# MUN: Image Forgery Localization Based on M$^3$ Encoder and UN Decoder

**Yaqi Liu[1*], Shuhuan Chen[2,3*], Haichao Shi[2], Xiao-Yu Zhang[2†], Song Xiao[1], Qiang Cai[4]**

[1]Beijing Electronic Science and Technology Institute
[2]Institute of Information Engineering, Chinese Academy of Sciences
[3]School of Cyber Security, University of Chinese Academy of Sciences
[4]Beijing Key Laboratory of Big Data Technology for Food Safety, Beijing Technology and Business University
liuyaqi@besti.edu.cn, {chenshuhuan, shihaichao, zhangxiaoyu}@iie.ac.cn, xiaosong@mail.xidian.edu.cn, caiq@th.btbu.edu.cn

## Abstract

Image forgeries can entirely change the semantic information of an image, and can be used for unscrupulous purposes. In this paper, we propose a novel image forgery localization network named as MUN, which consists of an M$^3$ encoder and a UN decoder. Firstly, the M$^3$ encoder is constructed based on a Multi-scale Max-pooling query module to extract Multi-clue forged features. Noiseprint++ is adopted to assist the RGB clue, and its deployment methodology is discussed. A Multi-scale Max-pooling Query (MMQ) module is proposed to integrate RGB and noise features. Secondly, a novel UN decoder is proposed to extract hierarchical features from both top-down and bottom-up directions, reconstructing both high-level and low-level features at the same time. Thirdly, we formulate an IoU-recalibrated Dynamic Cross-Entropy (IoUDCE) loss to dynamically adjust the weights on forged regions according to IoU which can adaptively balance the influence of authentic and forged regions. Last but not least, we propose a data augmentation method, i.e., Deviation Noise Augmentation (DNA), which acquires accessible prior knowledge of RGB distribution to improve the generalization ability. Extensive experiments on publicly available datasets show that MUN outperforms the state-of-the-art works.

**Code** — https://github.com/MrHuan3/MUN

## Introduction

With ubiquitous digital cameras and highly developed image editing tools, non-expert users can fabricate realistic forged images which may be used for vicious purposes, interfering with public security. For different types of image forgeries, e.g., splicing, copy-move, removal, there are different forensics solutions. For splicing detection, statistical inconsistencies left by tampering operations in RGB or spectral domains are researched (Yang et al. 2024; Xu et al. 2023). In copy-move forgery detection, visual similar features are extracted and matched in a hand-crafted manner (Wang et al. 2023; Li and Zhou 2019) or an end-to-end deep learning style (Liu et al. 2022b; Weng et al. 2024). Some works even attempt to detect splicing traces using visual similar features in the constrained image splicing detection and localization task (Liu et al. 2019; Tan et al. 2023). For removal forgeries, inpainting detection has been widely explored especially with the emergence of the deep learning based inpainting techniques (Li et al. 2024; Wu and Zhou 2022). However, with a suspected forged image at hands, we do not know it is tampered by which kinds of forgeries, and abovementioned methods are only effective under specific circumstances. In recent years, the image forgery localization task which aims to detect multi-type forgeries attracts more and more attention (Liu et al. 2023; Zhang et al. 2024).

In order to improve the discrimination and generalization abilities of detecting multi-type forgeries, numerous feature extractors and architectures have been explored. Some researches focus on locating forged regions with Convolutional Neural Networks (CNNs) (Zhuo et al. 2022; Niloy, Kumar Bhaumik, and Woo 2023). Recently, Transformer (Vaswani et al. 2017) has shown promising performance in down-stream computer vision tasks (Jain et al. 2023; Li et al. 2023). Researchers also applied Transformer or its variants to detect forgeries (Shi, Chen, and Zhang 2023; Wang et al. 2022). It is still an ambiguous question which kind of architecture is more suitable for image forgery localization in consideration of the performance and complexity.

Besides, researchers attempt to investigate multiple clues, e.g., DCT (Bianchi, De Rosa, and Piva 2011), SRM (Zhou et al. 2018) and Laplacian of Gaussian (Guo et al. 2023), to provide additional information and assist the RGB features. In (Cozzolino and Verdoliva 2020), a different perspective is considered, and the extraction of low-level artifacts is carried out by learning a sort of "camera model fingerprint" named as Noiseprint which bears traces of in-camera processing steps and is obtained by means of a siamese network. In (Guillaro et al. 2023), Noiseprint++ is proposed to overcome Noiseprint's limitations of poor robustness to image impairments induced by out-camera processes. Noiseprint++ is an improved version of Noiseprint, and it can work in more challenging scenarios. While it is still an open question how Noiseprint++ is deployed in the image forgery localization task when processing images with different resolutions.

In this paper, we propose a novel image forgery localization network, i.e., MUN, which consists of an M$^3$ encoder

*These authors contributed equally.
†indicates the corresponding author.

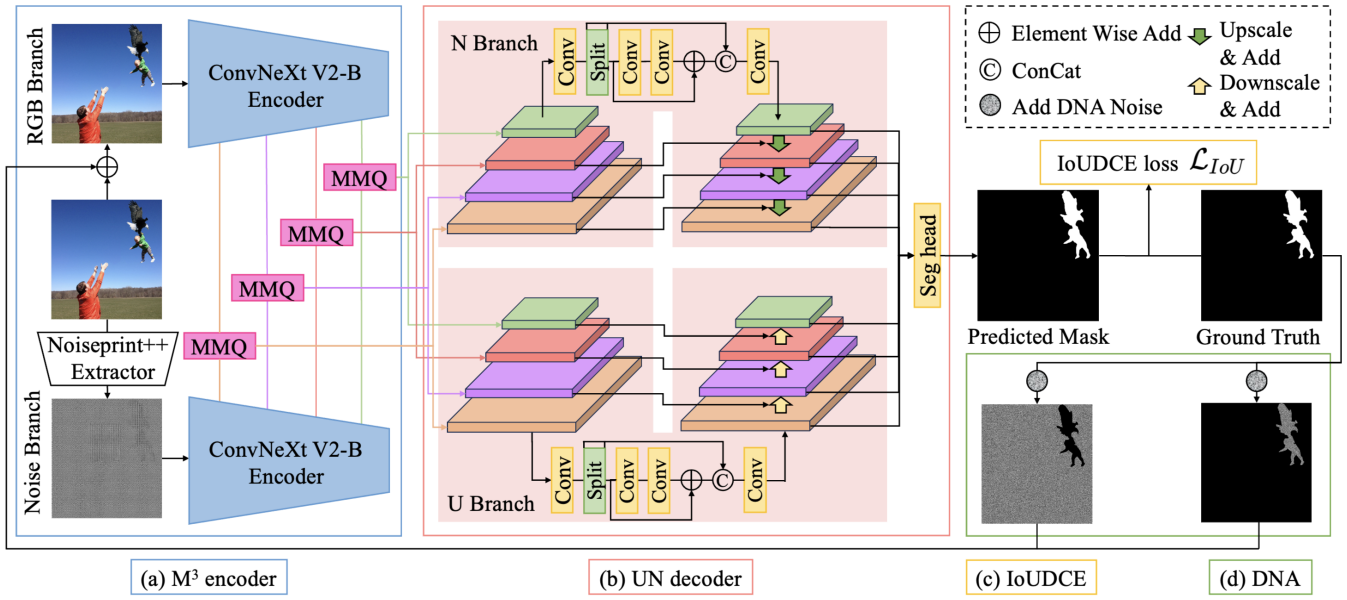


Figure 1: MUN framework. (a) M$^3$ encoder consists of a RGB branch and a noise branch. The feature maps of both branches are extracted by ConvNeXt V2, and the multi-clue features are integrated by the MMQ module. (b) UN decoder has parallel U and N branches which integrates hierarchical feature maps from MMQ. The integrated feature maps are concatenated to reconstruct the predicted mask. (c) IoUDCE loss dynamically recalibrated weights on forged regions. (d) DNA adds uniform noises multiplied with differences of RGB mean values bewteen training and universal images.

and a UN decoder as shown in Figure 1. In the course of our research, the first question we face is which kind of backbone is more suitable for image forgery localization. We select three the state-of-the-art backbones for comparison, i.e., CNN-style networks of ResNet (He et al. 2016), ConvNeXt V2 (Woo et al. 2023) and a Transformer-style network of Swin (Liu et al. 2021). ConvNeXt V2 shows better accuracy and lower computation cost with a similar number of parameters. Then, with the same backbone, i.e., ConvNeXt V2, we discuss the deployment of Noiseprint++ under three circumstances: "resize", "Noiseprint++ after resize", and "resize after Noiseprint++". And "resize after Noiseprint++" wins. On account of adequate comparisons on backbones and Noiseprint++, we construct a dual-stream feature extractor, and propose a Multi-scale Max-pooling Qurey (MMQ) module with which we can obtain the strong correlation between RGB and Noiseprint++ features. Since the Multi-clue Multi-scale Max-pooling nature, our feature extractor is named as **M$^3$ encoder**. To get ample hierarchical features, we choose hierarchical outputs from M$^3$ encoder and put them into an exquisitely designed two-branch decoder, i.e., **UN decoder**. In UN decoder, hierarchical feature maps are cloned into two branches, i.e., U and N branches. In the U branch, we extract low-order features with rich spatial information from the low-level feature map via light-weight successive convolutional layers with residual and concatenation operations, and pass the acquired feature up to fuse with other higher-level hierarchical features step by step. In a similar way, we extract high-order features from the high-level feature map via the same successive convolutional layers, and pass it down to fuse with other features. Then we concatenate U and N branch features to reconstruct the predicted mask. Besides, a dynamic loss function, i.e., IoU-recalibrated Dynamic Cross-Entropy (**IoUDCE**) loss, is designed to adjust the class weights of positive and negative samples by batch, guiding MUN to adaptively focus on intractable manipulated regions. Finally, we put forward a novel data augmentation method, named Deviation Noise Augmentation (**DNA**), by adjusting the RGB distribution of training images to enhance the generalization ability of MUN. Our contributions can be summarized as follows.

- A novel image forgery localization network, i.e., MUN, is proposed based on an M$^3$ encoder and a UN decoder.
- We propose an M$^3$ encoder which has dual-stream ConvNeXt V2 feature extractors to extract features from RGB and Noiseprint++ domains. MMQ module is proposed to integrate dual-domain features. We verify how Noiseprint++ can be deployed.
- A UN decoder is proposed to extract both low-order and high-order information from parallel U and N branches, enabling the extraction of more subtle forgery features.
- The IoUDCE loss is proposed to dynamically balance weights and focus on intractable forged regions according to IoU of predicted masks.
- DNA is proposed by narrowing the gap between synthetic training images and universal images to boost the generalization ability.
- Experiments show that MUN works not only on artificially tampered datasets but also on AI generated ones.

## Related Work

Conventional image forgery localization methods can be broadly divided into high-level methods, and low-level methods. In high-level methods, high-level semantic information is commonly analysed. For example, pristine and forged objects can reflect different lights and shadows (Peng et al. 2016), the properties of perspective and geometry can tell whether objects are contacted or not (Peng et al. 2017), and the inconsistency of blur types may offer clues for forgery localization (Bahrami et al. 2015). In low-level methods, pixel-level statistical inconsistencies are commonly investigated via hand-crafted features. The commonly seen low-level features include, PRNU (Photo Response Non-Uniformity) which reflects the non-uniformity of image sensors responding with a different voltage level (Korus and Huang 2017), CFA (Color Filter Array) which is a mosaic of tiny color filters to capture color information (Popescu and Farid 2005), SRM (Spatial Rich Model) which uses statistics of neighboring noise residual samples as features to capture dependency changes (Zhou et al. 2018). In recent years, with the prosperity of deep learning techniques, deep leaning based image forgery localization becomes the mainstream.

Researchers proposed numerous CNN-based methods to localize forged images. Multi-scale tampering detectors based on CNNs are proposed to generate a series of complementary tampering possibility maps (Liu et al. 2018). Niu et al. (Niu et al. 2021) used a local estimate of the primary quantization matrix to distinguish between forged regions and clustered the image blocks to refine the predicted results. CSR-Net (Zhang et al. 2024) utilizes a parameterization method to fit the target segmentation area adaptively by thinking the image forgery localization task from the perspective of regression.

Transformer has further promoted the progress in the field of image forgery localization. ObjectFormer adopts Transformer layers to fuse high-frequency features with RGB features to capture the patch-level consistencies (Wang et al. 2022). TBFormer uses Vision Transformer to extract tampered traces from both RGB and Bayar noise domains (Liu et al. 2023). TANet uses a stacked multi-scale Transformer branch to detect structured abnormalities of the input image at multiple levels (Shi, Chen, and Zhang 2023).

## Method

Our MUN is composed of an M$^3$ encoder and a UN decoder. MUN is trained with an IoUDCE loss on sythetic forged images after DNA data augmentation. We first introduce our M$^3$ encoder with the proposed MMQ. Then, we introduce UN decoder. After the introduction of our main architecture, we discuss the IoUDCE loss and the DNA methodology.

### M$^3$ Encoder

The M$^3$ encoder investigates clues from RGB and Noiseprint++ (NPP) domains. The RGB domain can provide clues like contrastive differences and unnatural boundaries between authentic and forged regions. Noiseprint++ can obtain low-level clues including camera models and image editing histories. In our MUN, the RGB image and Noiseprint++ map are fed into two parallel ConvNeXt V2 (Woo et al. 2023) feature extrators. The intermediate hierarchical features from two domains are integrated by the MMQ module which uses features from RGB domain to perform a multi-scale max-pooling query on the noise features.

To explore the right way of using Noiseprint++, we design three versions of operations, i.e., "resize", "Noiseprint++ after resize" and "resize after Noiseprint++". "Resize after Noiseprint++" achives the best performance. Specifically, the RGB image $\mathbf{I}_{RGB} \in \mathbb{R}^{H_o \times W_o \times 3}$ is fed into the Noiseprint++ module to obtain the noise map $\mathbf{I}_{NPP} \in \mathbb{R}^{H_o \times W_o}$, where $H_o$ and $W_o$ denote the original height and width of the image. $\mathbf{I}_{NPP}$ is duplicated along the channel dimension to get $\dot{\mathbf{I}}_{NPP} \in \mathbb{R}^{H_o \times W_o \times 3}$. Then we resize both $\mathbf{I}_{RGB}$ and $\dot{\mathbf{I}}_{NPP}$ to get the resized RGB image $\bar{\mathbf{I}}_{RGB} \in \mathbb{R}^{H \times W \times 3}$ and Noiseprint++ map $\bar{\mathbf{I}}_{NPP} \in \mathbb{R}^{H \times W \times 3}$, $H$ and $W$ denote input height and width.

Two parallel ConvNeXt V2 networks are constructed to obtain features from both RGB and Noiseprint++ domains. Two ConvNeXt V2 branches have the same architecture while their weights do not shared, so that each branch is more capable of focusing on their specific feature extraction. We choose the 3rd, 6th, 33th and 36th layer outputs from both branches as the intermediate feature maps. Let $\mathbf{C} = \{\mathbf{C}^{(0)}, \mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \mathbf{C}^{(3)}\}$ indicate the hierarchical features from the RGB domain, and $\mathbf{N} = \{\mathbf{N}^{(0)}, \mathbf{N}^{(1)}, \mathbf{N}^{(2)}, \mathbf{N}^{(3)}\}$ indicate the hierarchical features from the noise domain.

In order to investigate the correlation from RGB to noise domain and integrate their features, we construct a Multi-scale Max-pooling Query (MMQ) module, as shown in Figure 2. Take $\mathbf{C}^{(0)}$ and $\mathbf{N}^{(0)}$ as example, four max-pooling operations with different kernel sizes followed by $1 \times 1$ convolutions are performed on the RGB feature map $\mathbf{C}^{(0)}$, obtaining $\mathbf{Q}^{(0)} = \{\mathbf{Q}^{(0)}_{k \times k}\}$. It can be formulated as:

$$\mathbf{Q}^{(0)}_{k \times k} = Conv_{1 \times 1}(MaxPooling_{k \times k}(\mathbf{C}^{(0)})) \quad (1)$$

where $k \times k$ denotes the kernel size of max-pooling and $k \in \{1, 5, 11, 15\}$, $Conv_{1 \times 1}$ denotes the $1 \times 1$ convolution and $MaxPooling_{k \times k}$ stands for max-pooling with the $k \times k$ kernel. $\mathbf{Q}^{(0)}_{k \times k}$ contains the most significant local features at the scale of $k \times k$. We perform multi-scale query operations by calculating the matrix multiplication of $\mathbf{Q}^{(0)}_{k \times k}$ with Noiseprint++ feature map $\mathbf{N}^{(0)}$:

$$\dot{\mathbf{Q}}^{(0)}_{k \times k} = \mathbf{Q}^{(0)}_{k \times k} \otimes \mathbf{N}^{(0)} \quad (2)$$

where $\otimes$ denotes the matrix multiplication. After that, we construct a $1 \times 1$ convolution to reduce the dimension of $\dot{\mathbf{Q}}^{(0)}_{k \times k}$ and get $\ddot{\mathbf{Q}}^{(0)}_{k \times k}$. The channel dimension of $\dot{\mathbf{Q}}^{(0)}_{k \times k}$ is 4 times of $\ddot{\mathbf{Q}}^{(0)}_{k \times k}$. Finally, we concatenate $\mathbf{C}^{(0)}$ and four $\ddot{\mathbf{Q}}^{(0)}_{k \times k}$ to get the multi-scale feature map:

$$\begin{aligned} \hat{\mathbf{Q}}^{(0)} = Concat(&\mathbf{C}^{(0)}, \ddot{\mathbf{Q}}^{(0)}_{1 \times 1}, \\ &\ddot{\mathbf{Q}}^{(0)}_{5 \times 5}, \ddot{\mathbf{Q}}^{(0)}_{11 \times 11}, \ddot{\mathbf{Q}}^{(0)}_{15 \times 15}) \end{aligned} \quad (3)$$

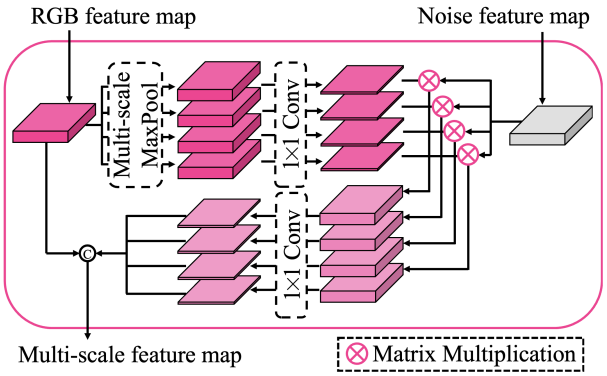


Figure 2: Multi-scale Max-pooling Query (MMQ).

where $Concat$ denotes concatenation along the channel dimension. The same MMQ modules are operated on $\mathbf{C}^{(1)}$, $\mathbf{C}^{(2)}$, $\mathbf{C}^{(3)}$ and $\mathbf{N}^{(1)}$, $\mathbf{N}^{(2)}$, $\mathbf{N}^{(3)}$ to get $\hat{\mathbf{Q}}^{(1)}$, $\hat{\mathbf{Q}}^{(2)}$, $\hat{\mathbf{Q}}^{(3)}$. With multi-scale max-pooling queries, we are capable of constructing complex relationship between RGB and NPP domains. The feature maps we extract from $\mathbf{M^3}$ encoder can be denoted as $\hat{\mathbf{Q}} = \{\hat{\mathbf{Q}}^{(0)}, \hat{\mathbf{Q}}^{(1)}, \hat{\mathbf{Q}}^{(2)}, \hat{\mathbf{Q}}^{(3)}\}$.

## UN Decoder

We design a UN decoder which extracts hierarchical features from two opposite directions. The UN decoder contains two branches, i.e., the U branch and the N branch, and each branch owns a clone of the feature map $\hat{\mathbf{Q}}$. We conduct similar-but-opposite operations on these two branches. For the U branch, the low-level layer $\hat{\mathbf{Q}}^{(0)}$ is taken by a lightweight successive convolutional layers to generate $\mathbf{BU}^{(0)}$. Firstly, we double the $\hat{\mathbf{Q}}^{(0)}$ channel dimension via a $1 \times 1$ convolutional module and then split it into two tensors $\mathbf{X}_0$ and $\mathbf{X}_1$ with the same channel dimensions. The split operation aims to reduce the computational costs and improve the representation ability. We obtain $\mathbf{X}'$ by passing $\mathbf{X}_1$ in turn through two modules $f_{3\times3}^{conv}$. The convolution module $f_{3\times3}^{conv}$ consists of a $3 \times 3$ convolutional layer, a batch norm layer, and a SiLU activation layer. Then we integrate $\mathbf{X}_0$, $\mathbf{X}_1$ and $\mathbf{X}'$ to get $\mathbf{BU}^{(0)}$:

$$\begin{aligned}\mathbf{BU}^{(0)} =& f_{1\times1}^{conv}(Concat(\mathbf{X}_0, \mathbf{X}_1, \\ & f_{3\times3}^{conv}(f_{3\times3}^{conv}(\mathbf{X}_1)) + \mathbf{X}_1))\end{aligned} \tag{4}$$

where $f_{1\times1}^{conv}$ denotes the $1 \times 1$ convolution module with batch normalization and SiLU activation. Then we update and fuse the features in the U branch bottom-up:

$$\mathbf{BU}^{(i+1)} = \hat{\mathbf{Q}}^{(i+1)} + f_{3\times3}^{conv\downarrow}(\mathbf{BU}^{(i)}) \tag{5}$$

where $i$ denotes the $i$-th feature map, $i \in \{0, 1, 2\}$, $f_{3\times3}^{conv\downarrow}$ denotes a down-sampling convolution module with a batch norm layer and a SiLU layer. Thus, we get the U branch feature maps $\mathbf{BU} = \{\mathbf{BU}^{(0)}, \mathbf{BU}^{(1)}, \mathbf{BU}^{(2)}, \mathbf{BU}^{(3)}\}$.

Similar to the process on the U branch, we construct another successive convolutional layers by taking the high-level feature $\hat{\mathbf{Q}}^{(3)}$ to obtain $\mathbf{TD}^{(3)}$. We double the $\hat{\mathbf{Q}}^{(3)}$ channel dimension and then split it into two tensors $\mathbf{Y}_0$ and $\mathbf{Y}_1$ with the same channel dimensions. After two convolution modules $f_{3\times3}^{conv}$, we obtain $\mathbf{Y}'$ from $\mathbf{Y}_1$. We add $\mathbf{Y}'$ with $\mathbf{Y}_1$ and then concatenated with $\mathbf{Y}_0$ and $\mathbf{Y}_1$, following a $1 \times 1$ convolution module to get $\mathbf{TD}^{(3)}$. And then we update and fuse the features in the N branch top-down, i.e., $\mathbf{TD}^{(i-1)} = \hat{\mathbf{Q}}^{(i-1)} + Resize(\mathbf{TD}^{(i)}), i \in \{1, 2, 3\}$, we generate $\mathbf{TD} = \{\mathbf{TD}^{(0)}, \mathbf{TD}^{(1)}, \mathbf{TD}^{(2)}, \mathbf{TD}^{(3)}\}$. Here, we expand the size of features by bilinear interpolation in the $Resize$ operation. Notably, the two successive convolutional layers in U and N branches do not have the same weights, and the U branch pays attention to the subtle details while the N branch focuses on the global abstract features.

After all these operations, we concatenate $\mathbf{BU}$ and $\mathbf{TD}$ between the same level to obtain joint level features $\mathbf{UN} = \{\mathbf{UN}^{(0)}, \mathbf{UN}^{(1)}, \mathbf{UN}^{(2)}, \mathbf{UN}^{(3)}\}$, specifically:

$$\mathbf{UN}^{(i)} = Resize(f_{1\times1}^{conv}(Concat(\mathbf{BU}^{(i)}, \mathbf{TD}^{(i)}))) \tag{6}$$

With four $\mathbf{UN}^{(i)}$ at the same size as $\mathbf{UN}^{(0)}$, we concatenate $\mathbf{UN}$ and pass them through a $3 \times 3$ convolutional module, a $1 \times 1$ convolutional layer and a resize operation, to obtain the predicted mask $\mathbf{M} \in \mathbb{R}^{(H_o \times W_o)}$:

$$\mathbf{M} = Resize(Conv_{1\times1}(f_{3\times3}^{conv}(Concat(\mathbf{UN})))) \tag{7}$$

## IoUDCE Loss

We introduce an IoU-recalibrated dynamic cross-entropy (IoUDCE) loss, which could adjust the class weights on positive and negative predictions during the training process. Let $\mathbf{GT} \in \mathbb{R}^{H_o \times W_o}$ denote the ground truth of the forged image, and the forged part of $\mathbf{GT}$ is 1 while the pristine part is 0. We resize $\mathbf{GT}$ and $\mathbf{M}$ and obtain $\mathbf{GT}_b \in \mathbb{R}^{H \times W}$ and $\mathbf{M}_b \in \mathbb{R}^{H \times W}$, respectively, and then binarize $\mathbf{M}_b$ by changing the part of $\mathbf{M}_b$ greater than 0.5 to 1 and the part less than 0.5 to 0. The IoUDCE loss can be calculated as follows:

$$\beta_{IoU} = \sum_{i}^{B} \frac{\mathbf{GT}_b(i) \cap \mathbf{M}_b(i)}{\mathbf{GT}_b(i) \cup \mathbf{M}_b(i)} \tag{8}$$

$$\begin{aligned}\mathcal{L}_{IoU} = -\frac{1}{B}\frac{1}{HW}\sum_{i=1}^{B}\sum_{j=1}^{HW}(&\alpha_0(1 - y(i,j)) \\ &\times \log(1 - p(i,j)) + \alpha_1 \\ &\times e^{\lambda(1-\beta_{IoU})} y(i,j) \log p(i,j))\end{aligned} \tag{9}$$

where $B$ denotes the batch size, $\cap$ and $\cup$ denote the intersection and the union, respectively, $y(i,j)$ denotes the $j$-th ground truth label of the $i$-th image in the batch and $p(i,j)$ denotes the $j$-th predicted score of the $i$-th image in the batch to be forged. $\alpha_0$, $\alpha_1$ and $\lambda$ are hyperparameters.

When the predicted mask is far different from the ground truth, $1 - \beta_{IoU}$ tends to be large, which leads to the increase of the class weight on positive regions, and guides MUN to pay more attention to them. With the training process going, the differences between predicted masks and ground truths

narrow, resulting in $1-\beta_{IoU}$ approaching to 0, and the positive class weight approaching to $\alpha_1$. Of course, when we face a "bad" batch of images with small $\beta_{IoU}$, we can appropriately improve the weight on the positive regions.

### Deviation Noise Augmentation

Deviation Noise Augmentation (DNA) is proposed based on the hypothesis that models are more likely to achieve better results on the dataset which has a similar distribution with the training dataset. The core idea of DNA is to narrow the distribution gap between universal images and training images. We choose the mean values of RGB channels to be the distinguishable distribution. We first calculate the mean values $m^t = \{m^t_R, m^t_G, m^t_B\}$ of the RGB channels of the training dataset as well as the mean values $m^u = \{m^u_R, m^u_G, m^u_B\}$ of ImageNet (Deng et al. 2009) , because we think that ImageNet can represents universal images of our real world. Then we subtract $m^t$ from $m^u$, obtaining the differences $\mathbf{d} = \{d_k\}$ where $d_k = m^u_k - m^t_k$, $k \in \{R, G, B\}$. We assume both pristine and forged parts in the manipulated images obey the distribution of universal images, and we generate two uniformly distributed noises $\mathbf{N}_p \in \mathbb{R}^{H\times W\times 3}$ and $\mathbf{N}_f \in \mathbb{R}^{H\times W\times 3}$. $\mathbf{GT}_b$ is duplicated along the channel dimension to get $\hat{\mathbf{GT}}_b \in \mathbb{R}^{H\times W\times 3}$ and we multiply both $\mathbf{N}_p$ and $\mathbf{N}_f$ with specific $d_k$ at the corresponding channel, obtaining $\hat{\mathbf{N}}_p \in \mathbb{R}^{H\times W\times 3}$ and $\hat{\mathbf{N}}_f \in \mathbb{R}^{H\times W\times 3}$:

$$\hat{\mathbf{N}}_p = Concat(d_R\mathbf{N}_p(0), d_G\mathbf{N}_p(1), d_B\mathbf{N}_p(2)) \tag{10}$$

$$\hat{\mathbf{N}}_f = Concat(d_R\mathbf{N}_f(0), d_G\mathbf{N}_f(1), d_B\mathbf{N}_f(2)) \tag{11}$$

where $\mathbf{N}_p(i)$ and $\mathbf{N}_f(i)$ denote the $i$-th dimension of $\mathbf{N}_p$ and $\mathbf{N}_f$, respectively. Then we perform the intersection on the pristine part of $\hat{\mathbf{GT}}_b$ and $\hat{\mathbf{N}}_p$, as well as on the forged part of $\hat{\mathbf{GT}}_b$ and $\hat{\mathbf{N}}_f$ to obtain $\mathbf{I}_{DNA} \in \mathbb{R}^{H\times W\times 3}$ as follows:

$$\begin{aligned}\mathbf{I}_{DNA} = \bar{\mathbf{I}}_{RGB} &+ (\hat{\mathbf{N}}_p \circ (\hat{\mathbf{GT}}_b = 0)) \\ &+ (\hat{\mathbf{N}}_f \circ (\hat{\mathbf{GT}}_b = 1))\end{aligned} \tag{12}$$

where $\circ$ denotes the Hadamard product.

## Experiments

### Implementation and Evaluation Details

**Implementation details.** MUN is built based on MMSegmentation. We use Noiseprint++ to generate the noise map of the original-sized image and then resize both the image and the noise map to $512 \times 512$ patches as inputs. The optimizer is SGD, and the learning strategy can be calculated as follows:

$$lr_c = \begin{cases} lr_s + (lr_0 - lr_s) \times \frac{iter_c}{iter_w}, iter_c < iter_w \\ lr_0 \times (1 - \frac{1}{iter_t - iter_c + 1}), iter_c \geq iter_w \end{cases} \tag{13}$$

where $lr_c$ denotes the current learning rate. $lr_s = 10^{-6}$ is the start learning rate and $lr_0 = 0.01$ is the initial learning rate, $iter_c$ stands for the current number of iterations, $iter_w = 1500$ denotes the warm-up number of iterations and $iter_t$ means the total number of iterations. In Eq. (9), $\alpha_0$, $\alpha_1$ are both set to 1.0 and $\lambda$ is set to 1.5. The batch size is 7 and the epoch is 4. Our training and inference experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU.

**Experimental datasets.** We train our model on the Synthesized Dataset in (Liu et al. 2023) and adopt five datasets, i.e., NIST16 (Guan et al. 2019), CASIA v1.0 (Dong, Wang, and Tan 2013), IMD2020 (Novozámský, Mahdian, and Saic 2020), CocoGlide (Guillaro et al. 2023), Wild (Huh et al. 2018), for evaluation. The Synthesized Dataset is built based on CASIA v2.0 (Wei et al. 2022) and ADE20k (Zhou et al. 2019), and contains splicing, copy-move and removal tampered images. There are 156006 synthesized images in the Synthesized Dataset (140432 for training, 7787 for validation, and 7787 for testing). NIST16 and IMD2020 contain splicing, copy-move, and removal images, CASIA v1.0 contains splicing and copy-move images, and Wild only contains splicing images. CocoGlide is an AI generated dataset using the GLIDE diffusion model (Nichol et al. 2022).

**Experimental metric.** We use F1 score, IoU, Accuracy and AUC as evaluation metrics which are commonly used in image forgery localization (Guillaro et al. 2023). When binarizing the predicted masks, the threshold is fixed to 0.5.

### Ablation Study

Four groups of experiments for ablation study are conducted on the testing set of the Synthesized Dataset. Details are shown as follows.

**Backbone evaluation.** Swin-B (Liu et al. 2021), ResNet-152 (He et al. 2016) and ConvNeXt V2-B (Woo et al. 2023) are compared as shown in Table 1. With the same $512 \times 512$ image as input, ConvNeXt V2-B has less FLOPS than Swin-B and ResNet-152. Since ConvNeXt V2-B achieves a higher F1 score than both Swin-B and ResNet-152 with less computational complexity and similar parameters, we select ConvNeXt V2-B as the backbone. As for the comparison between Transformer and convolution, our consideration is that local inconsistencies are more important in image forgery localization, and well-designed convolutional networks can generate ideal results without computationally complex attention operations. Thus, we concentrate on different local feature extraction strategies in our work.

**Ablation study of each component in MUN.** In Table 2, "RGB" indicates the version in which a single ConvNeXt V2 branch extracts clues from RGB images; "RGB + IoU" means we use the IoUDCE loss instead of the cross entropy loss; "RGB + IoU + UN" denotes the version which constructs the UN branches; "RGB + IoU + UN + NPP" adds the noise branch in which two-branch intermediate feature maps are concatenated directly. The results demonstrate that each component in MUN perfects our network and enhance its performance.

**Noiseprint++ deployment evaluation.** Resizing is one of the tampering types that Noiseprint++ can detect. As shown in Figure 3, we find that resizing can have a devastating effect on the noise map generated by Noiseprint++. So we

| Backbone | F1 | FLOPS (G) | Params (M) |
|---|---|---|---|
| ResNet-152 | 0.9326 | 123.988 | 91.233 |
| Swin-B | 0.9450 | 109.568 | 90.324 |
| ConvNeXt V2-B | 0.9519 | 94.204 | 91.139 |

Table 1: Backbone evaluation on Synthesized Dataset

| Variants | F1 | IoU | Acc | AUC |
|---|---|---|---|---|
| RGB | 0.9519 | 0.9156 | 0.9943 | 0.9985 |
| RGB+IoU | 0.9523 | 0.9161 | 0.9943 | 0.9985 |
| RGB+IoU+UN | 0.9525 | 0.9166 | 0.9943 | 0.9986 |
| RGB+IoU+UN+NPP | **0.9533** | **0.9170** | **0.9945** | **0.9989** |

Table 2: MUN ablation study on Synthesized Dataset

| Operation order | F1 | IoU | Acc | AUC |
|---|---|---|---|---|
| Resize | 0.9525 | 0.9166 | 0.9943 | 0.9986 |
| NPP after resize | 0.9521 | 0.9158 | 0.9942 | 0.9985 |
| Resize after NPP | **0.9533** | **0.9170** | **0.9945** | **0.9989** |

Table 3: Noiseprint++ (NPP) deployment evaluation on Synthesized Dataset

conduct $512 \times 512$ resize operation on RGB images, and we adopt Noiseprint++ after and before the $512 \times 512$ resize operation respectively, i.e., "NPP after resize" and "resize after NPP", to investigate the impacts. "Resize" denotes the version only with the resized RGB image input, and the versions of "NPP after resize" and "resize after NPP" concatenate RGB and Noiseprint++ feature maps directly without MMQ. As we can see from Table 3, "NPP after resize" has a negative effect on MUN, and "resize after NPP" leverages the information in the Noiseprint++ domain and boost the performance. Thus, we think that the correct way of deploying Noiseprint++ is to put it before the resize operation.

**Parameter setting of MMQ.** As shown in Table 4, the F1 score insceases by 0.07 % when MMQ has a single $1 \times 1$ max pooling stream. With more different kernel-sized max pooling streams, the performance can be further improved. MMQ can guide MUN to learn the correlation between multi-scale RGB features and Noiseprint++ features. By max pooling operation, some less important information can be ignored, and MUN pays more attention to the vital information of local features at different scales.

## Robustness Evaluation

In practice, one may disguise forged images with additional postprocessing. We consider three common postprocessing for robustness evaluation: (1) resizing forged images with two smaller ratio, (2) smoothing forged images with two different size kernels, and (3) compress forged images with two different quality factors. As shown in Table 5, it can be seen that MUN achieves consistently better performance, which indicates that MUN has good robustness.

(a) (b) (c)

Figure 3: (a) Original image. (b) Noiseprint++ map generated from original image. (c) Noiseprint++ map generated from resized image.

| Kernel size | F1 | IoU | Acc | AUC |
|---|---|---|---|---|
| None | 0.9533 | 0.9170 | 0.9945 | 0.9989 |
| 1 | 0.9540 | 0.9184 | 0.9947 | 0.9989 |
| 1, 5 | 0.9540 | 0.9183 | 0.9939 | 0.9989 |
| 1, 5, 11 | 0.9542 | 0.9186 | **0.9947** | 0.9989 |
| 1, 5, 11, 15 | **0.9543** | **0.9186** | 0.9946 | **0.9989** |

Table 4: MMQ parameter setting on Synthesized Dataset

| Distortions | MVSS-Net | MUN |
|---|---|---|
| no distortions | 0.814 | **0.885** |
| Resize(0.78×) | 0.799 (-0.015) | **0.874** (-0.011) |
| Resize(0.25×) | 0.700 (-0.114) | **0.835** (-0.050) |
| GaussianBlur(k=3) | 0.796 (-0.018) | **0.865** (-0.020) |
| GaussianBlur(k=15) | 0.761 (-0.053) | **0.813** (-0.072) |
| JPEGCompress(q=100) | 0.811 (-0.003) | **0.888** (+0.003) |
| JPEGCompress(q=50) | 0.786 (-0.028) | **0.818** (-0.067) |

Table 5: Robustness evaluation on IMD2020

## Comparison with State-of-the-art Methods

We compare MUN with recent SOTA works, including RGB-N (Zhou et al. 2018), ManTra-Net (Wu, AbdAlmageed, and Natarajan 2019), SPAN (Hu et al. 2020), MVSS-Net (Chen et al. 2021), PSCC-Net (Liu et al. 2022a), ObjectFormer (Wang et al. 2022), TANet (Shi, Chen, and Zhang 2023), TBFormer (Liu et al. 2023), HiFi (Guo et al. 2023), TruFor (Guillaro et al. 2023), CSR-Net (Zhang et al. 2024), NRL-Net (Zhu et al. 2024) and MGQFormer (Zeng et al. 2024). Table 6 shows the results of above-mentioned models on NIST16, CAISA v1.0, IMD2020, CocoGlide and Wild datasets, and the scores are borrowed from their original papers. It can be seen from Table 6, MUN works better on the majority of datasets. Because of the tremendous GPU memory usage of Noiseprint++ when processing ultra-high-resolution images in NIST16, we first seperate images into nine patches and then perform Noiseprint++ on each of them, which may result in the score decrease on NIST16. CocoGlide is generated using the GLIDE diffusion model, and the good performance on CocoGlide indicates that MUN is capable to deal with AI generated tampered images.

Visual comparisons are shown in Figure 4. In the 1st and

| Methods | NIST16 | CASIA v1.0 | IMD2020 | CocoGlide | Wild |
|---|---|---|---|---|---|
| RGB-N$^{CVPR'18}$ | 0.764 | 0.795 | - | - | - |
| ManTra-Net$^{CVPR'19}$ | 0.795 | 0.817 | 0.748 | 0.778 | 0.677 |
| SPAN$^{ECCV'20}$ | 0.840 | 0.797 | 0.750 | 0.475 | - |
| MVSS-Net$^{ICCV'21}$ | - | 0.815 | 0.814 | 0.654 | 0.768 |
| PSCC-Net$^{TCSVT'22}$ | 0.855 | 0.829 | 0.806 | 0.777 | 0.745 |
| ObjectFormer$^{CVPR'22}$ | 0.872 | 0.843 | 0.821 | - | - |
| TANet$^{TCSVT'23}$ | <u>0.898</u> | 0.853 | 0.849 | - | <u>0.832</u> |
| TBFormer$^{SPL'23}$ | 0.847 | 0.955 | 0.863 | 0.747 | 0.783 |
| HiFi$^{CVPR'23}$ | 0.869 | 0.866 | 0.834 | - | - |
| TruFor$^{CVPR'23}$ | 0.839 | 0.833 | 0.818 | 0.752 | - |
| CSR-Net$^{AAAI'24}$ | 0.883 | 0.881 | 0.854 | - | - |
| NRL-Net$^{AAAI'24}$ | **0.900** | 0.872 | 0.852 | - | - |
| MGQFormer$^{AAAI'24}$ | 0.862 | 0.886 | 0.883 | - | - |
| MUN | 0.857 | **0.967** | <u>0.885</u> | <u>0.811</u> | 0.805 |
| MUN* | 0.861 | <u>0.962</u> | **0.897** | **0.815** | **0.843** |

The bold entities denote the best results per column and the underlined ones denote the second best results. * denotes that the DNA data augmentation method is performed. AUC scores are reported.

Table 6: Comparison against the State-Of-The-Art methods

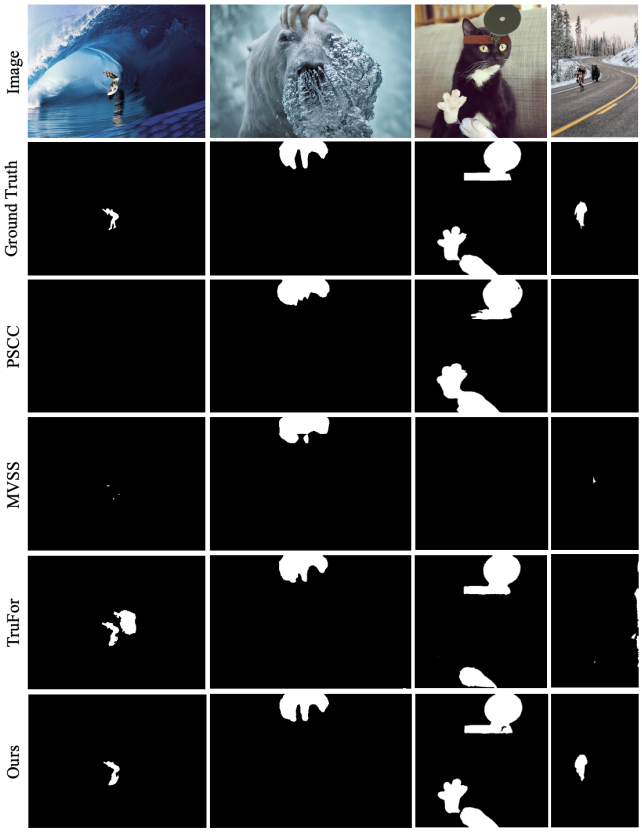


Figure 4: Visual comparison on IMD2020.

4th columns, some compared methods cannot detect the manipulated regions, MUN can accurately locate these regions. In the 2nd and 3rd columns, the compared methods cannot detect accurate boundaries or miss some forged regions, MUN has higher localizing accuracy and can extract the boundary of these tampered regions more precisely.

Besides, we evaluate the version of MUN trained with DNA, i.e., MUN*. As shown in the bottom line in Table 6. MUN* can achive better performance on the majority of datasets. The reason of the performance decline on CASIA v1.0 is that the distribution of CASIA v1.0 is similar to that of our training dataset, and DNA increases the gap between these two distributions. In general, DNA can enhance the performance of MUN and improve the generalization ability without fine-tuning it on every test set.

## Conclusion

We introduce an image forgery localization network named MUN. In M$^3$ encoder, two ConvNeXt V2 branches are acted as a multi-clue extractor to draw forged traces from both RGB and Noiseprint++ domains, and MMQ builds the correlation from prominent local RGB features to the noise domain. UN decoder can learn top-down and bottom-up hierarchical features, and reconstruct the predicted mask. The IoUDCE loss uses IoU to adjust the class weights dynamically, guiding networks to focus on intractable tampered regions. DNA narrows the gap between the RGB distribution of training images and universal images, improving the generalization ability of MUN. Experimental results verify that MUN outperforms the SOTA works. In the future, MUN can be enhanced by integrating additional clues, and further exploration could focus on identifying the type of noise distribution that best encompasses various image distributions, as well as examining its relationship with frequency domain information.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62102010, 62476013 and 62376265, in part by the Fundamental Research Funds for the Central Universities under Grant 3282023016, and in part by the Project of China North Industries Information Center: Image Acquisition and Processing Technology Adaptable to Multiple Environments and Digital Management Platform under Grant 20220100H0113.

## References

Bahrami, K.; Kot, A. C.; Li, L.; and Li, H. 2015. Blurred image splicing localization by exposing blur type inconsistency. *IEEE Transactions on Information Forensics and Security*, 10(5): 999–1009.

Bianchi, T.; De Rosa, A.; and Piva, A. 2011. Improved DCT coefficient analysis for forgery localization in JPEG images. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2444–2447.

Chen, X.; Dong, C.; Ji, J.; Cao, J.; and Li, X. 2021. Image Manipulation Detection by Multi-View Multi-Scale Supervision. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 14165–14173.

Cozzolino, D.; and Verdoliva, L. 2020. Noiseprint: A CNN-Based Camera Model Fingerprint. *IEEE Transactions on Information Forensics and Security*, 15: 144–159.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Dong, J.; Wang, W.; and Tan, T. 2013. CASIA Image Tampering Detection Evaluation Database. In *2013 IEEE China Summit and International Conference on Signal and Information Processing*, 422–426.

Guan, H.; Kozak, M.; Robertson, E.; Lee, Y.; Yates, A. N.; Delgado, A.; Zhou, D.; Kheyrkhah, T.; Smith, J.; and Fiscus, J. 2019. MFC Datasets: Large-Scale Benchmark Datasets for Media Forensic Challenge Evaluation. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 63–72.

Guillaro, F.; Cozzolino, D.; Sud, A.; Dufour, N.; and Verdoliva, L. 2023. TruFor: Leveraging All-Round Clues for Trustworthy Image Forgery Detection and Localization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20606–20615.

Guo, X.; Liu, X.; Ren, Z.; Grosz, S.; Masi, I.; and Liu, X. 2023. Hierarchical fine-grained image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3155–3165.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hu, X.; Zhang, Z.; Jiang, Z.; Chaudhuri, S.; Yang, Z.; and Nevatia, R. 2020. SPAN: Spatial Pyramid Attention Network for Image Manipulation Localization. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 312–328. Springer International Publishing.

Huh, M.; Liu, A.; Owens, A.; and Efros, A. A. 2018. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European conference on computer vision (ECCV)*, 101–117.

Jain, J.; Li, J.; Chiu, M. T.; Hassani, A.; Orlov, N.; and Shi, H. 2023. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2989–2998.

Korus, P.; and Huang, J. 2017. Multi-Scale Analysis Strategies in PRNU-Based Tampering Localization. *IEEE Transactions on Information Forensics and Security*, 12(4): 809–824.

Li, F.; Zhang, H.; Xu, H.; Liu, S.; Zhang, L.; Ni, L. M.; and Shum, H.-Y. 2023. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3041–3050.

Li, Y.; Hu, L.; Dong, L.; Wu, H.; Tian, J.; Zhou, J.; and Li, X. 2024. Transformer-Based Image Inpainting Detection via Label Decoupling and Constrained Adversarial Training. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(3): 1857–1872.

Li, Y.; and Zhou, J. 2019. Fast and Effective Image Copy-Move Forgery Detection via Hierarchical Feature Point Matching. *IEEE Transactions on Information Forensics and Security*, 14(5): 1307–1322.

Liu, X.; Liu, Y.; Chen, J.; and Liu, X. 2022a. PSCC-Net: Progressive Spatio-Channel Correlation Network for Image Manipulation Detection and Localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11): 7505–7517.

Liu, Y.; Guan, Q.; Zhao, X.; and Cao, Y. 2018. Image Forgery Localization based on Multi-Scale Convolutional Neural Networks. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '18, 85–90. New York, NY, USA: Association for Computing Machinery.

Liu, Y.; Lv, B.; Jin, X.; Chen, X.; and Zhang, X. 2023. TB-Former: Two-Branch Transformer for Image Forgery Localization. *IEEE Signal Processing Letters*, 30: 623–627.

Liu, Y.; Xia, C.; Zhu, X.; and Xu, S. 2022b. Two-Stage Copy-Move Forgery Detection With Self Deep Matching and Proposal SuperGlue. *IEEE Transactions on Image Processing*, 31: 541–555.

Liu, Y.; Zhu, X.; Zhao, X.; and Cao, Y. 2019. Adversarial Learning for Constrained Image Splicing Detection and Localization Based on Atrous Convolution. *IEEE Transactions on Information Forensics and Security*, 14(10): 2551–2566.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2022. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, 16784–16804.

Niloy, F. F.; Kumar Bhaumik, K.; and Woo, S. S. 2023. CFL-Net: Image Forgery Localization Using Contrastive Learning. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 4631–4640.

Niu, Y.; Tondi, B.; Zhao, Y.; Ni, R.; and Barni, M. 2021. Image Splicing Detection, Localization and Attribution via JPEG Primary Quantization Matrix Estimation and Clustering. *IEEE Transactions on Information Forensics and Security*, 16: 5397–5412.

Novozámský, A.; Mahdian, B.; and Saic, S. 2020. IMD2020: A Large-Scale Annotated Dataset Tailored for Detecting Manipulated Images. In *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 71–80.

Peng, B.; Wang, W.; Dong, J.; and Tan, T. 2016. Optimized 3D lighting environment estimation for image forgery detection. *IEEE Transactions on Information Forensics and Security*, 12(2): 479–494.

Peng, B.; Wang, W.; Dong, J.; and Tan, T. 2017. Image forensics based on planar contact constraints of 3d objects. *IEEE Transactions on Information Forensics and Security*, 13(2): 377–392.

Popescu, A. C.; and Farid, H. 2005. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10): 3948–3959.

Shi, Z.; Chen, H.; and Zhang, D. 2023. Transformer-Auxiliary Neural Networks for Image Manipulation Localization by Operator Inductions. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9): 4907–4920.

Tan, Y.; Li, Y.; Zeng, L.; Ye, J.; Wang, W.; and Li, X. 2023. Multi-scale Target-Aware Framework for Constrained Splicing Detection and Localization. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, 8790–8798. New York, NY, USA: Association for Computing Machinery.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, C.; Huang, Z.; Qi, S.; Yu, Y.; Shen, G.; and Zhang, Y. 2023. Shrinking the Semantic Gap: Spatial Pooling of Local Moment Invariants for Copy-Move Forgery Detection. *IEEE Transactions on Information Forensics and Security*, 18: 1064–1079.

Wang, J.; Wu, Z.; Chen, J.; Han, X.; Shrivastava, A.; Lim, S.-N.; and Jiang, Y.-G. 2022. ObjectFormer for Image Manipulation Detection and Localization. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2354–2363.

Wei, Y.; Ma, J.; Wang, Z.; Xiao, B.; and Zheng, W. 2022. Image splicing forgery detection by combining synthetic adversarial networks and hybrid dense U-net based on multiple spaces. *International Journal of Intelligent Systems*, 37(11): 8291–8308.

Weng, S.; Zhu, T.; Zhang, T.; and Zhang, C. 2024. UCM-Net: A U-Net-Like Tampered-Region-Related Framework for Copy-Move Forgery Detection. *IEEE Transactions on Multimedia*, 26: 750–763.

Woo, S.; Debnath, S.; Hu, R.; Chen, X.; Liu, Z.; Kweon, I. S.; and Xie, S. 2023. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16133–16142.

Wu, H.; and Zhou, J. 2022. IID-Net: Image Inpainting Detection Network via Neural Architecture Search and Attention. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3): 1172–1185.

Wu, Y.; AbdAlmageed, W.; and Natarajan, P. 2019. ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries With Anomalous Features. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9535–9544.

Xu, Y.; Irfan, M.; Fang, A.; and Zheng, J. 2023. Multi-scale Attention Network for Detection and Localization of Image Splicing Forgery. *IEEE Transactions on Instrumentation and Measurement*, 72: 1–15.

Yang, Z.; Liu, B.; Bi, X.; Xiao, B.; Li, W.; Wang, G.; and Gao, X. 2024. D-Net: A dual-encoder network for image splicing forgery detection and localization. *Pattern Recognition*, 155: 110727.

Zeng, K.; Cheng, R.; Tan, W.; and Yan, B. 2024. MGQ-Former: Mask-Guided Query-Based Transformer for Image Manipulation Localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 6944–6952.

Zhang, L.; Xu, M.; Li, D.; Du, J.; and Wang, R. 2024. CatmullRom Splines-Based Regression for Image Forgery Localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 7196–7204.

Zhou, B.; Zhao, H.; Puig, X.; Xiao, T.; Fidler, S.; Barriuso, A.; and Torralba, A. 2019. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127: 302–321.

Zhou, P.; Han, X.; Morariu, V. I.; and Davis, L. S. 2018. Learning Rich Features for Image Manipulation Detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1053–1061.

Zhu, J.; Li, D.; Fu, X.; Yang, G.; Huang, J.; Liu, A.; and Zha, Z.-J. 2024. Learning Discriminative Noise Guidance for Image Forgery Detection and Localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 7739–7747.

Zhuo, L.; Tan, S.; Li, B.; and Huang, J. 2022. Self-adversarial training incorporating forgery attention for image forgery localization. *IEEE Transactions on Information Forensics and Security*, 17: 819–834.