIST16, and average, and its performance on Coverage is second only to SPAN. It exemplifies the ability of FP-Net to pinpoint the tamper region. This is attributable to the global frequency attention module and frequency-separation sensing module. The capture of multi-scale global information greatly enhances the processing of small details in the model. Meanwhile, compared with SPAN, our method

Table 8 Pixel-level AUC comparison results of the fine-tuning models on three benchmark datasets

Method	Training	COVERAGE	Columbia	NIST16	Avg
ELA [24]	Unsupervised	0.583 / 0.222	0.581 / 0.470	0.429 / 0.236	0.531 / 0.309
NOI1 [25]	Unsupervised	0.587 / 0.269	0.546 / 0.574	0.487 / 0.285	0.540 / 0.376
CFA1 [26]	Unsupervised	0.485 / 0.190	0.720 / 0.467	0.501 / 0.174	0.569 / 0.277
J-LSTM [1]	Fine-tuning	0.614 / –	– /–	0.764 / –	0.689 /–
H-LSTM [7]	Fine-tuning	0.712 / –	– / –	0.794 / –	0.753 / –
RGB-N [2]	Fine-tuning	0.817 / 0.437	0.858 / 0.697	0.937 / 0.722	0.871 / 0.619
GSR-Net [10]	Fine-tuning	0.768 / 0.477	– / –	0.945 / 0.736	0.857 / 0.607
SPAN [16]	Fine-tuning	0.937 / 0.558	0.936 / 0.815	0.961 / 0.582	0.945 / 0.652
FP-Net	Fine-tuning	0.876 / 0.498	0.980 / 0.893	0.978 / 0.746	0.945 / 0.712

‘-’ denotes that the result is not available in the literature

Table 9 Pixel-level AUC/F1 comparison results of various manipulation types on NIST16

Method	Training	Copy-Move	Splicing	Removal	Avg
MVSS-Net [12]	Pre-training	0.675 / 0.217	0.613 / 0.231	0.651 / 0.208	0.646 / 0.219
ManTra-Net* [8]	Pre-training	0.654 / 0.150	0.783 / 0.285	0.519 / 0.151	0.652 / 0.195
FP-Net	Pre-training	0.681 / 0.220	0.821 / 0.447	0.652 / 0.223	0.718 / 0.297
SPAN [16]	Fine-tuning	0.909 / 0.405	0.992 / 0.829	0.910 / 0.499	0.937 / 0.578
FP-Net	Fine-tuning	0.942 / 0.487	0.986 / 0.803	0.969 / 0.749	0.966 / 0.680

does not rely on large-scale training data to complete high-quality work. Facts have proved that FP-Net was successful in achieving this aim.

We analyzed the performance of different methods in the table. The J-LSTM [1] and H-LSTM [7] want to input image blocks to identify local features, but this restricts the network from locating tampered areas in the global space. The operation of linear processing makes it difficult to obtain context information. RGB-N [2] significantly improves the performance by introducing the dual stream feature, but there is less training data. GSR-Net [10] provides the generation steps of tamper images and uses detection network training. However, from the results, the AUC and F1 of GSR-Net do not significantly exceed RGB-N, and the generation module needs to be improved. SPAN [16] constructs a pyramid of local self-attention blocks and effectively models the relationship between multi-scale image blocks. The quantitative results are far better than the previous models. Although FP-Net is significantly better than SPAN in other datasets and average results, the prediction in Coverage is not very optimistic. We believe that there are too few images on Coverage and that the pre-trained dataset has a large difference in frequency from the images of Coverage. Therefore, it may not be sensitive to the dataset.

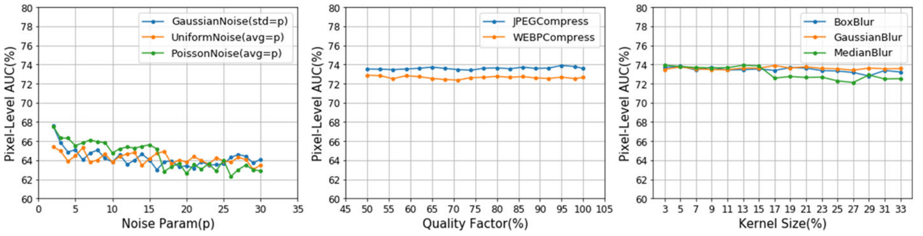
4.3.3 Manipulation types comparison

The generalization of FP-Net enables the program to efficiently investigate various operating technologies. Because it incorporates the three tampering techniques of copy-move, splicing, and removal, NIST16 was selected as the benchmark. To examine the detection performance, the evaluation model can be split into a pre-training model and a fine-tuning model for comparison. Table 9 compares the AUC/F1 performance of MVSS-Net, ManTra-Net*, SPAN, and FP-Net when testing various operation types on NIST16.

For the pre-training model, FP-Net shows extremely high positioning ability in three forgery operations. The results demonstrate that our approach handles the forgery traces left by the different tampered images quite effectively. For the fine-tuning model, the average AUC and F1 measures of FP-Net for all operations exceeded 2.9% and 10.2% of SPAN, respectively. In particular, the differences in F1 suggest that FP-Net can produce more balanced class predictions than SPAN.

4.4 Robustness evaluation

To evaluate the robustness of the proposed FP-Net for different distortions, we conducted two independent sets of robustness experiments on NIST16. Figure 6 shows the network performance curve of pixel-level AUC under pre-training for various attacks, including noise attacks with Gaussian, Uniform and Poisson, JPEG and WEBP compression, Box, Gaussian,



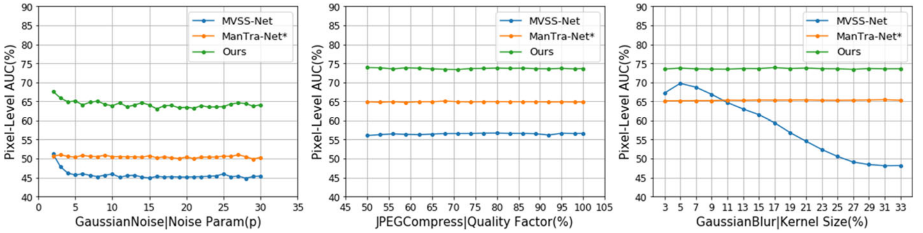
(a) (b) (c)

Fig. 6 Robustness evaluations on NIST16 under various attacks, including noise attacks with Gaussian, Uniform and Poisson, JPEG and WEBP compression, Box, Gaussian and Median blur

and Median blur, where each data point represents the performance under a fixed distortion. FP-Net tends to be stable under compression and blur attacks. It is more sensitive to noise attacks, especially Poisson Blur, but this degradation is traceable and acceptable. Figure 7 displays the robustness comparisons of our approach with state-of-the-art methods on NIST16 under specific attacks, including Gaussian noise, JPEG compression, and Gaussian blur. The FP-Net consistently outperformed the MVSS-Net and the ManTra-Net* regardless of the attack. In particular, the MVSS-Net suffers fatal degradation under Gaussian blurring, indicating that our proposed method has stronger performance and superior plausibility.

4.5 Visualization results and analysis

Figure 8 illustrates the visualization results of FP-Net under pre-training on Columbia, Coverage, NIST16, IMD20, and Wild, respectively. From top to bottom, the samples contain the tampered images, the ground-truth mask, and the predicted binary mask. The profile of the predicted mask is close to the ground-truth, which is extremely difficult. This is because the frequency distribution and the tampered objects on the pre-training dataset are substantially different from those on the test dataset. Furthermore, we compare FP-Net with ManTra-Net*, MVSS-Net, and SPAN on three benchmark datasets in Fig. 9 for visual comparisons of localization results under fine-tuning. The localization completeness and source-target object discrimination of FP-Net are significantly outperformed by the other methods. This is demonstrated by the fact that the comparison method in the first row fails to identify the red-boxed region as a tampered region; the predictions of both ManTra-Net* and MVSS-Net in the fourth row incorrectly identify the source object on the left side of the copy-move



(a) (b) (c)

Fig. 7 Robustness evaluations of MVSS-Net, ManTra-Net* and our proposed method on NIST16 under specific attacks, including Gaussian noise attack, JPEG compression attack and Gaussian blur attack

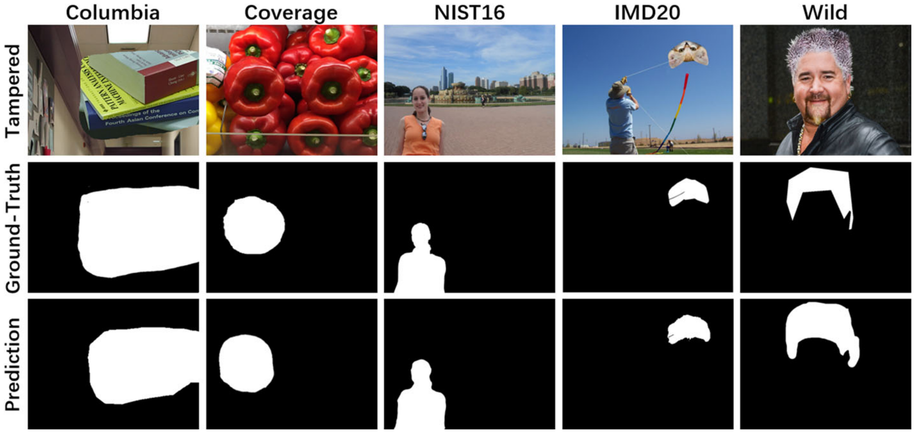


Fig. 8 Sample results of our framework from Columbia, Coverage, NIST16, IMD20, and Wild. From the top to the bottom are tampered image, ground-truth, and predicted mask

image as a spurious region, and only our method completely identifies the specific contours of the removed image in the fifth row. These cases greatly demonstrate the generality of our

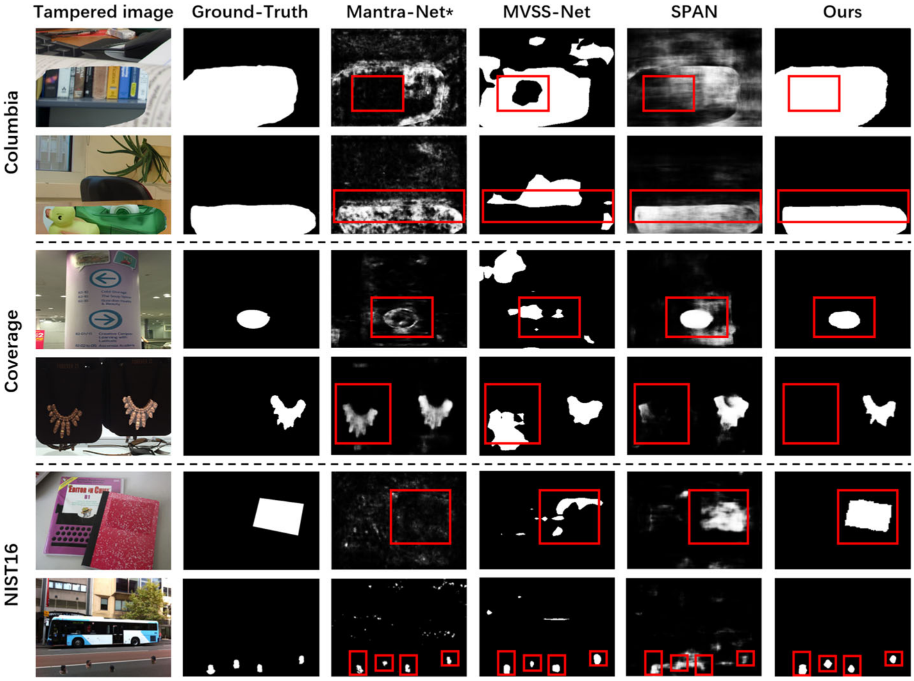


Fig. 9 Qualitative comparative visualization of localization predictions with state-of-the-art methods on Columbia, Coverage, and NIST16 under fine-tuning. From the left to the right are: tampered image, ground-truth mask, Mantra-Net* prediction, MVSS-Net prediction, SPAN prediction, and FP-Net prediction. Red rectangular boxes indicate regions of significant contrast

proposed method. We will enhance the generalizability in the future by data augmentation of the pre-training dataset.

5 Conclusion

In this paper, we investigate the frequency inconsistency between authentic and tampered regions and propose a frequency-perception network with adversarial training for image manipulation localization (FP-Net). Two functional modules are carefully designed, namely the frequency-separation sensing module and the global frequency attention module. In addition, we employ adversarial training to improve the overall performance of the model in both the spatial and frequency domains, substantially enhancing the performance and generalizability of the model. Extensive experimental results on five widely known benchmark localization datasets demonstrate that FP-Net is more effective than the other state-of-the-art methods for image manipulation localization. For example, FP-Net achieves the best pixel-level AUC/F1 of 98.0% / 89.3% on Columbia and 97.8% / 74.6% on NIST16. However, with the distribution gap between the training set and the test set in the pre-training strategy, the generalization of the model is seriously compromised. In the future, we will address this issue by introducing semi-supervision and domain shift.

Acknowledgements This research is supported by the National Key Research and Development Program of China (2018YFB0804202, 2018YFB0804203), the Regional Joint Fund of NSFC (U19A2057), the National Natural Science Foundation of China (61876070), the Ministry of Science and Technology of China (2018YFC0830103), and the Jilin Province Science and Technology Development Plan Project (20190303134SF).

Declarations

Conflicts of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bappy JH, Roy-Chowdhury AK, Bunk J, Nataraj L, Manjunath BS (2017) Exploiting spatial structure for localizing manipulated image regions. IEEE International conference on computer vision IEEE computer society(ICCV). https://doi.org/10.1109/ICCV.2017.532
2. Zhou P, Han X, Morariu VI, Davis LS (2018) Learning rich features for image manipulation detection. IEEE/CVF Conference on computer vision and pattern recognition(CVPR). https://doi.org/10.1109/CVPR.2018.00116
3. Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T (2022) A ConvNet for the 2020s. IEEE/CVF Conference on computer vision and pattern recognition(CVPR). https://doi.org/10.48550/arXiv.2201.03545
4. Zhuo L, Tan S, Li B, Huang J (2021) Self-adversarial training incorporating forgery attention for image forgery localization. IEEE Trans Inf Forensics Secur. https://doi.org/10.48550/arXiv.2107.02434
5. Lin TY, Maire M, Belongie S, Hays J, Zitnick CL (2014) Microsoft coco: common objects in context. In: Proceedings of the European conference on computer vision (ECCV):740-755
6. Mahfoudi G, Tajini B, Retraint F (2019) DEFACTO: image and face manipulation dataset. European signal processing conference (EUSIPCO)
7. Bappy JH, Simons C, Nataraj L, Manjunath BS, Roy-Chowdhury AK (2019) Hybrid LSTM and Encoder-Decoder Architecture for Detection of Image Forgeries. IEEE Trans Image Process. https://doi.org/10.1109/TIP.2019.2895466

8. Wu Y, AbdAlmageed W, Natarajan P (2019) ManTra-Net: manipulation tracing network for detection and localization of image forgeries with anomalous features. IEEE/CVF Conference on computer vision and pattern recognition (CVPR). https://doi.org/10.1109/CVPR.2019.00977
9. Salloum, Ren R, Kuo Y (2018) Image splicing localization using a multi-task fully convolutional network (MFCN). J. Visual Commun Image Represent 201–209
10. Zhou P, Chen BC, Han X, Najibi M, Davis L (2020) Generate, segment, and refine: towards generic manipulation segmentation. Proceedings of the AAAI Conference on Artificial Intelligence. https://doi.org/10.48550/arXiv.2105.14447
11. Hao J , Zhang Z , Yang S (2021) TransForensics: image forgery localization with dense self-attention. IEEE International conference on computer vision. https://arxiv.org/pdf/2108.03871.pdf
12. Chen X, Dong C, Ji J, Cao J, Li X (2021) Image manipulation detection by multi-view multi-scale supervision. IEEE/CVF Conference on computer vision and pattern recognition (CVPR). https://doi.org/10.48550/arXiv.2104.06832
13. Liu X, Liu Y, Chen J, Liu X (2021) PSCC-Net: progressive spatio-channel correlation network for image manipulation detection and localization. https://doi.org/10.48550/arXiv.2103.10596
14. Goodfellow IJ, Shlens J , Szegedy C (2014) Explaining and harnessing adversarial examples. Computer vision. Computer science. https://doi.org/10.48550/arXiv.1412.6572
15. Madry A , Makelov A , Schmidt L (2018) Towards deep learning models resistant to adversarial attacks. International conference on learning representations. Computer science. https://doi.org/10.48550/arXiv.1706.06083
16. Hu X, Zhang Z, Jiang Z, Chaudhuri S, Yang Z, Nevatia R (2020) SPAN: spatial pyramid attention network for image manipulation localization. In: Proceedings of the European conference on computer vision (ECCV)
17. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. Computer science. https://doi.org/10.48550/arXiv.1412.6980
18. Ng TT, Hsu J, Chang SF (2009) Columbia image splicing detection evaluation dataset. DVMM lab. Columbia Univ CalPhotos Digit Libr. http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/photographers.htm
19. Wen B, Ye Z, Subramanian R, Ng TT, Winkler S (2016) COVERAGE - A novel database for copy-move forgery detection. IEEE International conference on image processing (ICIP). https://doi.org/10.1109/ICIP.2016.7532339
20. Novozamsky A, Mahdian B, Saic S (2020) IMD2020: a large-scale annotated dataset tailored for detecting manipulated images. IEEE Winter applications of computer vision workshops (WACVW). https://doi.org/10.1109/WACVW50321.2020.9096940
21. Huh M, Liu A, Owens A, Efros AA (2018) Fighting fake news: image splice detection via learned self-consistency. Proceedings of the European conference on computer vision (ECCV)
22. NIST: Nist nimble 2016 datasets (2016). https://www.nist.gov/itl/iad/mig/
23. Paszke A, Gross S, Massa F, Lerer A, Chintala S (2019) Pytorch: an imperative style, high-performance deep learning library. 33rd Conference on neural information processing systems (NeurIPS). https://doi.org/10.48550/arXiv.1912.01703
24. Krawetz N, Solutions HF (2017) A pictures worth. Hacker Factor Solutions
25. Mahdian B, Saic S (2009) Using noise inconsistencies for blind image forensics. Image Vision Comput pp 1497–1503
26. Ferrara P, Bianchi T, Rosa De, Piva A (2012) Image forgery localization via fine-grained analysis of CFA artifacts. IEEE Trans Inf Forensics Secur. https://doi.org/10.1109/TIFS.2012.2202227
27. Qian Y, Yin G, Sheng L (2020) Thinking in frequency: face forgery detection by mining frequency-aware clues. European Conference on Computer Vision. https://doi.org/10.1007/978-3-030-58610-26
28. Luo W, Huang J, Qiu G (2020) Robust detection of region-duplication forgery in digital image. International conference on pattern recognition
29. Bayram S, Sencar HT, Memon ND (2009) An efficient and robust method for detecting copy-move forgery[C]// IEEE International Conference on Acoustics
30. Zhao J, Guo J (2013) Passive forensics for copy-move image forgery using a method based on DCT and SVD. Forensic Science International
31. Lowe D (2004) Distinctive image features from scale-invariant key points. International Journal of Computer Vision
32. Gao Z, Sun C, Cheng Z (2021) TBNet: two-stream boundary-aware network for generic image manipulation localization. International Journal of Computer Vision. https://doi.org/10.48550/arXiv.2108.04508
33. Luo Y, Zhang Y, Yan J (2021) Generalizing face forgery detection with high-frequency features. Computer Vision and Pattern Recognition. https://doi.org/10.48550/arXiv.2103.12376

34. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. Medical image computing and computer-assisted intervention- MICCAI

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