**Journal of Intelligent Computing and Networking**
https://www.ffspub.com/index.php/jicn/index
ISSN 3079-9228 (print)
ISSN 3107-2798 (online)
E-mail: jicn.office@ffspub.com

*Article*

# Forgery Localization Via Extracting Generic Features And Multiple Prior Fusion

Jiaqi Zhang[1,‡], Liqiong Jian[2,‡], Ziping Ma[3], Jinlin Ma[1,†], Xiaoshuai Huang[4,5†]
[1]*School of Computer Science and Engineering, North Minzu University, Yinchuan 750021, China*
[2]*Blood Center of Ningxia Hui Autonomous Region, Yinchuan 750011, China*
[3]*School of Mathematics and Information Science, North Minzu University, Yinchuan 750021, China*
[4]*Key Laboratory of Carcinogenesis and Translational Research (Ministry of Education/Beijing), Peking University Cancer Hospital & Institute, Beijing 100142, China*
[5]*Beijing Advanced Center of Cellular Homeostasis and Aging-Related Diseases, Biomedical Engineering Department, International Cancer Institute, Institute of Advanced Clinical Medicine, Health Science Center, Peking University, Beijing 100191, China*
[‡]*These authors contributed equally to this work.*
[†]*E-mails: majinlin@nmu.edu.cn (J. M.), hsx@hsc.pku.edu.cn (X. H.)*

*Received: April 8, 2026 / Revised: May 12, 2026 / Accepted: May 26, 2026 / Published online: May 31, 2026*

**Abstract:** Nowadays, it is easy to create fake images using image editing software, which poses many security risks. Many current deep learning approaches tend to mix multiple types of faked samples for training to improve the model's generality. This ignores the unique nature of the different forged image types. In this paper, we provide a new perspective on training strategies for the generality of forensic tasks. Our analysis and experiments suggest that, under the conditions of this study, copy-move samples may be more effective in learning generic features than splicing samples. To explore this generic feature, we trained using only copy-move samples and tested both splicing and copy-move samples. In addition, we observed that original images of different quality underwent the same forgery process and did not produce precisely the same forgery cues, even though they had the same semantic information. Therefore, in order to improve the generalisation of the method to different datasets, three copy-move datasets were created based on the different qualities of the original images. Instead of mixing the three copy-move datasets directly for training, we use a multi-prior fusion strategy for training. The effectiveness of our proposed method was demonstrated through experimental testing on the public datasets.

**Keywords:** Image Forensics; Image Forgery Localization; Copy-Move; Splicing; Multi-Prior Fusion
https://doi.org/10.64509/jicn.22.94

## 1 Introduction

With the development of social networks, images are increasingly widely used as an essential medium for information transmission. At the same time, it has become more accessible for people to edit the content of images. Some people can easily use it to create rumors, fake news, and other harmful influences. The task of image forensics is to detect these tampered images and locate the tampered areas to avoid the security risks posed by forged images.

Image local forgery refers to the use of image editing software (e.g. Photoshop) to replace the content of an area of the original image with new content to achieve the effect of changing the semantics of the original image. Splicing and copy-move are distinguished by the source of the tampered region content; the tampered content of a splicing image comes from another real image, while the tampered content of a copy-move image comes from its image. Due to the different sources of forgery content, the two forgery types generate unique cues that researchers target for tampering detection and localization. Detection of splicing forged images usually utilizes some difference features such as photo-response non-uniformity noise [1–4], lighting inconsistency [5–7], JPEG compression differences [8–10], EXIF metadata [11], etc. Since the tampered content of copy-move originates from itself and does not have these differences, the detection of copy-move usually looks for the existence of two regions in a pair of images with the same features [12–16]. The

† Corresponding author: Jinlin Ma, Xiaoshuai Huang
*Academic Editor: Xueyang Fu

above approach to forgery detection by the unique characteristics of one forgery type can be summarised as a one-to-one approach. This is shown in Figure 1(a). Researchers have focused more on one specific forgery type.

Recent developments in deep learning have led to an increasing number of researchers exploring a generalized deep model that is expected to be effective for multiple tamper types while studying individual tamper types [17–19]. Most approaches tend to mix all tamper types to train the model with the expectation that the deep learning model can learn the common tamper features, which is a blind training strategy, as shown in Figure 1(b). The researchers trained by mixing samples of multiple forgery types, expecting the model to work for multiple forgery type samples during the testing phase. Different types of tampered data have common tampering features and unique features. When we study a generic approach for multiple forgery types, we should make the deep model focus more on the common features rather than the idiosyncratic ones. Once the model focuses more on specific features, it will only work on a certain type of forgery data, which is not the desired goal of a generic approach.

Experimental analysis indicates that, in our setup, copy-move data appears more conducive than splicing data for deep neural networks to learn common features of both forgery types. When we use GroundTruth supervised model to learn forgery regions, the neural network pays more attention to the artifact cues left at the edges by the copy-move data tampering process. Because copy-move image tampering regions originate from their images and have the same properties as other regions. This feature is also valid for splicing detection because splicing has a similar snap-paste image content process as copy-move. Thus we only need to train the model by copy-move to be effective on both forgery types. Our method is shown in Figure 1(c). We train using only copy-move forgery samples, allowing the model to learn the generic features used for forgery localization, which works for both splicing and copy-move samples during testing.

In this paper, we only use copy-move data for training to explore the effectiveness of copy-move samples in providing generic features. Two broad approaches exist for the training process of deep learning methods currently studying image tampering detection and localization tasks. The first is to divide the existing public datasets into training samples and test samples [19–21]; the other one is to select some target datasets from other domains (e.g. MS-COCO [22]) to produce tampering datasets using the provided object masks [18, 23, 24]. We found that images of different qualities underwent the same forgery process and that the forgery features used for detection and localization were not identical, even though they had the same semantic content. However, the quality of the real-world tampered original images is uncertain. We made three copy-move datasets with different original image qualities to simulate real-world situations. It is worth noting that the original image source is the same for all three datasets. We use a multi-prior fusion (MPF) strategy inspired by meta-learning to better learn multiple prior knowledge. We use the current popular encoder-decoder structure as our network and demonstrate the effectiveness of our proposed method through test experiments on publicly available datasets.

The remainder of this paper is organized as follows. Section 2 summarizes related work. Section 3 introduces that copy-move data is better for deep network models to learn generic features than splicing data and our proposed training strategy of multi-prior fusion. Section 4 evaluates the effectiveness of our proposed method through some experiments. Section 5 discusses our method. Section 6 concludes this paper.

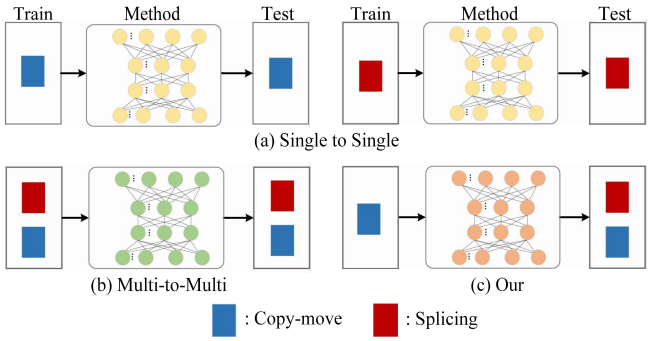



**Figure 1**: Illustration of the different methods of image forensics tasks. Where the different colored rectangle represent different samples of the type of forgery, the blue rectangle represent copy-move samples, and the red rectangle represent splicing samples. (a) Research for specific types of forgery. (b) Research for multiple forgery types. They are often trained directly with multiple types of forgery samples. (c) Our method uses only copy-move for training and tests both splicing and copy-move.

# 2 Related Work

With the development of science and technology, many methods of forgery have emerged. Meanwhile, researchers have never stopped exploring image forensic tasks. More and more methods have been used to detect various tampering methods with great results. Splicing and copy-move as traditional forgery methods are still in the attention of researchers today.

## 2.1 Research on Specific Types of Forgery

Many research approaches to splicing and copy-move forgery focus only on particular types of forgery. The main reason is that the different sources of tampered content give rise to their unique features, which researchers use for forgery detection. Since the tampered content of splicing comes from another image, this process gives rise to many discrepant features. Fu *et al.* [25] proposed a method for forgery detection localization using lateral chromatic aberration, offset optical components at different wavelengths leading to misalignment between color channels. Ferrara *et al.* [26] based on CFA features to measure the presence of artifacts of demosaicing at the local level and a new statistical model that allows deriving the tampering probability of each small image block without affecting the image quality. Pasquini *et al.* [10] proposed a novel forensic detector for JPEG compressed traces in images stored in uncompressed format, based on a theoretical analysis of the Benford-Fourier coefficients computed on the $8\times8$ block discrete cosine transform (DCT) domain. Cozzolino *et al.* [27] used PRNU as a camera device fingerprint and extracted this fingerprint for forgery detection by training a deep

learning model. Huh *et al*. [11] trained a Siamese network model using binary labels of 83-dimensional EXIF metadata to distinguish whether two image blocks are from the same image.

However, the tampered content of copy-move originates from its image, so more approaches expect to find regions where the two features are consistent. Cozzolino *et al*. [28] proposed an algorithm to accurately detect and localize copy-move forgeries based on a rotation-invariant feature that is computed intensively on the image. Emam *et al*. [29] proposed an efficient technique for detecting copy-move forgery under geometric transformation. Bi *et al*. [12, 30] proposed a novel and fast reflective offset-guided searching method for image copy-move forgery detection. Zhong *et al*. [13] proposed an end-to-end, multi-dimensional dense feature-connected deep neural network model that automatically learns feature correlations and searches for possible fake blocks through matching cues.

These methods for specifying a single type have achieved good results by exploiting the unique characteristics of the forgery type. The most obvious drawback, however, is the lack of generality in the model. At the same time, many researchers would like the proposed methods to handle a wider variety of forgery samples.

### 2.2 Research on Multiple Types of Forgery

In recent years, many approaches have used deep learning methods to explore the generality and generalization of image forensics tasks. Wu *et al*. [17] proposed a network model called Mantra-Net, which is also end-to-end and includes a feature extraction module and an anomaly localization module. First, the feature extraction module performs 385 image global manipulation type classification as a pre-training process. It is then trained with the anomaly localization module using multiple tamper types data. The training data includes four types: splicing, copy-move, removal, and enhancement. Zhou *et al*. [19] proposed a Faster R-CNN based method for tamper detection, including two branches, RGB stream and noise stream. It localizes the effect as a rectangular box. Hu *et al*. [18] proposed a feature pyramid network for tamper detection and localization. This architecture efficiently and effectively models the relationship between image patches at multiple scales by constructing a pyramid of local self-attention blocks. Chen *et al*. [23] proposed a two-branch structure including noise branch and edge branch, addressing both aspects by multi-view feature learning and multi-scale supervision. Since deep neural network models rely on the training of large amounts of data, the lack of artificially produced datasets for local image tampering has been a troubling problem. However, these methods tend to collect and produce datasets of various forged types. One common solution to the lack of datasets is to use datasets from object segmentation (e.g. MS-COCO) to generate tampered samples by algorithms [19, 31]. Such datasets provide the original RGB image and the binary mask of the objects in the image. The algorithm can generate a large amount of tampered data for training the model. For example, DEFACTO [32] generated over 200k forged images using forgeries automatically generated by MS-COCO. Most methods then tend to mix the multiple forgery datasets produced together for training. This process is blind, and we cannot tell which cues the deep model utilizes.

Theoretically, if we want a model with generality, it is expected that the deep neural network is more focused on generic features. We found through analysis and experimentation that copy-move is more conducive to deep models learning generic features than splicing samples. In addition, it is practical to use the algorithm to generate fake datasets for training purposes from some real datasets. However, the original images originate from the same dataset and have similar quality. The actual tampering process is unpredictable. We need to provide data that is more realistic in order for the model to have generalization capability.

## 3 Proposed Approach

This section describes and analyzes how our proposed copy-move is more conducive to deep neural network models learning generic features than splicing. Such generic features work on both copy-move and splicing test data. In addition, we present the design ideas and training procedures of the multi-prior fusion strategy we use.

### 3.1 The Model Learns Generic Features Through Copy-Move

Historically, the research concepts on copy-move forgery and splicing forgery have been different. Since splicing's tampering comes from another image, this will introduce features that differ from other original regions. So the research on splicing forgery is to find reliable differential features for detection [1, 11]. These differential features do not apply to copy-move forgery, and researchers are more concerned with finding regions in the copy-move data where two features are consistent. Therefore, in many cases, there are two areas in the location mask for copy-move forgery [12, 13].

However, now we expect to locate the forgery area for copy-move and splicing. We conduct a set of forgery localization experiments with the DEFACTO [32] dataset. We randomly select 5500 splicing sample data and 5500 copy-move sample data in DEFACTO. For both types of forgery, 5000 images are divided as the training set, and 500 images are used as the test set. We use a simple encoder-decoder structure, where the encoder uses a VGG-16 [33] network, and the decoder uses a combination of up-sampling and convolution. Train separately using the training set divided by the two fake types to get two models. For analysis purposes, $M_A$ represents a model trained using only splicing samples. $M_B$ represents a model trained using only copy-move data. We then tested the two models on the two test sets, respectively.

We randomly selected 500 test data of two forgery types, downscaled the spatial features of these test samples by t-SNE, and visualized the distribution of spatial features of these test samples in the shallow, middle and deep layers of the model. Also, we output the final prediction map for each test sample and calculate the *F1* metric with GroundTruth. We set the color shades for each sample visualization according to the metrics. The darker the color means the better the final prediction. The detailed results are shown in

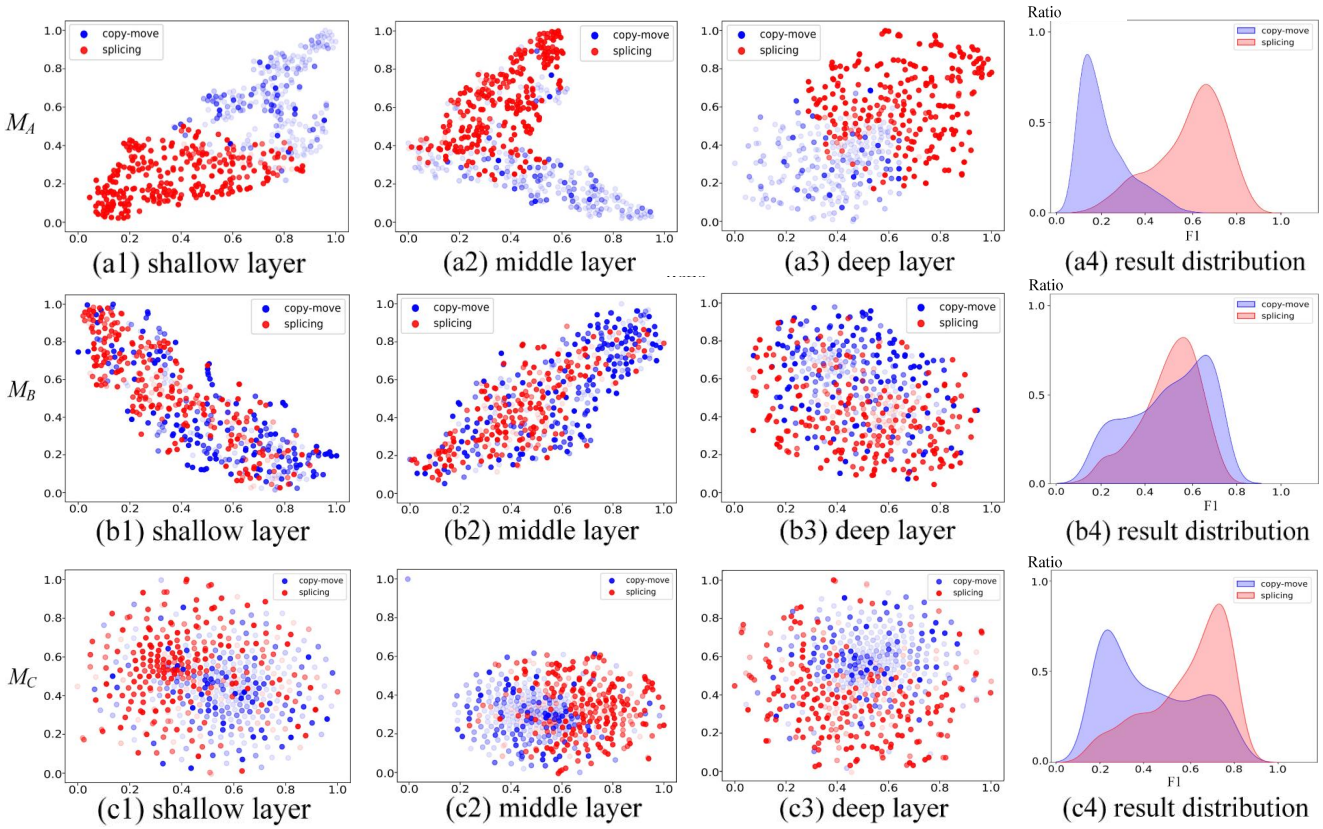



**Figure 2**: Two forgery types of test data are visualized in t-SNE space features of two separately trained models. (a1), (a2), (a3) are from $M_A$, (b1), (b2), (b3) are from $M_B$. We show the feature visualization in the shallow, middle, and deep layers of the model, respectively. (a1), (b1) show shallow layer features, (a2), (b2) show middle layer features and (a3), (b3) show deep layer features. In addition, we control the color shades according to the *F1* metric of the final prediction map for each sample. The darker the color, the better the result. Best viewed in color.

Figure 2(a1)–(a3),(b1)–(b3). In contrast to the traditional focus on observing t-SNE results, we do not aim to classify splicing and copy-move forgeries. Because we train the models using a single forgery type data, we are more concerned with the ability of both models to extract generic features. Using the feature visualization plots of the features of both forgery test data, we can observe that the feature distribution of $M_B$ is similar for both splicing and copy-move test data. However, the feature distributions of $M_A$ for both splicing and copy-move test data are significantly segregated. That is, only $M_B$ will treat the two forgery types of test data as one type, and the features it extracts by copy-move for forgery localization are common.

On the other hand, we can observe the final prediction by the shade of color. The results show that $M_B$ effectively predicts the final prediction for both types of falsified data. In contrast, $M_A$ was not as effective in predicting the copy-move test data. To see more clearly the difference in the predictive effect of $M_A$ and $M_B$ on the two types of falsified samples, we calculated the *F1* metric for the final predictions of all the test samples. The distribution of the effects of these test samples is then shown. These quantitative distributions, visually detailed in Figure 2(a4),(b4), further confirm the generalization advantage of $M_B$. In addition, when splicing and copy-move samples are mixed together directly, the performance of the directly mixed model is different for the two test samples, as copy-move does not have exactly the same differential features as splicing. We trained the model using both copy-move and splicing forged samples without changing the model structure and hyperparameters. We refer to this model as $M_C$. As demonstrated in Figure 2(c1)–(c4), this mixed model fails to unify the feature spaces, explicitly confirming the limitations of direct mixing. Examples of the final predicted masks of the two models for the splicing and copy-move test samples are shown in Figure 3(c), (d). These visual results are directly benchmarked against the raw inputs and GroundTruths provided in Figure 3(a),(b). We can observe that $M_B$ has better results for both forgery types of samples. In this way, we trained using only copy-move forgery samples, and the model learned the common features of splicing and copy-move forgery localization.

## 3.2 Why The Features Learned By Copy-Move Are Generic?

Both splicing and copy-move have a step of pasting the contents of the tampered area during the tampering process. This process causes a difference between the boundary of the forged area and the natural boundary of the normal camera shot. In fact, the manipulation would break the natural statistical information of the image in the manipulated boundary region [34, 35]. The positioning of a forged copy-move sample relies more on this clue. In contrast to previous approaches that only study a single forgery type of copy-move, we use

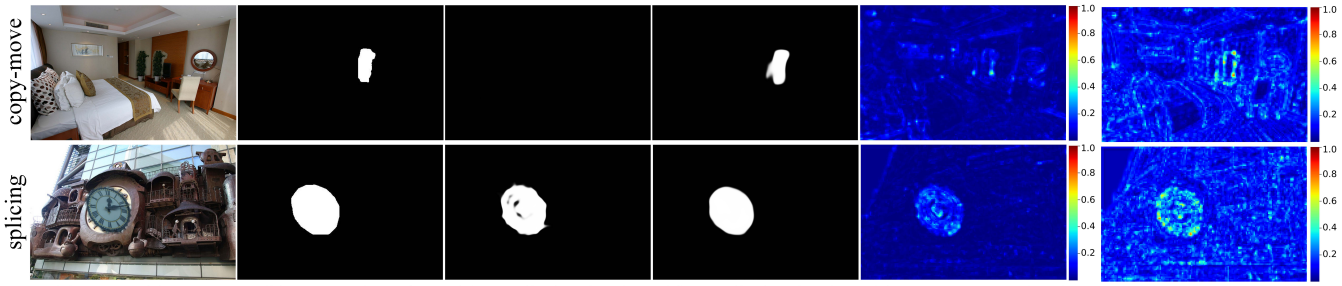



**Figure 3**: Visualisation of two forged test samples. (a) faked images. (b) GroundTruth. (c) prediction results of $M_A$ (trained with splicing samples only). (d) prediction results of $M_B$ (trained with copy-move samples only). (e) Feature map visualization of $M_A$. (f) Feature map visualization of $M_B$.

a strongly supervised approach that allows the model to localize on the forged region rather than on two regions of the copy-move sample with consistent features. In addition, the copy-move sample forgery is derived from its image. The image properties (e.g. compression rate, noise distribution) are the same on both sides of the forgery boundary. Once we use a strongly supervised approach to make the deep network model locate only the forged region, the deep neural network pays more attention to the cues on the forged boundary. However, the source of the forged content of the splicing sample is another image, which introduces more inconsistent features that the copy-move sample does not have as cues.

To show more clearly that the forgery detection cues utilized by copy-move are the edges of the forged regions, we visualize the shallow feature maps of the encoders of the trained $M_A$ and $M_B$ in the form of heat maps. Figure 3(e), (f) show the heat map visualization of the feature map. We can observe that the model trained using only copy-move samples focuses mainly on the edges of the forged regions for both forgery types of test data, which is consistent with our analysis.

## 3.3 The Multi-Prior Fusion Strategy

For the image forgery detection task, the researchers are equally concerned with the generalization ability of the deep neural network model. That is, whether the test data is equally valid once it comes from a dataset that has not been involved in the training. This situation is more in line with real-world requirements, where we cannot determine the origin of real-world forged images. The current number of hand-crafted public datasets is still insufficient. A common approach is to use algorithms for object-segmented datasets to generate tampered samples. However, the original images in the dataset have similar image quality to each other. We conducted the following experiments to explore the effect of image quality.

We parse the validation set of COCO2017 to obtain the corresponding single-object binary masks, discarding those that are too large and too small. We randomly select 4000 images and the corresponding masks of a certain object. We make three copy-move datasets using 4000 images, called $Set_{\text{ori}}$, $Set_{\text{jpeg}}$, and $Set_{\text{blur}}$. $Set_{\text{ori}}$ is direct tampering without further processing, and $Set_{\text{jpeg}}$ is JPEG compression of the original image followed by tampering, and $Set_{\text{blur}}$ is blurring of the original image followed by tampering. The number of all three datasets is 4000 forged images with similar semantic information. We divide the training and test sets in a 9:1 ratio for each dataset. We train a model using each copy-move dataset separately. We choose the popular encoder-decoder network architecture, where the encoder uses VGG-16, and the decoder uses a combination of upsampling and convolutional structure. This gives us three models $Model_{\text{ori}}$, $Model_{\text{jpeg}}$, $Model_{\text{blur}}$. The three test sets were tested on each model, and the results are shown at the top of Table 1.

**Table 1**: Results on the three copy-move test sets in terms of *F1* and *MCC*.

| Methods | $Set_{\text{ori}}$ | | $Set_{\text{jpeg}}$ | | $Set_{\text{blur}}$ | |
|---|---|---|---|---|---|---|
| | *F1* | *MCC* | *F1* | *MCC* | *F1* | *MCC* |
| $Model_{\text{ori}}$ | **0.792** | **0.786** | 0.504 | 0.542 | 0.771 | 0.785 |
| $Model_{\text{jpeg}}$ | 0.412 | 0.428 | **0.741** | **0.756** | 0.583 | 0.572 |
| $Model_{\text{blur}}$ | 0.245 | 0.248 | 0.314 | 0.309 | **0.826** | **0.842** |
| Model (Mixed) | 0.523 | 0.564 | 0.746 | 0.765 | 0.782 | 0.799 |
| Model (MPF) | **0.759** | **0.766** | **0.893** | **0.896** | **0.901** | **0.906** |

The results show that even though the semantic content of the original images is similar, the models appear different during testing when there is a difference in the image quality between the training and test data. This non-semantic information can impact the learning process of deep neural networks. This also suggests that original images of different quality undergo the same forgery process and that the features used for forgery detection and localization are different, even though they have the same semantic content. However, the real world cannot know the original image source of the forged image in advance, and a simple and direct way to obtain more a priori knowledge is to train all three copy-move data. Directly mixing all datasets may interfere with learning; MPF aims to address this by accumulating loss across datasets before back-propagation. Inspired by meta-learning, we designed a multi-prior fusion (MPF) strategy to accumulate losses from different datasets before back-propagation, which differs from ordinary mixed multi-source training. Our model framework is shown in Figure 4.

Specifically, we constructed a mini-batch using samples from a single type of dataset in each iteration and calculated the corresponding loss by forwarding propagation. Then we did not rush to propagate the loss backward but continued to

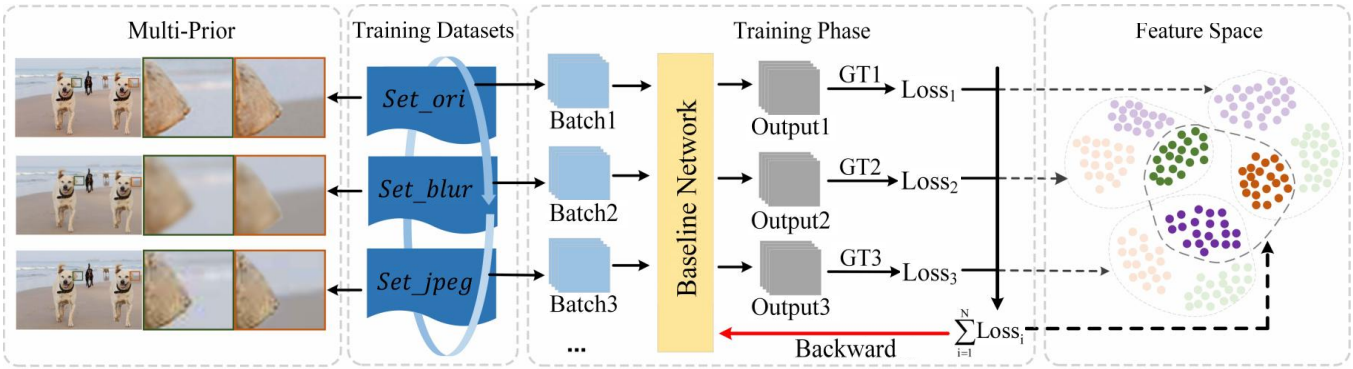



**Figure 4**: The overall framework of our method. The multiple prior fusion strategy is that we do not rush to back-propagate the gradient after calculating the loss for a single batch, but accumulate the loss for three batches before back-propagation. The three batches are sampled from different copy-move datasets.

iterate the mini-batch for another data type to obtain the corresponding loss. When each data type had undergone forward propagation, the respective losses were accumulated, and the whole gradient was back-propagated. We then repeated this alternating training strategy. Each reverse update of our model weights was a contribution containing multiple data types with the same mini-batch size for each data type. We used the cross-entropy loss function in the training as follows:

$$L_{CE} = -\sum_{i=1}^{N}\sum_{j=1}^{M} \lambda^i y_j^i \log(p_j^i) \tag{1}$$

where $N$ denotes the number of different types of tampering training sets, $M$ means the number of image pixels, $p_j^i$ refers to the prediction probability of the $j$-th pixel on the $i$-th dataset, and $y_j^i$ denotes the corresponding label of the $j$-th pixel on the $i$-th dataset. $\lambda^i$ denotes the loss weight on the $i$-th dataset and we set $\lambda^i = 1$.

The lower part of Table 1 shows the effect of direct mixed training and using the multi-prior fusion strategy on the three test sets. They are trained using the three copy-move training sets described above. Direct mix training combines the three copy-move datasets mentioned above into one dataset. Then each mini-batch is randomly sampled for the mixed dataset to train the network. The network architecture and hyperparameters are essentially the same, and it can be seen that the multi-prior fusion (MPF) strategy outperforms the mixed training strategy on all three data sets. It is worth noting that the MPF strategy does not introduce an additional computational cost in the testing phase of the model. Both the training and test samples here are generated by our algorithm. We also validate the performance of the two training strategies on some publicly available datasets in Section 4.3.

# 4 Experimental Analysis

In this section, we discuss the effectiveness of our proposed training using only copy-move data and the multi-prior fusion strategy. Our test data includes copy-move and splicing forgery samples. Our training set includes only copy-move forgery data produced by ourselves. For the test set, we prefer to be able to verify that the method has some effect on unseen datasets as well, so we chose four commonly used public datasets, including Coverage [36], CASIAv1 [37], CASIAv2 [37], and NIST [38].

## 4.1 Datasets and Evaluation Metrics

Due to the small number of available artificial public datasets and the data distribution of the same public dataset tends to be similar, we made copy-move dataset with COCO dataset and corresponding masks in order to reflect the method with some generality and generalization ability. We parsed the validation set of COCO2017 to generate the object masks corresponding to the images, removing those images with too large regions (>50%) and too small regions (<5%). From the remaining images, 4000 original images and the corresponding mask of a region were randomly selected. Four thousand images were then subjected to JPEG compression and Gaussian blurring to simulate the phenomenon of different image properties in reality, so that we obtained three datasets with 4000 images each, and the semantic information of the three datasets was basically the same. We used each dataset image and mask to make copy-move dataset, so that we got three copy-move datasets for training. The specific production process is as follows.

We first generate a mask of the fake object according to the real object mask.

$$M_{fake} = T(Box(mask)) \circledast mask_2 \tag{2}$$

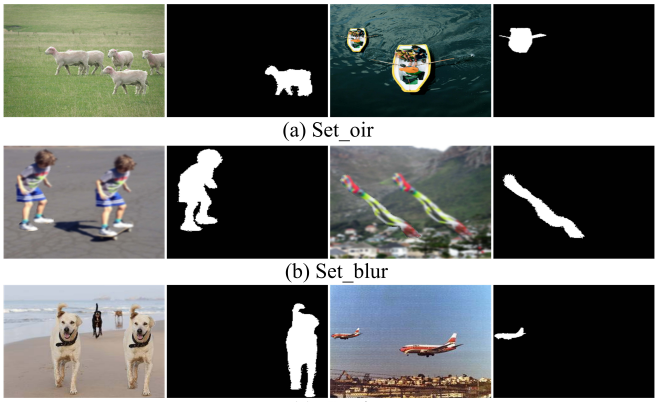
Then we generate the forged image.

$$\begin{aligned} I_{fake} = &(I_{real} \odot (1 - M_{fake})) \\ &+ T(Box(I_{real} \odot mask)) \circledast mask_2 \end{aligned} \tag{3}$$

Where $M_{fake}$ denotes the binary mask corresponding to the forged image. $I_{fake}$ refers to the final tampered image. $I_{real}$ denotes the COCO real original image, and $mask$ is its corresponding binary map of a certain area. $mask_2$ represents an image that is the same size as the original image, with all pixel values at 0. $Box$ denotes the bounding box that calculates the maximum outline of the region and crops it. Try to crop out the redundant areas with a pixel value of 0. $T()$ denotes a random transformation of size. $\circledast$ indicates a random position paste. $T()$ and $\circledast$ are operating exactly the same in

Equation 2 and Equation 3, and randomness is the process for each forged image.

We made three copy-move training sets using 4000 original images, and each training set includes about 8000 forged images. We call them $Set_{\text{ori}}$, $Set_{\text{blur}}$, $Set_{\text{jpeg}}$, respectively. Due to the tamper size and position transformation, multiple tampered images can be made from one original image. Some copy-move samples are shown in Figure 5. The difference between the three copy-move datasets is that the global processing of the original images is different. This includes no processing, JPEG compression with random QF values, and Gaussian blurring, expecting to obtain images of different quality, and then image local content tampering is performed.



(a) Set_oir

(b) Set_blur

(c) Set_jpeg

**Figure 5**: Some sample examples from the three copy-move datasets we produced. (a) direct forgery of the original image. (b) random Gaussian blurring of the original image followed by forgery. (c) random JPEG compression of the image followed by forgery.

Our training set includes only the copy-move samples we produced. To demonstrate that training with only copy-move forgery samples would allow the neural network model to learn a generic common feature, our test data included samples of copy-move and splicing forgery types. In addition, to verify that our method has some generalization, we use publicly available forgery datasets for testing. Table 2 shows some information about the test set.

**Table 2**: Information on the dataset tested in our experiments.

| Datasets | Size | Num. | Format |
|---|---|---|---|
| Coverage [36] | 489×380 | 100 | TIFF |
| CASIAv1 [37] | 384×256 | 921 | JPEG |
| CASIAv2 [37] | 473×322 | 5123 | JPEG, TIFF |
| NIST [38] | 3561×2516 | 380 | JPEG |

Forged localisation is essentially a dichotomous classification of each pixel. The evaluation metric we use relies on the following four values. TP and TN denote the number of positive and negative pixels declared correct respectively, FN is the number of positive pixels detected as negative and FP denotes the number of negative pixels detected as positive. We use the *F1* and Matthews correlation coefficient (*MCC*) as evaluation indicators. The calculations are as follows.

$$F1 = \frac{2TP}{2TP+FN+FP} \tag{4}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{5}$$

## 4.2 Experimental Setup

We are more interested in whether training with copy-move samples alone allows the network to learn generic features and whether the network is equally effective on unseen datasets. Our neural network architecture uses an encoder-decoder structure, which is by far the most commonly used network structure. So we are using the most conventional network model. Among the encoders we can use structures such as VGG-16, ResNet, Vision-Transformer, etc. In the following experiments, the encoder of our model uses the VGG-16 network structure if not otherwise specified. For the decoder we use a combination of multilayer upsampling and convolution. We used PyTorch to implement our proposed model. The input size of our training samples is 256×256. The initial learning rate is set to 0.0001 with 10% decrement every 10 epochs, mini-batch size of 4, using Adam optimizer.

## 4.3 Ablation Studies

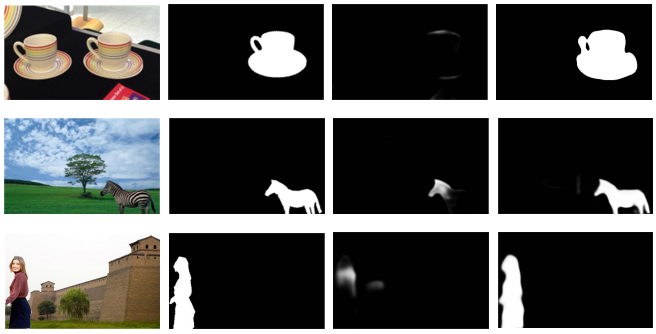
Before comparing with other methods, we discuss our proposal that it is easier to obtain generic features using copy-move data training only and the effectiveness of our use of multiple prior fusion strategies. We conducted a series of ablation experiments. First, we made 4000 splicing samples using the same 4000 original images, following the procedure for making falsified data introduced in Section 4.1. We ensured that the only difference compared to copy-move was that the forgery area of the splicing sample came from the contents of random other images. This ensures that the production process and the original image are exactly the same as the copy-move sample. We trained three models using VGG-16 as the encoder. The difference between the three models is the different type composition of the training data. The training data type composition was 4000 splicing samples, 4000 copy-move samples, and 2000 splicing and 2000 copy-move samples mixed, respectively. The three models were compared on the Coverage and CASIAv1 datasets for forged localization performance. Detailed experimental data are shown at the top of Table 3. We can see that the model trained using only copy-move samples is better at forgery localization than the other two models. When two forgery types are mixed together, the model does not work as well as when only one type of forgery sample is trained. This is most likely due to the unique differential features of splicing and the interference between the common features of the two types. If we want the model to capture the common features of both forgery types so that the model is generic, training with copy-move samples only is a better option.

We next validate the effectiveness of our proposed multiple prior fusion strategy. The three copy-move datasets we produced were trained in two ways, mixed training and training using the strategy of MPF. We then show the objective metric results for both strategies on the public dataset at the bottom of Table 3. We can see that when using the training strategy of MPF, there is an improvement of about 10% in the *F1* metric over direct hybrid

training on the Coverage dataset. There is also a 4% improvement on CASIAv1. This demonstrates the effectiveness of our use of a multi-prior fusion strategy. Also, we show the qualitative results for the two training strategies in Figure 6. The results show that our multi-prior fusion strategy worked better than direct mixing training.

**Table 3**: Results of the ablation experiment. The metric used is *F1*.

| Model | Type | Num. | Coverage | CASIAv1 |
|---|---|---|---|---|
| Model_A | splicing | 4000 | 0.392 | 0.321 |
| Model_B | spli.&copy. | 4000 | 0.378 | 0.293 |
| Model_C | copy-move | 4000 | **0.455** | **0.332** |
| Model (Mixed) | copy-move | 24000 | 0.356 | 0.325 |
| Model (MPF) | copy-move | 24000 | **0.467** | **0.363** |



(a) Forged image (b) GroundTruth (c) Prediction (Mixed) (d) Prediction (MPF)

**Figure 6**: Quantitative results for each version of our model. (a) the forged images. (b) GroundTruth. (c) prediction results of the mixed training model. (d) prediction results of a multi-prior fusion strategy training model.

The model structure we use is an encoder-decoder structure, where the encoder structure can use some of the common network structures available today. We conducted experiments where the encoder part was replaced to explore the effect of MPF with different network structures. Our encoder structures used DenseNet-169 [39], DenseNet-201 [39], ResNet-50 [40], MixFormer [41], and VGG-16 [33], respectively. The results are shown in Figure 7. Our MPF strategy has some enhancement effects using different model structures. Both training strategies achieved only comparable results on the DenseNet-169 model. As the network structure deepened, the MPF results on the DenseNet-201 model outperformed the mixed training strategy. It is worth noting that this does not introduce additional calculation costs in the testing phase.

## 4.4 Comparison With Other Methods

To show that our proposed training with only copy-move samples allows the model to learn generic features and the effectiveness of using the MPF training strategy, we chose 9 methods to compare with our model in pixel-level forgery localization performance, including 3 traditional and 6 deep learning methods. All deep learning methods were not fine-tuned. The main comparison methods are described below.

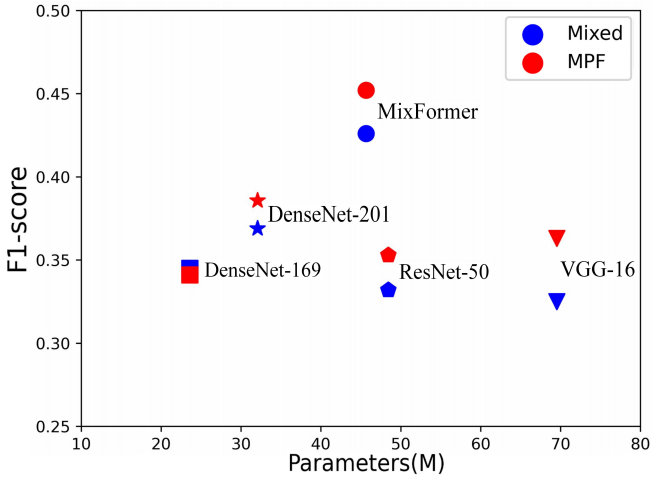



**Figure 7**: A comparison of the performance of different network structures using direct mixed training and MPF training strategies.

ELA [42]: An error-level analysis method designed to find differences between tampered and real regions by varying the quality of JPEG compression.

CFA [26]: A CFA-based method estimates the intra-camera CFA pattern for each patch in the image and segments the regions with abnormal CFA features into manipulated regions.

NOI [3]: A noise inconsistency method where the local noise of the manipulated region is not consistent with the real region.

MFCN [43]: A two-branch full convolutional neural network using both mask and edge.

EXIF [11]: A Siamese network structure, supervised using the EXIF information of the images, aims to determine whether two image patches come from the same imaging pipeline. Forgery localization is performed using Mean Shift.

ManTra-Net [17]: An end-to-end network that performs both detection and localization without extra preprocessing and postprocessing.

Noiseprint [27]: Image forgery detection and localization by Siamese network architecture to extract camera model fingerprints.

FS [44]: A CNN-based Siamese network to determine whether a pair of image patches contains the same or different forensic traces, i.e. source camera model and processing history.

MVSS-Net [23]: A two-branch structure including noisy branches and edge branches addresses both aspects through multi-view feature learning and multi-scale supervision.

These methods have not seen data from the test set during training. In recent years, Vision-Transformer has also proven to be equally powerful in computer vision. To demonstrate the validity of our two proposed ideas, we choose two popular methods, CNN and Vision-Transformer, for the model's encoder. The two structures are VGG-16 and MixFormer, respectively.

As can be seen by the results in Table 4, we show the performance of all models for pixel-level localization on four public datasets. We give results for both *F1* and *MCC* metrics. In addition, we give the ranks in parentheses after the

resulting values. We have marked the two best methods in red and the others in blue. With no fine-tuning of all methods, our model achieves better results regardless of whether we use a CNN or Vision-Transformer architecture for the encoder part of our model. Specifically, when we use VGG-16 as our encoder structure, our model achieves essentially the 2nd best results. When we use MixFormer as our encoder structure, our model achieves the top 2 results. This demonstrates that by training a neural network model using only copy-move samples, the model can learn generic features, even if the model has not seen these test datasets. The original images for our training set were taken from 4000 images, and we globally processed the original images to simulate the diverse image quality in the real world. We use a multi-prior fusion strategy for training, which has been experimentally shown to achieve better results. A possible reason for relatively lower performance on the NIST dataset may be the difference in image quality between our COCO-based training images and the high-quality NIST images; further investigation is needed. In addition, the copy-move dataset we produced was based on the original images, and we could have done some processing on the original images to simulate different quality images in the real world, but this would normally have reduced the quality of the images. We are unable to improve the quality of the original images. In our analysis, the same tampering process was experienced in images of different quality, and the features used for forgery detection localization were different, even though the semantic content of their forged images was identical. That said, it is theoretically more efficient to have higher quality raw images available for us to use for the training set.

Among the methods, EXIF, Noiseprint, and FS are based on camera model attribute features for manipulation detection and localization. There will be limitations in testing on some public datasets. The different features extracted by these methods are fragile and highly susceptible to corruption. Once the test sample is a low-resolution image, it is often invalid. From Table 4 it can be seen that for CASIAv2 and CASIAv1, these low-resolution test sets do not work well. For high-resolution images, the results are better. Our two models perform significantly better than these methods on these low-resolution datasets, at least 10% better on both *F1* and *MCC* metrics.

Our models, Mantra-Net, MFCN, and MVSS-Net, all use faked datasets during training, and none of the models have seen a public test set. On the other hand, these models are all end-to-end models at the pixel level. Specifically, Mantra-Net had 64k samples for training, MFCN had 35k samples for training, and MVSS-Net had 84k samples for training. The training samples for these methods include a variety of forgery types. However, we only use copy-move samples for training. This is a total of 24k samples from 4000 original images. Our model demonstrated competitive performance across the four test sets compared to the baseline methods.

The qualitative results of our model and MFCN, Mantra-Net, MVSS-Net on Coverage, CASIAv2, CASIAv1, and NIST datasets are shown in Figure 8. Our model is also effective in splicing forged images even if it is trained using only the copy-move data we produced. This is mainly because the model learns the discrepancy features between the edges of the forged regions and the edges of the actual regions. It is also useful for splicing forgery samples.

## 4.5 Robustness Analysis

We chose three post-processing methods of attack as a way to perform a robustness analysis of our model and further validated the effectiveness and robustness of our model. The three post-processing methods are JPEG compression, Gaussian blur, and Gaussian noise. We choose the NIST [38] dataset for testing. The specific results are shown in Figure 9, where we can see that after different post-processing methods, our model has a stable performance compared to other models. In our experiments, the performance of our method declines gradually under increasing post-processing attacks, indicating relative robustness compared to other methods tested. This is mainly because these attacks interfere with our model. However, the performance degradation of our model is smooth compared to the other three methods, whereas the other methods show a rapid degradation. For the JPEG attack, the performance of Mantra-Net and MVSS-Net showed a rapid degradation once JPEG compression of the test samples had occurred. In particular, for the Gaussian blur attack, Mantra-Net showed a sharp drop in performance from close to that of our method.

**Table 4**: The pixel-level localization performance of our proposed model is compared with existing methods on different datasets. The metrics are *F1* and Matthews correlation coefficient (*MCC*).

| Methods | Coverage | | CASIAv2 | | CASIAv1 | | NIST | |
|---|---|---|---|---|---|---|---|---|
| | *F1* | *MCC* | *F1* | *MCC* | *F1* | *MCC* | *F1* | *MCC* |
| ELA [42] | 0.273 (9) | 0.172 (11) | 0.192 (11) | 0.180 (7) | 0.213 (9) | 0.214 (7) | 0.236 (10) | 0.192 (8) |
| CFA [26] | 0.312 (7) | 0.190 (10) | 0.205 (8) | 0.108 (10) | 0.212 (10) | 0.132 (11) | 0.174 (11) | 0.103 (11) |
| NOI [3] | 0.269 (10) | 0.207 (8) | 0.232 (5) | 0.187 (6) | 0.263 (5) | 0.202 (9) | 0.285 (8) | 0.212 (6) |
| MFCN [43] | 0.363 (4) | 0.302 (5) | 0.215 (7) | 0.180 (7) | 0.245 (6) | 0.215 (6) | 0.242 (9) | 0.184 (9) |
| EXIF [11] | 0.223 (11) | 0.216 (7) | 0.197 (10) | 0.127 (9) | 0.204 (11) | 0.154 (10) | 0.315 (6) | 0.232 (5) |
| ManTra-Net [17] | 0.435 (3) | 0.413 (3) | 0.238 (4) | 0.268 (3) | 0.329 (3) | 0.297 (2) | 0.327 (5) | 0.193 (7) |
| Noiseprint [27] | 0.334 (5) | 0.306 (4) | 0.201 (9) | 0.237 (5) | 0.215 (8) | 0.205 (8) | **0.397 (1)** | **0.387 (1)** |
| FS [44] | 0.321 (6) | 0.299 (6) | 0.230 (6) | 0.254 (4) | 0.227 (7) | 0.267 (4) | 0.363 (3) | 0.361 (4) |
| MVSS-Net [23] | 0.282 (8) | 0.198 (9) | 0.249 (3) | 0.173 (8) | 0.320 (4) | 0.219 (5) | 0.289 (7) | 0.178 (10) |
| Proposed (VGG-16) | 0.465 (2) | 0.426 (2) | 0.315 (2) | 0.309 (2) | 0.363 (2) | 0.283 (3) | 0.357 (4) | 0.365 (3) |
| Proposed (MixFormer) | **0.472 (1)** | **0.476 (1)** | **0.331 (1)** | **0.329 (1)** | **0.453 (1)** | **0.425 (1)** | 0.382 (2) | 0.367 (2) |

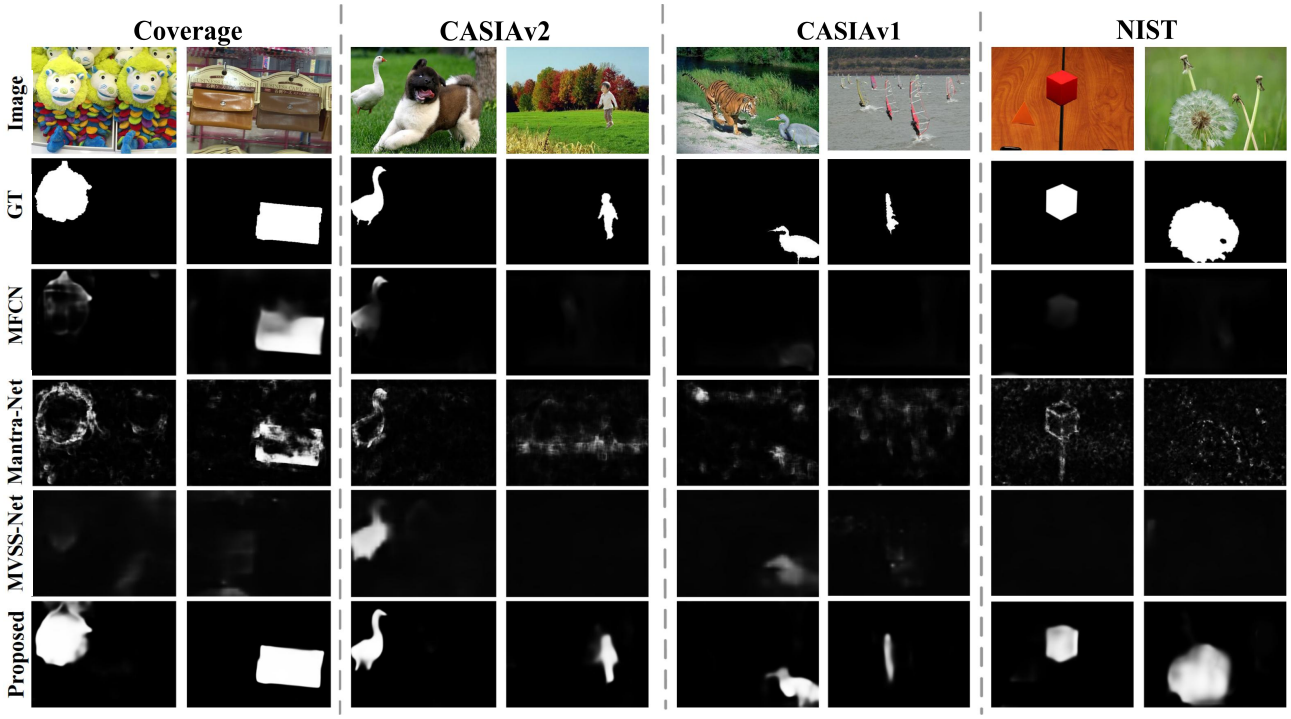



**Figure 8**: Qualitative results of proposed model and MFCN, Mantra-Net, MVSS-Net on Coverage, CASIAv2, CASIAv1, NIST datasets. From top to bottom are the tampered images, GroundTruth, the prediction of MFCN, Mantra-Net, MVSS-Net, and Proposed.

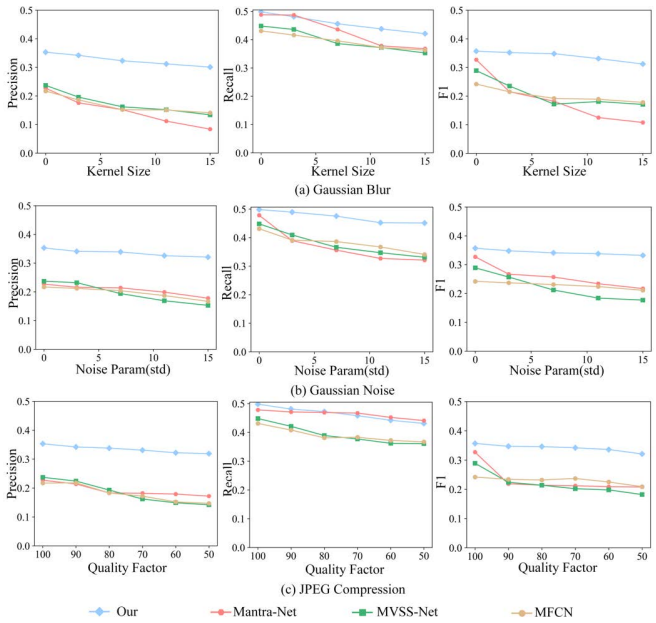



**Figure 9**: Comparing the robustness of models for different post-processing methods. The first, second and third columns plot Precision, Recall, and F1-score, respectively. Rows (a), (b), and (c) show the results under Gaussian blur, Gaussian noise, and JPEG compression, respectively.

# 5 Discussion

When we trained using only copy-move, we achieved comparable or better performance than other methods of training multiple forged sample types. This also demonstrates that training with only copy-move samples makes it easier for the model to learn generic features. Due to the small number of artificially faked samples, using some other real dataset to produce faked samples algorithmically is often used. If we want the model to focus more on generic features of different forgery types, we can use only copy-move type samples for training in the pre-training phase. We found that the original image's quality also affects the model's performance in forgery localization. As we analyzed in Section 4.4, the model's performance degrades when the quality of the original images we used to produce the copy-move dataset differs significantly from the original images of the realistic forged samples. We can perform some global operations on the original images to simulate images of different quality.These operations can produce both degraded and enhanced versions of the images.When we use higher quality original images for the faked dataset, it is more beneficial to the model's generalization ability.

# 6 Conclusions

In this paper, we demonstrate that under our experimental conditions, copy-move samples are more conducive to models learning generic features. Even when trained only on copy-move samples, our model demonstrates some effectiveness for splicing forged samples in our experiments. This is because both splicing and copy-move have a process of pasting the image content, and the model captures the different features at the edges due to the forgery process. This demonstrates another perspective, where simple straightforward hybrid training will often not achieve optimal results in terms of model generality due to the unique nature of the different forgery types. In addition, we found that images of

different quality underwent the same tampering process and that the tampering features used to detect localization were not identical, even though they had the same semantic content. We have produced three types of copy-move datasets for training and used a multiple prior fusion strategy to train the model, using a generic architecture of encoder-decoder for our model. Extensive experiments were conducted on multiple public datasets, showing that our approach performs better than other existing approaches.

# Funding

This research was funded by the National Key Research and Development Program of China (No. 2022YFF0712500), the Basic Research Business of Central Universities of North Minzu University (Grant No. 2023ZRLG02), the High School Scientific Research Project of Ningxia (Grant No. NYG2024066), the National Natural Science Foundation of China (Grant Nos. 62562002 and 62462001), and the Ningxia Natural Science Foundation (Grant No. 2025AAC020007).

# Author Contributions

Conceptualization, Jiaqi Zhang, Liqiong Jian and Jinlin Ma; methodology, Jiaqi Zhang and Liqiong Jian; software, Jiaqi Zhang; validation, Jiaqi Zhang, Liqiong Jian and Ziping Ma; formal analysis, Jiaqi Zhang and Liqiong Jian; investigation, Jiaqi Zhang, Liqiong Jian and Ziping Ma; resources, Liqiong Jian, Jinlin Ma and Xiaoshuai Huang; data curation, Jiaqi Zhang and Liqiong Jian; writing—original draft preparation, Jiaqi Zhang and Liqiong Jian; writing—review and editing, Jinlin Ma, Xiaoshuai Huang and Ziping Ma; visualization, Jiaqi Zhang and Liqiong Jian; supervision, Jinlin Ma and Xiaoshuai Huang; project administration, Jinlin Ma; funding acquisition, Ziping Ma, Jinlin Ma and Xiaoshuai Huang All authors have read and agreed to the published version of the manuscript.

# Conflict of Interest

All the authors declare that they have no conflict of interest.

# References

[1] Chierchia, G., Poggi, G., Sansone, C., Verdoliva, L.: A Bayesian-MRF approach for PRNU-based image forgery detection. IEEE Transactions on Information Forensics and Security **9**(4), 554–567 (2014). https://doi.org/10.1109/TIFS.2014.2302078

[2] Korus, P., Huang, J.: Multi-scale analysis strategies in PRNU-based tampering localization. IEEE Transactions on Information Forensics and Security **12**(4), 809–824 (2016). https://doi.org/10.1109/TIFS.2016.2636089

[3] Mahdian, B., Saic, S.: Using noise inconsistencies for blind image forensics. Image and Vision Computing **27**(10), 1497–1503 (2009). https://doi.org/10.1016/j.imavis.2009.02.001

[4] Qu, C., Zhong, Y., Liu, C., Xu, G., Peng, D., Guo, F., Jin, L.: Towards Modern Image Manipulation Localization: A Large-Scale Dataset and Novel Methods. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10781–10790 (2024). https://doi.org/10.1109/CVPR52733.2024.01025

[5] Johnson, M.K., Farid, H.: Exposing digital forgeries by detecting inconsistencies in lighting. In Proceedings of the 7th Workshop on Multimedia and Security, pp. 1–10 (2005). https://doi.org/10.1145/1073170.1073171

[6] Matern, F., Riess, C., Stamminger, M.: Gradient-based illumination description for image forgery detection. IEEE Transactions on Information Forensics and Security **15**, 1303–1317 (2019). https://doi.org/10.1109/TIFS.2019.2935913

[7] Li, S., Ma, W., Guo, J., Xu, S., Li, B., Zhang, X.: UnionFormer: Unified-Learning Transformer with Multi-View Representation for Image Manipulation Detection and Localization, 12523–12533 (2024). https://doi.org/10.1109/CVPR52733.2024.01190

[8] Bianchi, T., Piva, A.: Image forgery localization via block-grained analysis of JPEG artifacts. IEEE Transactions on Information Forensics and Security **7**(3), 1003–1017 (2012). https://doi.org/10.1109/TIFS.2012.2187516

[9] Iakovidou, C., Zampoglou, M., Papadopoulos, S., Kompatsiaris, Y.: Content-aware detection of JPEG grid inconsistencies for intuitive image forensics. Journal of Visual Communication and Image Representation **54**, 155–170 (2018). https://doi.org/10.1016/j.jvcir.2018.05.011

[10] Pasquini, C., Boato, G., Pérez-González, F.: Statistical detection of JPEG traces in digital images in uncompressed formats. IEEE Transactions on Information Forensics and Security **12**(12), 2890–2905 (2017). https://doi.org/10.1109/TIFS.2017.2725201

[11] Huh, M., Liu, A., Owens, A., Efros, A.A.: Fighting Fake News: Image Splice Detection via Learned Self-Consistency. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 106–124 (2018). https://doi.org/10.1007/978-3-030-01252-6_7

[12] Bi, X., Pun, C.-M.: Fast reflective offset-guided searching method for copy-move forgery detection. Information Sciences **418**, 531–545 (2017). https://doi.org/10.1016/j.ins.2017.08.044

[13] Zhong, J.-L., Pun, C.-M.: An End-to-End Dense-InceptionNet for Image Copy-Move Forgery Detection. IEEE Transactions on Information Forensics and Security **15**, 2134–2146 (2019). https://doi.org/10.1109/TIFS.2019.2957693

[14] Pan, X., Lyu, S.: Region duplication detection using

image feature matching. IEEE Transactions on Information Forensics and Security **5**(4), 857–867 (2010). https://doi.org/10.1109/TIFS.2010.2078506

[15] Shivakumar, B., Baboo, S.S.: Detection of Region Duplication Forgery in Digital Images Using SURF. International Journal of Computer Science Issues (IJCSI) **8**(4), 199–205 (2011)

[16] Yu, Z., Ni, J., Zhang, J., Deng, H., Lin, Y.: Reinforced Multi-teacher Knowledge Distillation for Efficient General Image Forgery Detection and Localization. Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence, 10085–10094 (2025). https://doi.org/10.1609/aaai.v39i1.32085

[17] Wu, Y., AbdAlmageed, W., Natarajan, P.: Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9535–9544 (2019). https://doi.org/10.1109/CVPR.2019.00977

[18] Hu, X., Zhang, Z., Jiang, Z., Chaudhuri, S., Yang, Z., Nevatia, R.: SPAN: spatial pyramid attention network for image manipulation localization. In European Conference on Computer Vision, pp. 312–328 (2020). https://doi.org/10.1007/978-3-030-58589-1_19

[19] Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Learning rich features for image manipulation detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1053–1061 (2018). https://doi.org/10.1109/CVPR.2018.00116

[20] Bi, X., Zhang, Z., Xiao, B.: Reality Transform Adversarial Generators for Image Splicing Forgery Detection and Localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 14294–14303 (2021). https://doi.org/10.1109/ICCV48922.2021.01403

[21] Hao, J., Zhang, Z., Yang, S., Xie, D., Pu, S.: TransForensics: Image Forgery Localization With Dense Self-Attention. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 15035–15044 (2021). https://doi.org/10.1109/ICCV48922.2021.01478

[22] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In European Conference on Computer Vision, pp. 740–755 (2014). https://doi.org/10.1007/978-3-319-10602-1_48

[23] Chen, X., Dong, C., Ji, J., Cao, J., Li, X.: Image Manipulation Detection by Multi-View Multi-Scale Supervision. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 14185–14193 (2021). https://doi.org/10.1109/ICCV48922.2021.01392

[24] Li, D., Zhu, J., Liu, Y., Lu, X., Fu, X., Liu, J., Liu, A., Zha, Z.-J.: Learnable Frequency Decomposition for Image Forgery Detection and Localization. Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), 1359–1367 (2025). https://doi.org/10.24963/ijcai.2025/152

[25] Fu, H., Cao, X.: Forgery authentication in extreme wide-angle lens using distortion cue and fake saliency map. IEEE Transactions on Information Forensics and Security **7**(4), 1301–1314 (2012). https://doi.org/10.1109/TIFS.2012.2195492

[26] Ferrara, P., Bianchi, T., De Rosa, A., Piva, A.: Image forgery localization via fine-grained analysis of CFA artifacts. IEEE Transactions on Information Forensics and Security **7**(5), 1566–1577 (2012). https://doi.org/10.1109/TIFS.2012.2202227

[27] Cozzolino, D., Verdoliva, L.: Noiseprint: a CNN-based camera model fingerprint. IEEE Transactions on Information Forensics and Security **15**, 144–159 (2019). https://doi.org/10.1109/TIFS.2019.2916364

[28] Cozzolino, D., Poggi, G., Verdoliva, L.: Efficient dense-field copy–move forgery detection. IEEE Transactions on Information Forensics and Security **10**(11), 2284–2297 (2015). https://doi.org/10.1109/TIFS.2015.2455334

[29] Emam, M., Han, Q., Niu, X.: PCET based copy-move forgery detection in images under geometric transforms. Multimedia Tools and Applications **75**(18), 11513–11527 (2016). https://doi.org/10.1007/s11042-015-2872-2

[30] Jiang, L., Lu, Z., Gao, Y., Wang, Y.: Image Copy-Move Forgery Detection and Localization Scheme: How to Avoid Missed Detection and False Alarm. arXiv preprint arXiv:2406.03271 (2024). https://doi.org/10.48550/arXiv.2406.03271

[31] Wang, J., Wu, Z., Chen, J., Han, X., Shrivastava, A., Lim, S.-N., Jiang, Y.-G.: ObjectFormer for Image Manipulation Detection and Localization. arXiv preprint arXiv:2203.14681 (2022). https://doi.org/10.48550/arXiv.2203.14681

[32] Mahfoudi, G., Tajini, B., Retraint, F., Morain-Nicolier, F., Dugelay, J.L., Marc, P.: DEFACTO: Image and face manipulation dataset. In 2019 27th European Signal Processing Conference (EUSIPCO), pp. 1–5 (2019). https://doi.org/10.23919/EUSIPCO.2019.8903181

[33] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014). https://doi.org/10.48550/arXiv.1409.1556

[34] Bappy, J.H., Roy-Chowdhury, A.K., Bunk, J., Nataraj,

L., Manjunath, B.: Exploiting spatial structure for localizing manipulated image regions. In Proceedings of the IEEE International Conference on Computer Vision, pp. 4980–4989. https://doi.org/10.1109/ICCV.2017.532

[35] Bappy, J.H., Simons, C., Nataraj, L., Manjunath, B., Roy-Chowdhury, A.K.: Hybrid lstm and encoder–decoder architecture for detection of image forgeries. IEEE Transactions on Image Processing **28**(7), 3286–3300 (2019). https://doi.org/10.1109/TIP.2019.2895466

[36] Wen, B., Zhu, Y., Subramanian, R., Ng, T.-T., Shen, X., Winkler, S.: COVERAGE—A novel database for copy-move forgery detection. In 2016 IEEE International Conference on Image Processing (ICIP), pp. 161–165 (2016). https://doi.org/10.1109/ICIP.2016.7532339

[37] Dong, J., Wang, W., Tan, T.: Casia image tampering detection evaluation database. In 2013 IEEE China Summit and International Conference on Signal and Information Processing, pp. 422–426 (2013). https://doi.org/10.1109/ChinaSIP.2013.6625374

[38] Guan, H., Kozak, M., Robertson, E., Lee, Y., Yates, A.N., Delgado, A., Zhou, D., Kheyrkhah, T., Smith, J., Fiscus, J.: MFC datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW), pp. 63–72 (2019). https://doi.org/10.1109/WACVW.2019.00018

[39] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243

[40] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90

[41] Cui, Y., Jiang, C., Wang, L., Wu, G.: MixFormer: End-to-End Tracking with Iterative Mixed Attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13598–13608 (2022). https://doi.org/10.1109/CVPR52688.2022.01324

[42] Krawetz, N.: A picture's worth. Hacker Factor Solutions **6**(2), 2 (2007)

[43] Salloum, R., Ren, Y., Kuo, C.-C.J.: Image splicing localization using a multi-task fully convolutional network (MFCN). Journal of Visual Communication and Image Representation **51**, 201–209 (2018). https://doi.org/10.1016/j.jvcir.2018.01.010

[44] Mayer, O., Stamm, M.C.: Forensic similarity for digital images. IEEE Transactions on Information Forensics and Security **15**, 1331–1346 (2019). https://doi.org/10.1109/TIFS.2019.2924552