mathematics

MDPI

Article

MMFD-Net: A Novel Network for Image Forgery Detection and Localization via Multi-Stream Edge Feature Learning and Multi-Dimensional Information Fusion

Haichang Yin [1], KinTak U [1,*], Jing Wang [2,*] and Zhuofan Gan [3]

[1] Faculty of Innovation Engineering, Macau University of Science and Technology, Macau, China; 2009853gia30003@student.must.edu.mo
[2] CEPREI Certification Body, Guangzhou CEPREI Certification Center Service Co., Ltd., Guangzhou 511370, China
[3] School of Computer Science, Guangdong University of Finance, Guangzhou 510520, China; 221526103@m.gduf.edu.cn
* Correspondence: ktu@must.edu.mo (K.U.); wangjing@ceprei.org (J.W.)

Abstract

With the rapid advancement of image processing techniques, digital image forgery detection has emerged as a critical research area in information forensics. This paper proposes a novel deep learning model based on Multi-view Multi-dimensional Forgery Detection Networks (MMFD-Net), designed to simultaneously determine whether an image has been tampered with and precisely localize the forged regions. By integrating a Multi-stream Edge Feature Learning module with a Multi-dimensional Information Fusion module, MMFD-Net employs joint supervised learning to extract semantics-agnostic forgery features, thereby enhancing both detection performance and model generalization. Extensive experiments demonstrate that MMFD-Net achieves state-of-the-art results on multiple public datasets, excelling in both pixel-level localization and image-level classification tasks, while maintaining robust performance in complex scenarios.

Keywords: digital image forgery detection; multi-branch multi-dimensional forgery detection; multi-stream edge feature learning module; multi-dimensional information fusion module

MSC: 68U10

check for updates

Academic Editor: Sheng-Wei Chi

Received: 15 July 2025
Revised: 4 September 2025
Accepted: 15 September 2025
Published: 1 October 2025

Citation: Yin, H.; U, K.; Wang, J.; Gan, Z. MMFD-Net: A Novel Network for Image Forgery Detection and Localization via Multi-Stream Edge Feature Learning and Multi-Dimensional Information Fusion. *Mathematics* **2025**, *13*, 3136. https://doi.org/10.3390/math13193136

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Image editing tools such as Photoshop, GIMP, and Photoscape have become highly accessible thanks to advancements in image processing and deep learning. However, this accessibility has also given rise to a concerning issue: the malicious exploitation of these tools to manipulate digital images and disseminate altered content across the Internet. Such actions have led to widespread public skepticism, misinformation, and outrage, triggering a cascade of public opinion crises and emerging as a significant threat to social stability.

Digital image forgery detection [1] is a critical field that employs advanced scientific methods and cutting-edge technical approaches to analyze digital images meticulously. Its primary objective is to determine whether an image has been subjected to manipulation and, if so, to precisely identify the regions that have been altered.

As a long-standing research focus within the domain of information forensics, digital image forgery detection plays a pivotal role in equipping the public and governmental

bodies with the tools to discern and counteract the dissemination of fraudulent digital images. It not only safeguards social stability but also protects the legitimate rights and interests of individuals and organizations. Moreover, it provides indispensable forensic evidence for judicial authorities, facilitating the accurate adjudication of cases involving digital image manipulation.

Currently, the most common image forgery methods are threefold [2]: inpainting (removing content in an image that one does not wish to display), splicing (copying and pasting a segment of one image into another image), and copy-move (copying and pasting a segment of an image to other regions within the same image). Forensic methods targeting these forgery techniques can generally be divided into two categories: traditional feature extraction-based forensic algorithms and deep learning-based forensic algorithms.

Traditional feature extraction-based forensic algorithms mainly focus on extracting distinctive features from forged images to differentiate between authentic and forged regions. These features can be extracted through various means, including compression artifacts in media formats [3,4], inconsistencies in lighting [5,6], statistical patterns [7], and local noise assessment [8]. However, the greatest challenge faced by traditional detection algorithms is their inability to address multiple forgery methods using a single feature.

To address the aforementioned challenges, the state-of-the-art algorithms are currently mostly based on deep neural network (DNN) models. Some of these models focus solely on authenticating the image at the image level [9], while others concentrate exclusively on locating forgery regions at the pixel level [10–13]. Some models perform authentication and forgery localization at both levels [2,14–18]. However, if the task of pixel-level localization is regarded as a simplified version of semantic segmentation, the design of the detection model may overly emphasize the extraction of semantic information from the image, thereby neglecting the differences between authentic and forged content. This can also lead to the model becoming overly reliant on the dataset, thereby giving rise to generalization issues [2,19]. Therefore, constructing and training a network model that can accurately identify forgery regions in complex scenarios (i.e., extracting semantically irrelevant features) has become a key issue.

To enable DNN models to accurately detect forgery traces, some studies have attempted to transform the RGB view into other views. Detection methods based on such transformations can be categorized into two types: noise perception-based algorithms [9,11,20–26] and edge detection-based algorithms [10,27–31]. Algorithms based on noise-aware exploit the characteristic that the noise distribution of spliced or removed forgery images differs from that of authentic images. This difference can be exposed through noise perception algorithms. However, for image copy-move forgery, the forgery regions originate from the original image itself, which invalidates the aforementioned assumption in this type of manipulation.

The other approach, edge detection-based algorithms, attempts to identify edge inconsistencies between forgery and authentic regions. Existing techniques unify the features from each layer of the backbone network by summation or concatenation before feeding them into auxiliary branches. However, these methods still treat the features from each layer as semantically aware, and thus the issue of model generalization remains.

Moreover, in evaluating the generalization ability of models, existing deep learning methods, after being trained on public datasets, only assess performance at the pixel level using forged images from other public datasets, while neglecting the evaluation of performance on authentic images at the image level. In real-world scenarios, the probability of encountering authentic images is the highest, which also leads to a persistently high false positive rate for authentic images. Therefore, reducing the false positive rate for authentic images is of paramount importance.

We introduce MMFD-Net (Figure 1) to jointly detect three manipulation types and boost generalization. The network comprises three branches:

Top: Image-Level Detection (ILD); Middle: Multi-Stream Edge Feature Learning (MSEFL, fuses side outputs from ILD and PLD); Bottom: Pixel-Level Detection (PLD).

Both the ILD and PLD branches employ ResNet-50 as their backbone network, while the MSEFL branch takes the side outputs from the backbone networks of the aforementioned two branches as its input.

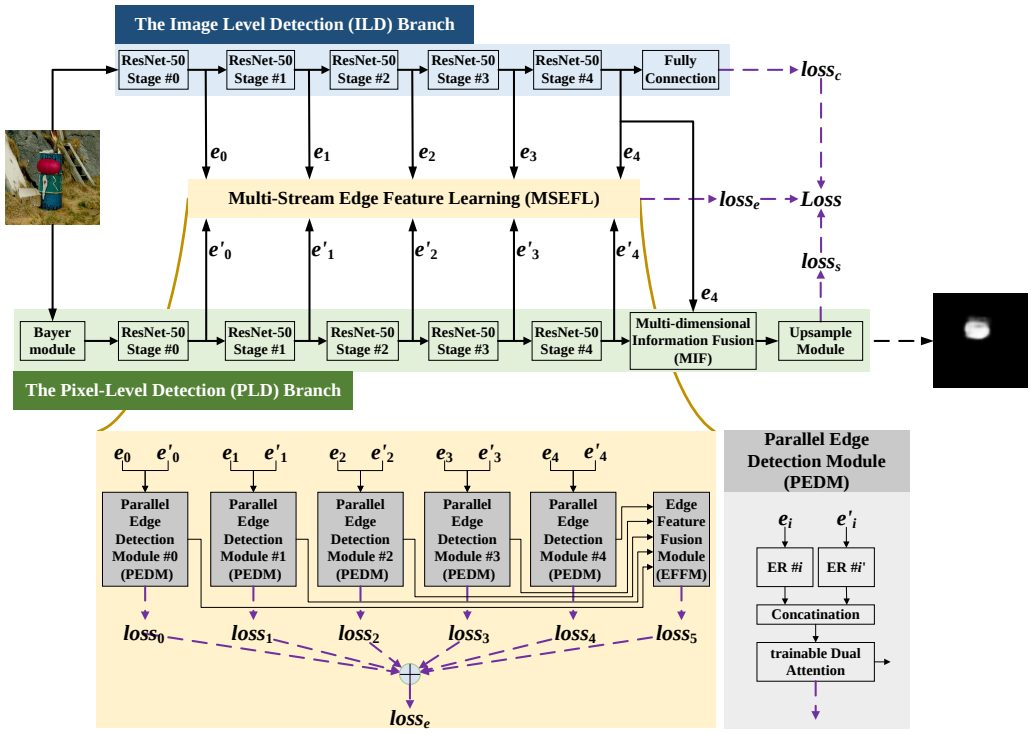


Figure 1. Concept diagram of the proposed Multi-branch Multi-dimensional Forgery Detection Networks.

In summary, the contributions of the method are as follows:

1. Multi-Stream Edge Feature Learning (MSEFL):
   A lightweight module that fuses low-level edges with high-level semantics to generate forgery-edge-sensitive representations. Edge-supervised training propagates these cues to the rest of the network, boosting manipulation-trace perception and domain transfer.
2. Multi-dimensional Information Fusion (MIF) in PLD:
   Dual self-attention blocks attend to noise-view and color-view cues independently, then fuse them to achieve sharper pixel-level localization of forged regions.
3. Unified MMFD-Net framework:
   The first architecture to jointly optimize image-level authenticity classification, pixel-level forgery segmentation, and edge-aware supervision. This end-to-end tri-task learning extracts semantic-agnostic forgery signatures, yielding superior detection accuracy and robust generalization.

2. Related Work

CNNs excel at modeling local correlations within images but struggle to capture global dependencies. Vision Transformers (ViTs) overcome this limitation via self-attention, establishing direct relationships among all image features and delivering stronger global perception. In image forgery forensics, each architecture offers distinct advantages; dual-

stream networks that fuse both modalities jointly capture local and global cues, enhancing detection accuracy and robustness.

Beyond CNNs and ViTs, techniques such as LSTMs and GANs have also been adopted, further enriching the methodological landscape. When classified by the type of forgery traces they target, existing approaches fall into three categories: noise inconsistency, edge-based detection, and multi-feature fusion. We briefly outline implementations within these categories and highlight our contributions.

2.1. Forgery Detection Methods Based on Image Content Noise Features

In recent years, forgery detection methods based on content noise features have garnered widespread attention due to their sensitivity to manipulation traces. These methods primarily exploit inconsistencies in noise within an image to identify forgery regions. The following is a summary of relevant research on noise feature-based forensic methods: Zhou et al. [11] combined RGB features with noise features to capture subtle differences between forgery and authentic regions. Huang et al. [20] proposed using noise maps generated by Steganalysis Rich Model (SRM) filters to enhance detection accuracy. Niu et al. [21] introduced a guided and multi-scale feature aggregation network for image forgery localization. Zhu et al. [22] developed a two-step discriminative noise-guided approach that explicitly enhances the representation and utilization of noise inconsistencies. This method significantly improves detection accuracy and robustness. Kwon et al. [24] proposed detecting and localizing image forgery by learning JPEG compression artifacts. This method leverages artifacts generated during the JPEG compression process as features of forgery traces. Wang et al. [23] proposed the ObjectFormer method for capturing subtle forgery traces that are no longer visible in the RGB domain. Hu et al. [9] introduced a Spatial Pyramid Attention Network (SPAN) architecture that effectively models the relationships between multi-scale image patches by constructing a pyramid of local self-attention blocks for detecting and localizing various types of image forgery. Guillermo et al. [25] proposed the TruFor framework, which combines RGB images with a transformer that learns noise-sensitive fingerprints in a fusion architecture to extract both high-level and low-level traces, enabling robust detection of various image forgery methods.

2.2. Forgery Detection Methods Based on Edge-Based Detection

Forgery operations often leave edge traces around the manipulated regions, making edge feature-based detection an effective approach for image forgery detection. For example, Salloum et al. [10] proposed a multi-task fully convolutional network (MFCN) to improve the localization accuracy of image splicing forgery regions. UGEE-Net [27] focuses on the fusion and interaction of high-level features in the spatial domain, balancing global semantics and local details. Additionally, it incorporates frequency-domain features to extract edge information, further improving localization accuracy.

Sun et al. [29] proposed SAFL-Net, which enhances the model's generalization ability by constraining the feature extractor to learn semantic-agnostic features through the design of specific modules and auxiliary tasks. Lin et al. [28] improved detection accuracy by combining multiple forgery traces and enhancing edge artifacts. This method leverages forgery traces along image edges, which are crucial for detecting manipulated regions. By enhancing edge artifacts, the method can more effectively identify and localize forgery regions in images. Ma et al. [30] proposed the IML-ViT method, which is based on the Vision Transformer (ViT) and aims to improve image forgery detection performance by capturing forgery traces.

2.3. Forgery Detection Methods Based on Multi-Feature Fusion Methods

To enhance the robustness and accuracy of forgery detection, many researchers have begun to explore multi-feature fusion methods. These methods typically combine RGB features, noise features, and edge features to comprehensively capture forgery traces. For example, Hu et al. [9] proposed a Spatial Pyramid Attention Network (SPAN), which integrates RGB features and noise features to improve the performance of forgery detection. Additionally, Chen et al. [2] introduced a Multi-View Salient Structure Network (MVSS-Net) that enhances edge feature extraction by incorporating an edge-supervised branch. Han et al. [19] designed a novel end-to-end network called HDF-Net to extract homogeneity difference features for precise localization of manipulation artifacts, significantly improving localization accuracy and edge refinement.

Despite the significant progress made in existing research, current methods still face challenges when dealing with complex forgery operations such as copy-move and splicing. In this paper, we propose the Multi-branch Multi-dimensional Forgery Detection Networks (MMFD-Net), which capture homogeneity difference features between forgery and authentic regions to achieve precise localization of forgery areas. MMFD-Net fuses color and noise features to detect forged images at both the image level and pixel level, while the Multi-Stream Edge Feature Learning (MSEFL) module learns low-level edge features and high-level abstract features between forgery and authentic regions. This enhances the model's perception of forgery edges during feature extraction, thereby improving the accuracy and robustness of tamper detection.

3. The Proposed Method

Our goal is to build a multi-branch deep network, M. The network uses three different annotations for supervised learning of forgery features, which can not only determine whether an image has been manipulated but also indicate the forgery regions in the image. Moreover, the forgery edge annotation plays a key role in the supervised training. It can make the network pay attention to the artificial traces between the real and forgery regions, and improve the generalization ability of the network.

Given an input image $x \in \mathbb{R}^{H\times W\times 3}$, for our proposed model M, the image-level detection (ILD) module is denoted as $M_{ILD}(x)$, the pixel-level detection (PLD) module is denoted as $M_{PLD}(x, M_{ILD}(x))$, and the Multi-Stream Edge Feature Learning (MSEFL) module is denoted as $M_{MSEL}(M_{ILD}(x), M_{PLD}(x, M_{ILD}(x)))$. At the image level, M_{ILD} outputs the probability P_{ILD} that the image has been forged. At the pixel level, M_{PLD} and M_{MSEL} output a forgery region probability map P_{PLD} and a forgery edge probability map P_{MSEFL}, respectively, with the same dimensions as the detection image. The overall description of the proposed model is shown in Equation (1).

$$M = \begin{cases} M_{ILD}(x) \rightarrow P_{ILD} \\ M_{PLD}(x, M_{ILD}(x)) \rightarrow P_{PLD} \\ M_{MSEL}(M_{ILD}(x), M_{PLD}(x)) \rightarrow P_{MSEFL} \end{cases} \tag{1}$$

In Equation (1), the side output of M_{ILD} and detection image x serve as inputs for M_{PLD}, while the side outputs of M_{ILD} and M_{PLD} serve as inputs for M_{MSEFL}.

In the inference phase, P_{ILD} is used to predict whether the image has been forged at the image level, while P_{PLD} predicts whether image pixels have been forged at the pixel level.

3.1. The Image-Level Detection Branch

In this branch, the main task is to perform binary classification of images based on whether there are traces of forgery within the image. Currently, in common image

classification tasks, many network models can excellently complete tasks based on the semantic structure within images. However, unlike these classification tasks, the forgery regions in manipulated images are semantic-agnostic. Therefore, the model should classify images based on forgery traces. Additionally, because the pixels in forgery regions occupy a low proportion of the entire image, the model is required to retain forgery features even as the number of network layers increases to prevent network performance degradation due to gradient vanishing.

The core of ResNet-50 is Residual Learning, which enables the network to learn residual mappings rather than directly learning the target function by introducing Skip Connections. Additionally, skip connections allow gradient information to be directly transmitted from later layers to earlier layers, alleviating the vanishing gradient problem. Meanwhile, through layer-by-layer stacking of convolutional layers, the receptive field gradually expands, enabling it to capture both local features and model global patterns to adapt to different types of forgery methods. Moreover, forgery operations introduce subtle local artifacts (such as edge discontinuities and texture anomalies); thus, ResNet-50's stacking approach can extract multi-level features from the input image. This is beneficial for detecting the edges of forgery regions. Here, $P_{ILD} \in \mathbb{R}^{B\times 2\times 1}$ denotes the output of the fully connected layer; $e_i \in \mathbb{R}^{B\times C_i\times W_i\times H_i}$ denotes the side output of ResNet-50 stage $i \in \{0,\ldots,4\}$ in ILD. As shown in Figure 1, given the input image x, ILD is defined as Equation (2).

$$ResNet(x) \rightarrow \{e_i, P_{ILD} \mid 0 \leq i \leq 4\} \tag{2}$$

The side outputs (e_i) are used for subsequent edge detection supervised learning, while P_{ILD} is used for binary classification supervised learning.

3.2. The Pixel-Level Detection Branch

In this branch, ResNet-50 is used as the backbone network, and a noise-aware convolutional neural network is introduced as the preprocessing module of PLD, replacing the fully connected layer with an upsampling layer. Meanwhile, a multi-dimensional information fusion (MIF) module based on self-attention is designed to fuse the forgery feature information of ILD and PLD. Here, $BayarConv(\cdot)$ denotes the noise-aware convolutional neural network; $P_{PLD} \in \mathbb{R}^{B\times C_4\times W\times H}$ denotes the output of PLD, used for supervised learning of forgery region segmentation; $e'_i \in \mathbb{R}^{B\times C_i\times W_i\times H_i}$ denotes the side output of ResNet-50 stage $i \in \{0,\ldots,4\}$ in PLD. The PLD, as shown in Figure 1, is defined as Equation (3).

$$ResNet(BayarConv(x), e_4) \rightarrow \left\{e'_i, P_{PLD} \mid 0 \leq i \leq 4\right\} \tag{3}$$

The side outputs ($\{e'_i \mid 0 \leq i \leq 4\}$) of PLD and the side outputs ($\{e_i \mid 0 \leq i \leq 4\}$) of ILD are jointly used for subsequent edge detection supervised learning, while e'_4 is combined with e_4 for feature enhancement of the forgery region, thereby conducting subsequent image segmentation supervised learning.

3.2.1. The Noise-Aware Module

According to Dong et al. [2] and Bayar [32], BayarConv has excellent noise perception capabilities. It can distinguish the differences in noise characteristics between the pasted regions and the authentic regions. BayarConv is a set of convolutional kernels for supervised training to detect noise characteristic differences. Utilizing this characteristic, implementing it as a preprocessing step for ResNet-50 is beneficial for subsequent modules to segment the forgery regions of the image based on noise differences.

In *BayarConv*, each convolutional kernel $\omega \in \mathbb{R}^{h \times w}$ is subject to two constraints, as shown in Equation (4).

$$\begin{cases} \omega(i,j) = -1, & \text{if } i = \lfloor h/2 \rfloor + 1 \text{ and } j = \lfloor w/2 \rfloor + 1 \\ \sum \omega(i,j) = 1, & \text{otherwise} \end{cases}, \quad \text{where} \begin{cases} i \in [0,h) \\ j \in [0,w) \end{cases} \tag{4}$$

In short, for the parameters of kernel ω, its center weight is set to -1, and the sum of the other weights equals 1. This constraint is applied throughout the entire supervised training process of the BayarConv module.

3.2.2. The Multi-Dimensional Information Fusion (MIF) Module

The ILD extracts image features from the color view, while the PLD extracts image features from the noise view. The forgery information extracted from a single branch is limited and insufficient to accurately localize the forgery region. The work of Dong et al. [2] has demonstrated that fusing features from both branches can effectively localize the forgery region. Meanwhile, the self-attention mechanism has achieved remarkable performance in many tasks and can also be applied to fuse image features to obtain more prominent forgery region information. Inspired by this, we design a Multi-dimensional Information Fusion (MIF) module based on self-attention for locating the forgery region, as shown in Figure 2. Here, K_c and K_n denote convolution kernels with 1×1 size, used to independently process the forgery region information of the two branches; $CA(\cdot)$ denotes a color attention module; $NA(\cdot)$ denotes a noise attention module; $*$ is the convolution operation. The process of MIF is defined in Equation (5):

$$M = K_c * CA(e_4) + K_n * NA(e_4') \tag{5}$$

The features from the color view branch (e_4) and noise view branch (e_4') are separately enhanced by their corresponding attention modules ($CA(\cdot)$ and $NA(\cdot)$). They are then weighted by convolution kernels (K_c, K_n) and linearly combined, producing a more discriminative representation of forgery regions. Here, $SA(\cdot)$ denotes the self-attention module, so $CA(\cdot)$ and $NA(\cdot)$ are defined in Equation (6):

$$\begin{cases} CA(e_4) = SA(e_4) \times e_4 + e_4 \\ NA(e_4') = SA(e_4') \times e_4' + e_4' \end{cases} \tag{6}$$

where $\times$ represents element-wise multiplication. This equation describes how self-attention (SA) is applied to both color and noise feature maps. By multiplying the input with its attention weights and adding it back to the original map, the model highlights the most relevant forgery-related patterns while preserving the original feature context. The specific implementation of self-attention is as follows:

(a) For an input map $I \in \mathbb{R}^{c \times w \times h}$, three kernels ($K_q$, K_v, K_k) with 1×1 size are used to implement convolution operations with I, respectively, to obtain three feature maps (queries q, values v, keys k);
(b) The three feature maps are flattened, and then q is transposed;
(c) q and k undergo matrix multiplication. The results are processed by the Softmax function to obtain attention weights w;
(d) The element-wise multiplication of v and w yields the final attention maps.

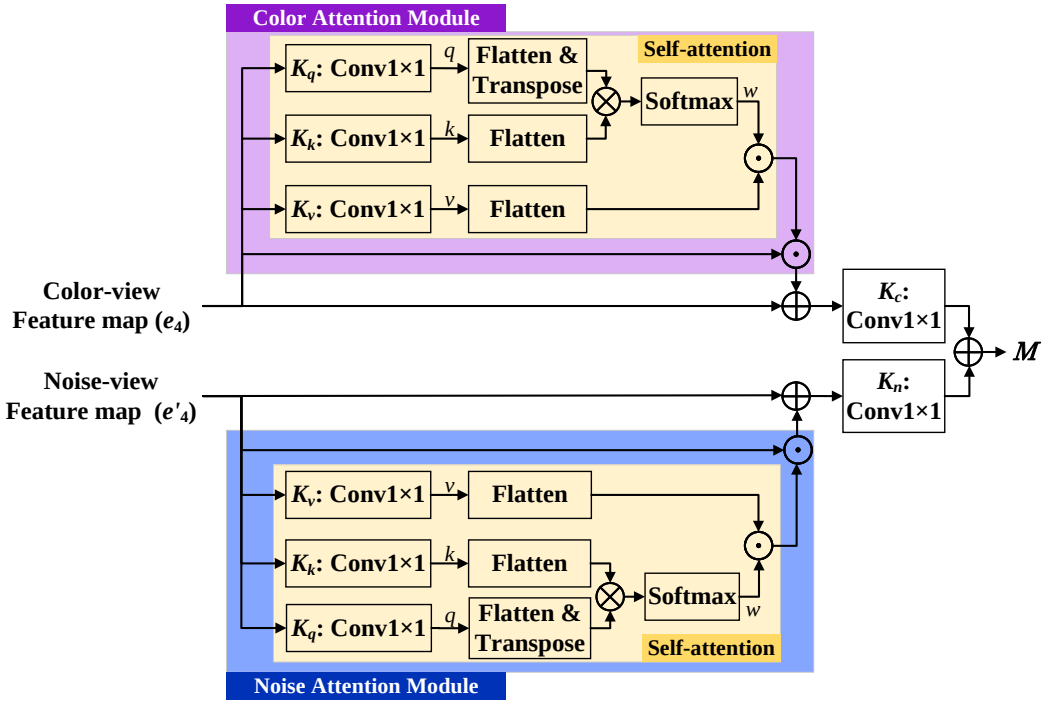


Figure 2. The structure of the Multi-dimensional Information Fusion module.

3.3. The Multi-Stream Edge Feature Learning Branch

In ResNet-50, the side outputs from the lower stages contain rich low-level edge features, while the side outputs from the higher stages possess high-level abstract features. Meanwhile, in the two branches, the backbone network extracts features from both the color view and noise view. As a result, the side outputs from the two branches complement each other in terms of tamper region edge information. Additionally, performing edge-supervised learning on the side outputs of the backbone has been shown to improve the model's generalization ability and overall performance in image classification and segmentation tasks [14].

Therefore, we propose a Multi-Stream Edge Feature Learning (MSEFL) Module for edge-supervised learning. By combining the side outputs from different stages, the MSEFL module can simultaneously leverage low-level edge features and high-level abstract features to form multi-level features. These multi-level features are then utilized for tamper edge-supervised learning, enhancing the model's perception of boundary changes between forgery regions and authentic regions (and vice versa).

In the MSEFL module, defined in Equation (7), there are five structurally identical Parallel Edge Detection Modules (PEDMs) and one Edge Feature Fusion Module (EFFM). Given two arrays of side outputs $\{e_i \mid 0 \le i \le 4\}$ and $\{e'_i \mid 0 \le i \le 4\}$ from ILD and PLD, two side output elements of the two arrays are input into different PEDMs, respectively. The five output maps $\left\{E^i_{\text{edge}} \mid 0 \le i \le 4\right\}$ from the PEDMs are not only used for subsequent supervised learning but also serve as inputs to the EFFM. Finally, the MSEFL module outputs six maps $\{E^i_{\text{edge}} \mid 0 \le i \le 5\}$used for supervised learning.

$$MSED(x) \rightarrow P_{\text{edge}} = \left\{E^i_{\text{edge}} \mid 0 \le i \le 5\right\} \tag{7}$$

This equation defines the output of MSEFL. It produces six edge prediction maps (from different backbone stages and fusion layers), which are used to capture tampering boundaries at multiple levels of abstraction.

3.3.1. Parallel Edge Detection Module

The side outputs ($e_i \in \mathbb{R}^{B\times C_i\times H_i\times W_i}$ and $e_i' \in \mathbb{R}^{B\times C_i\times H_i\times W_i}$) of stage i from ILD and PLD contain edge information corresponding to the respective stages of the two backbones. Therefore, in a PEDM, as shown in Figure 1, Edge Refining (ER) modules are applied separately to e_i and e_i', followed by trainable Dual Attention (tDA) [33] to fuse the two refined features.

(1) ER module: To refine the forgery region edge, we designed an ER module, as illustrated in Figure 3, to remove irrelevant edge information and preserve the forgery region edge. The specific implementation is as follows:

Given e_i or e_i', ER performs:

(a) Conv3 × 3 (stride 1, padding 1) with $C_i \rightarrow \lfloor C_i/16 \rfloor$, followed by BatchNorm and ReLU;
(b) Conv3 × 3 (stride 1, padding 1) with $\lfloor C_i/16 \rfloor \rightarrow 256$, followed by BatchNorm.

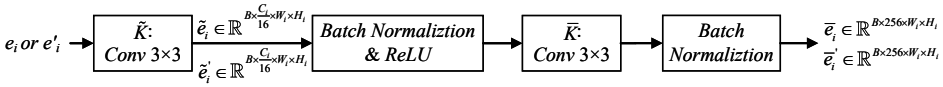


Figure 3. The structure of the edge refining module.

The ER output is thus $\bar{e}_i$ or $\bar{e}_i' \in \mathbb{R}^{B\times 256\times H_i\times W_i}$. We apply ER to both e_i and e_i' to obtain $\bar{e}_i$ and $\bar{e}_i'$. This concrete design stacks 3 × 3 convolutions with normalization and non-linearity while fixing channel widths for stable optimization.

(2) tDA module: The output maps $\bar{e}_i$ and $\bar{e}_i'$ from the ER module are concatenated to $E_a = Concat(\bar{e}_i, \bar{e}_i') \in \mathbb{R}^{B\times 512\times H_i\times W_i}$, which contains forged region edge information from two branches. It is noteworthy that the information carried by tampered edges constitutes only a minor fraction of the overall image content. Therefore, we need to further integrate this information to enhance the model's perception of forged region edges. To achieve this, the integration should be guided by an attention-driven mechanism to strengthen the features of tamper region edges while weakening the features of non-tamper region edges. Furthermore, these artifacts can be reliably captured if the model considers both where (spatial continuity) and what (channel semantics) aspects of the features. To this end, a trainable Dual Attention (tDA) module [33], as shown in Figure 4, combines two complementary attentions consisting of two attention modules: Position Attention (PA) module and Channel Attention (CA) module.

PA models long-range spatial dependencies by computing pairwise correlations across all pixel positions. This enables the network to highlight coherent edge regions and suppress isolated noisy responses. The specific implementation of PA is as follows:

(a) Three convolution kernels (K_b, K_c and $K_d \in \mathbb{R}^{1\times 1}$) are used to compute three projections of E_a, i.e., $E_b \in \mathbb{R}^{B\times 32\times W_i\times H_i}$, $E_c \in \mathbb{R}^{B\times 32\times W_i\times H_i}$ and $E_d \in \mathbb{R}^{B\times 512\times W_i\times H_i}$.
(b) Here, E_b and E_c are flattened to $\mathbb{R}^{B\times 32\times N}$, and E_d is flattened to $\mathbb{R}^{B\times 512\times N}$, where $N = W_i \times H_i$; E_b^T is the transposed matrix of E_b; S denotes the spatial attention map. According to the softmax function, as shown in Equation (8), obtained by calculation of spatial attention map $S \in \mathbb{R}^{B\times N\times N}$.

$$S = softmax(E_b^T \times E_c) \tag{8}$$

where × represents the matrix multiplication. Here, S is computed by measuring the similarity between feature vectors at different pixel locations. This operation helps the model identify which spatial regions are strongly correlated, a key step for detecting boundary inconsistencies introduced by image manipulation.

(c) Here, S^T denotes the transposed matrix of S; α is a trainable weight. E_a, E_d, and S^T calculated the feature map $E_{SA} \in \mathbb{R}^{B\times 512\times W_i\times H_i}$ according to Equation (9).

$$E_{SA} = \alpha \cdot reshape(E_d \times S^T) + E_a \tag{9}$$

where $\times$ represents the matrix multiplication, α is initialized as 0.

This equation refines the spatially attended features. The attention weights S are applied to the feature map E_d, reshaped back to the original size, and combined with the input feature E_a. The trainable parameter α controls the balance between the original and attention-enhanced features, thereby emphasizing manipulation boundaries.

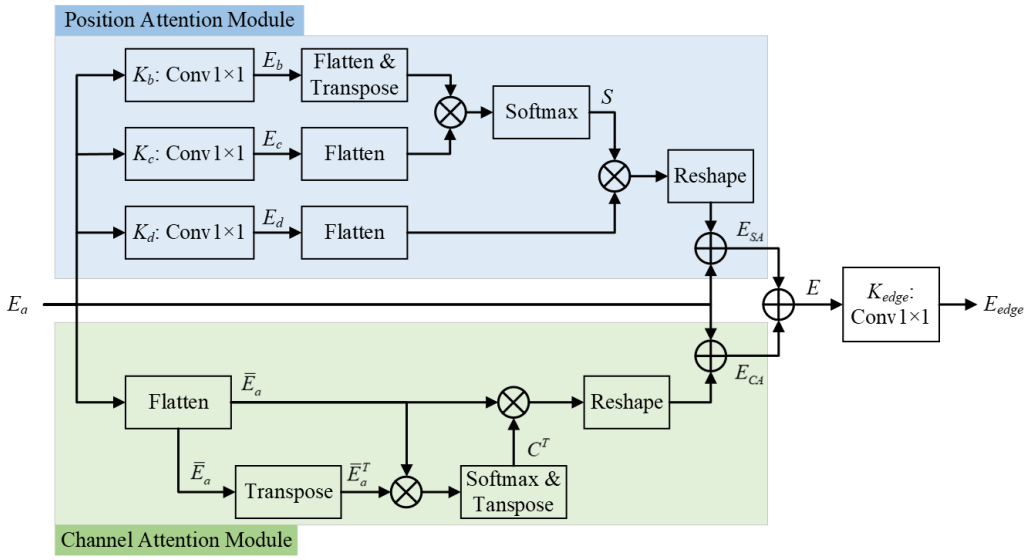


Figure 4. The structure of the trainable Dual Attention module.

The CA module models the inter-dependencies between feature channels, thereby enhancing feature dimensions that are highly relevant to manipulation traces (e.g., abnormal textures or resampling patterns). The specific implementation of CA is as follows:

(a) Here, E_a is flattened to $\mathbb{R}^{B\times 512\times N}$, resulting in $\overline{E}_a$; $\overline{E}_a^T$ is the transpose matrix of $\overline{E}_a$. The channel attention map $C \in \mathbb{R}^{B\times 512\times 512}$ is obtained through a calculation based on Equation (10).

$$C = softmax\left(\overline{E}_a \times \overline{E}_a^T\right) \tag{10}$$

where $\times$ represents the matrix multiplication.

This equation defines the channel attention map C. It measures the correlations between feature channels, indicating how much each channel contributes to detecting forgery cues. Highly correlated channels are assigned stronger weights.

(b) Here, C^T denotes the transpose matrix of C; β is a trainable scale parameter. The feature map $E_{CA} \in \mathbb{R}^{B\times 512\times W_i\times H_i}$ is calculated by E_a, $\overline{E}_a$ and C^T according to Equation (11).

$$E_{CA} = \beta \cdot reshape\left(C^T \times \overline{E}_a\right) + E_a \tag{11}$$

where β is initialized as 0.

This equation applies channel attention to enhance forgery-related channels. The attention-weighted channel features are reshaped and added back to the original input, with β serving as a trainable scaling factor. This mechanism amplifies channels sensitive to tampered edges while suppressing irrelevant ones.

After completing the two attention modules, the feature maps E_{SA} and E_{CA} are added together to obtain the final map $E \in \mathbb{R}^{B\times 512\times W_i\times H_i}$. And then, the edge prediction result $E_{edge} \in \mathbb{R}^{B\times 1\times W_i\times H_i}$ is obtained using a convolution kernel $K_{edge} \in \mathbb{R}^{1\times 1}$ for feature graph E.

3.3.2. Edge Feature Fusion Module

In the PEDMs, there are five single-channel edge predictions $\left\{E^i_{edge} \mid 0 \le i \le 4\right\}$ with edge information of different levels. The edge information should be fused to generate the final forgery region edge prediction. To this end, we design an Edge Feature Fusion Module (EFFM), whose detailed workflow is presented in Algorithm 1.

The five-edge prediction maps have different sizes, so we upsample them to the highest resolution $H_i \times W_i$ by a parameter-free bilinear up-sampling operation. After that, they are concatenated together.

Subsequently, we expand channels with a Conv1 × 1, then apply Conv3 × 3–BN–ReLU twice to integrate spatial and channel cues to produce the final edge probability map $E^5_{edge} \in \mathbb{R}^{B\times 1\times W_1\times H_1}$.

Algorithm 1: Edge Feature Fusion Module (EFFM)

Input: $\{E^i_{edge}\}^4_{i=0}$, where $E^i_{edge} \in \mathbb{R}^{B\times 1\times H_i\times W_i}$
Output: E^5_{edge}
$E_0, E_1 \leftarrow E^0_{edge}, E^1_{edge}$;
$E_2 \leftarrow \text{UpBilinear}(E^2_{edge}, \text{scale} = 2, \text{align_corners} = False)$;
$E_3 \leftarrow \text{UpBilinear}(E^3_{edge}, \text{scale} = 4, \text{align_corners} = False)$;
$E_4 \leftarrow \text{UpBilinear}(E^4_{edge}, \text{scale} = 4, \text{align_corners} = False)$;
$E_{\text{cat}} \leftarrow \text{Concat}(E_0, E_1, E_2, E_3, E_4)$; // $\in \mathbb{R}^{B\times 5\times H_1\times W_1}$
$F \leftarrow \text{Conv}_{1\times 1}(E_{\text{cat}}, \text{out_ch} = 64)$;
$F \leftarrow \text{Conv}_{3\times 3}(F) \rightarrow \text{BN} \rightarrow \text{ReLU}$;
$F \leftarrow \text{Conv}_{3\times 3}(F) \rightarrow \text{BN} \rightarrow \text{ReLU}$;
$E^{edge}_5 \leftarrow \sigma(\text{Conv}_{1\times 1}(F, \text{out_ch} = 1))$;
return E^{edge}_5;

3.4. Multi-Dimensional Supervision

In the training phase, the input training set is denoted by $T = \{(X, Y)\}$, where $X = \{x_i \mid 1 \le i \le |X|\}$ denotes the detected image set, $Y = \left\{\{y^{ILD}_i, y^{PLD}_i, y^{MSEFL}_i\} \mid 1 \le i \le |Y|\right\}$ denotes the corresponding ground truth for X. y^{ILD}_i is the binary label, and y^{PLD}_i and y^{MSEFL}_i are the binary maps with forgery region and forgery edge, respectively. In the training set, the size of x_i is resized to $W \times H \times 3$, and those of y^{PLD}_i and y^{MSEFL}_i are $W \times H \times 1$.

In the proposed model, the three branches have distinct task objectives. Therefore, the losses of all three branches need to be taken into account to enhance the performance of the network. The loss of ILD (image-scale loss) is used to boost the model's specificity at the image level. The loss of PLD (pixel-scale loss) is used to boost the model's sensitivity for the forgery region at the pixel level. The loss of MSEFL (edge loss) is used to learn semantic-agnostic information and boost the model's generalization at both the image and pixel levels.

Image-scale Loss: In ILD, this is a binary classification task. Therefore, we use the famous binary cross-entropy (BCE) to calculate the image-scale loss, defined as Equation (12).

$$loss_c = -\left(y^{ILD}\log(P_{ILD}) + (1 - y^{ILD})\log(1 - P_{ILD})\right) \tag{12}$$

This loss focuses on classifying each image as authentic or manipulated, ensuring that the model can make a reliable high-level decision about the presence of tampering. It is critical for improving image-level specificity and reducing false positives, preventing genuine images from being misclassified as forged. The image-level loss is most important when the primary objective is global classification rather than fine-grained localization.

Pixel-scale Loss: For a forgery image, the proportion of manipulated pixels is typically low. The Dice loss, defined in Equation (13), has proved effective for learning from imbalanced data [34], so we use it to compute the pixel-scale loss.

$$loss_s = 1 - \frac{2\sum_{i,j}\left(P_{PLD}(i,j)\cdot y^{PLD}(i,j)\right)}{\sum_{i,j}(P_{PLD}(i,j))^2 + \sum_{i,j}(yPLD(i,j))^2} \tag{13}$$

This loss targets the precise localization of manipulated regions within an image, enabling accurate pixel-wise identification and segmentation of tampered areas. It is key to improving sensitivity (recall) and reducing false negatives, ensuring that manipulated pixels are not missed. The pixel-level loss is most important when accurate localization is the goal. It is particularly effective under class-imbalance conditions where forged pixels are vastly outnumbered by authentic pixels.

Edge Loss: In computing the edge loss, the facing problem is the same as that of the pixel-scale loss, so the Dice loss is also used. Here, a collection $P_{MSEL} = \left\{E^i_{MSEL} \mid 0 \le i \le 5\right\}$ is outputted from MSEFL; $\delta_i \in (0,1)$ is the positive weight of the i^{th} stream, which should follow $\sum_{i=0}^{m}\delta_i = 1$; $\ell^{(i)}_{MSEL}$ denotes the Dice loss of the i^{th} stream at the pixel level, defined in Equation (14):

$$\ell_{MSEL} = 1 - \frac{2\sum_{i,j}\left(E_{MSEL}(i,j)\cdot y^{MSEL}_{i,j}\right)}{\sum_{i,j}(E_{MSEL}(i,j))^2 + \sum_{i,j}\left(y^{MSEL}_{i,j}\right)^2} \tag{14}$$

Therefore, the edge loss is defined as Equation (15):

$$loss_e = \sum_{i=0}^{m}\delta_i\,\ell^{(i)}_{MSEL}\left(E^i_{MSEL}, y^{MSEL}\right) \tag{15}$$

This loss encourages learning semantic-agnostic forgery cues by leveraging edge information. By emphasizing boundary discontinuities and inconsistencies, it supports robust detection and localization of manipulated regions and improves generalization. The edge loss is most important when detection relies on boundary cues rather than semantic content, and it performs especially well for complex manipulations where semantics are not a reliable indicator of forgery.

Combined Loss: After computing the three losses, a combined loss is obtained by a linear combination, defined as Equation (16):

$$Loss = \lambda \cdot loss_c + \mu \cdot loss_s + (1 - \lambda - \mu)\cdot loss_e \tag{16}$$

where $\lambda, \mu \in (0,1)$ are positive weights.

4. Experiments

4.1. Implementation Details

We implemented our model using PyTorch and trained it on an NVIDIA RTX 3090 GPU. In both training and testing, the size of all images was resized to 512×512. ILD and PLD, which use ResNet-50 as the backbone, were initialized with the corresponding models pre-trained on ImageNet.

Hyperparameter Setting. In the training phase, the Adam [35] optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with a weight decay of 1×10^{-4} was used to adjust the model parameters, with the learning rate adjusted cyclically from 10^{-4} to 10^{-7} using CosineAnnealingWarmRestarts with $T_0 = 10$ and $T_{mult} = 2$. The batch size was set as four, and early stopping with a patience of 10 was adopted based on the validation F_1 score. For the two hyperparameters (λ and μ) in the combined loss, we empirically set them as 0.1 and 0.2, respectively.

Data Augmentation. During the training process, we applied a fixed, seed-synchronized augmentation pipeline implemented with Albumentations, such that geometric transforms were applied identically to the image and mask, while photometric transforms affected the image only. The exact transformations and parameters of the operations are listed in detail in Table 1. Beyond generic augmentations, we synthesized manipulation priors with two custom dual-target transforms: Copy–Move, which copies a random rectangular patch ($h, w \in [50, 256]$) to a different location ($p = 0.1$), and Inpainting, which replaces a random window ($h, w \in [50, 256]$) using OpenCV inpainting (TELEA or Navier–Stokes; $p = 0.1$).

Table 1. The details of data augmentation operations.

Operation	Key Parameter	Act on	Applied Prob. (p)
Brightness	`brightness_limit` $= (-0.1, 0.1)$	Image	1.0
Contrast	`contrast_limit` $= 0.1$	Image	1.0
Rotation	90°	Image, Mask	0.5
Horizontal Flip	–	Image, Mask	0.5
Vertical Flip	–	Image, Mask	0.5
Blur	`blur_limit` $= (3, 7)$	Image	0.3
JPEG Compression	`quality`$\in [70, 100]$	Image	0.2
Copy-Move Manip.	Window size: $h, w \in [50, 256]$	Image, Mask	0.1
Inpainting Manip.	Window size: $h, w \in [50, 256]$; Algorithm: `cv2.INPAINT_TELEA` or `cv2.INPAINT_NS`	Image, Mask	0.1

4.2. Experiment Setting

Dataset. To ensure the scientific rigor and comparability of our research results, we carefully designed the selection and construction of our datasets.

For training and validation, we chose the well-established CASIAv2 dataset [36], which provides a solid foundation for model training with its rich samples. For testing, we selected multiple widely recognized datasets, including COVER [37], NIST16 [38], CASIAv1 [36], and IMD [39], to comprehensively evaluate the model's generalization ability and performance.

Following the dataset construction method of Dong et al. [2], we combined DEFACTO [40] and MS-COCO [41] to create a training set named DEF-84k and a testing set named DEF-12k. There is no data overlap between these two datasets to prevent data leakage. For DEF-84k, there are 64,000 forged images from DEFACTO and 20,000 authentic images from MS-COCO. For DEF-12k, there are 6000 forged images from DEFACTO and 6000 authentic images from MS-COCO.

In DEF-84k and CASIA v2, we held out 10% of the data as a validation set using a fixed random seed (set as 2147483647) and stratified by manipulation type (copy–move/splicing/inpainting) to maintain class balance across the train/validation splits.

In summary, our experiments involve two training sets and six test sets, with specific details shown in Table 2.

Table 2. The specific details of two training sets and six test sets.

Application	Dataset	Authentic Images	Forgery Images		
			cmpv.	spli.	inpa.
Training (Validation)	DEF-84k	18,000 (2000)	11,500 (1277)	30,720 (3413)	15,757 (1750)
	CASIA v2	6742(749)	2912 (323)	1646 (182)	0
Testing	COVER	100	100	0	0
	NIST16	0	68	288	208
	CASIA v1	800	459	461	0
	IMD	414	-	2010	-
	DEF-12k	6000	2000	2000	2000

Note: The IMD dataset does not classify the forged images by type. The total number of forged images is 2010.

Evaluation Criteria. A comprehensive set of evaluation criteria is essential to assess detection models' performance accurately. These criteria cover various aspects, including pixel-level and image-level detection. We calculate the F_1 score, which provides a balanced measure of these two metrics. Furthermore, we report the AUC (Area Under the ROC Curve) to evaluate the model's ability to distinguish between forged and authentic images. An AUC value closer to 1 indicates better performance in distinguishing forged images from authentic ones.

- **Pixel-Level Detection Metrics**

In evaluating pixel-level manipulation detection, we calculated *Precision* and *Recall* for forgery pixel identification. To offer a comprehensive assessment of the model's effectiveness, we also report the F_1 score, which serves as the harmonic mean of *Precision* and *Recall*, thereby balancing the trade-offs between these two metrics. Below are the detailed evaluation criteria used in our experiments:

F_1 Score (Pixel-Level): The F_1 score is the harmonic mean of precision and recall, providing a balanced measure of the model's ability to detect forgery pixels. A higher F_1 score indicates better detection performance at the pixel level.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{17}$$

- **Comprehensive Evaluation Metrics**

Com-F_1: The Com-F_1 is the harmonic mean of pixel-level F_1 and image-level F_1, providing a comprehensive measure of the model's performance. Com-F_1 is sensitive to the lowest value of pixel-F_1 and image-F_1. In particular, it scores 0 when either pixel-F_1 or image-F_1 is 0, which does not hold for the arithmetic mean. A higher Com-F_1 score indicates that the model performs well in both pixel-level and image-level detection.

$$\text{Com-}F_1 = 2 \times \frac{\text{Pixel-level } F_1 \times \text{Image-level } F_1}{\text{Pixel-level } F_1 + \text{Image-level } F_1} \tag{18}$$

During evaluation, some papers [9,15,17] provide information on the optimal decision threshold, which can enable the model to achieve satisfactory performance in an ideal situation. However, in practice, the ideal situation does not always exist, since the decision

threshold should be preset. To make the evaluation closer to real-world situations, we set a default decision threshold (0.5) for F_1 computation at both the image level and the pixel level.

4.3. Ablation Study

In the ablation study, as listed in Table 3, setups #1 (ILD) and #2 (PLD) are used as two complete detection algorithms. While keeping the ResNet-50 structure of PLD and ILD unchanged, the output of ResNet-50 stage-4 is used as a side output. After undergoing up-sampling operations, the up-sampling result of the side output is used for the localization of forgery areas. In addition, a fully connected module, analogous to the ILD structure, is incorporated into setup #2 to facilitate image-level implementation. In setups #3–#5, "Dual branch" represents ILD and PLD. MIF, based on self-attention, is used to fuse the image features of the two branches, so in setups #4 and #5, the columns of "Self-attention" are labeled as "+". In setups #3–#5, if the columns of Dual Attention are labeled as "+", MSEFL uses Dual Attention to fuse the edge features of the two branches; otherwise, MSEFL just concatenates the edge features in the channel dimension.

From the configuration described above, it is evident that Setup #1 (ILD) and Setup #2 (PLD) serve as the baseline for two independent branches. Setup #3 incorporates MSEFL on the dual-branch framework without attention-based fusion. Setup #4 builds upon Setup #3 by integrating MIF based on self-attention. Finally, Setup #5 enhances Setup #4 by introducing trainable dual attention (tDA) for cross-branch edge fusion within MSEFL.

Table 3. Ablation study.

Setup	Component		Pixel-Level Evaluation (F_1)				Image-Level Evaluation		Com-F_1
	Self-Attention	Dual Attention	*cpmv.*	*spli.*	*inpa.*	Mean	AUC	F_1	
1: ILD	-	-	45.3	60.2	41.1	48.9	79.7	65.3	55.9
2: PLD	-	-	41.4	70.8	45.8	52.7	82.3	72.7	61.1
3: Dual Branch-MSEFL	-	-	44.8	71.9	46.2	54.3	84.8	74.1	62.7
4: Dual Branch-MSEFL-MIF	+	-	47.9	74.7	50.6	57.7	85.1	74.7	65.1
5: Dual Branch-MSEFL-MIF	+	+	49.5	75.4	51.9	58.9	89.1	81.3	68.3

Note: Copy–move, splicing, and inpainting are abbreviated as *cpmv.*, *spli.*, and *inpa.*, respectively. Self-attention refers to the MIF module; Dual Attention refers to tDA used in MSEFL for cross-branch edge fusion. All F_1 scores are computed at a fixed decision threshold of 0.5.

(1) Comparison between ILD and PLD. Comparing ILD and PLD in Table 3, ILD achieves a higher pixel-level F_1 score of 45.3 on *cpmv.*, whereas PLD demonstrates superior performance on *spli.* and *inpa.*, with respective F_1 scores of 70.8 and 45.8. Consequently, the average pixel-level F_1 score increases from 48.9 to 52.7 (+3.8), the image-level F_1 score improves from 65.3 to 72.7 (+7.4), and the comprehensive F_1 score (Com-F_1) rises from 55.9 to 61.1 (+5.2). For *cpmv.*, the forgery regions come from the original image, in which the noise distribution of the forgery image does not change significantly, so that comparing with PLD using the difference of noise distribution, ILD using the boundary artifacts achieves better performance. On the contrary, *spli.* and *inpa.* have a great difference in noise distribution, so PLD can achieve better results. Due to the quantity difference between *cpmv.* and *spli./ inpa.*, PLD outperforms ILD in the image-level evaluation.

(2) Influence of multi-branch fusion. MSEFL serves as the adhesive to fuse multiple side outputs of the two branches for multi-stream supervised learning. Without incorporating attention fusion, implementing a two-branch multi-side output framework for multi-stream edge supervision results in a pixel-level mean increase from 52.7 in setup #2 to 54.3 (+1.6), an image-level F_1 score improvement from 72.7 to 74.1 (+1.4), and a Com-F_1 score enhancement from 61.1 to 62.7 (+1.6). This indicates that even without employing attention mechanisms, edge supervision already demonstrated stable enhancements in both

image-level and pixel-level performance, thereby validating the effectiveness of mapping tampered boundaries across multi-scale and multi-source features.

(3) Influence of MIF. MIF uses self-attention to increase the difference between forged and authentic regions in PLD. Upon the incorporation of MIF in setup #4, compared with setup #3, a comprehensive enhancement in pixel-level F_1 scores across all three categories was observed: *cpmv.* increased from 44.8 to 47.9; *spli.* from 71.9 to 74.7; and *inpa.* from 46.2 to 50.6. This resulted in an average pixel-level improvement of +3.4 (from 54.3 to 57.7). Concurrently, the image-level F_1 score showed a marginal increase (from 74.1 to 74.7), while the Com-F_1 score experienced a more substantial rise of +2.4 (from 62.7 to 65.1). This indicates that the Color Perspective (ILD) and Noise Perspective (PLD) can form more discriminative tampering representations after being weighted and realigned by self-attention, with the enhancement primarily manifested in pixel-level localization accuracy.

(4) Influence of MSEFL with Dual Attention. MSEFL primarily processes the edge information of the side outputs from the corresponding stages of the two branches. It then uses the dual attention module to fuse the edge information from each stage of the two branches. Comparing with setup #4, the introduction of dual attention for cross-branch edge fusion in MSEFL in setup #5 leads to a continued enhancement in pixel-level mean, rising from 57.7 to 58.9 (+1.2). More notably, there was a significant leap in image-level AUC/F_1 metrics (AUC 85.1→89.1; F_1 74.7→81.3, +6.6), propelling the Com-F_1 from 65.1 to 68.3 (+3.2). It is the reason that tDA enhances the discriminability of "tampered boundary-context" along both spatial and channel dimensions, thereby fortifying image-level discrimination and exerting a positive pull on pixel-level localization.

In summary, a consistent and steady enhancement in pixel-level localization, image-level discrimination, and Com-F_1 is observed, progressing from the independent branches of setup #1 and #2 to the comprehensive structure of setup #5.

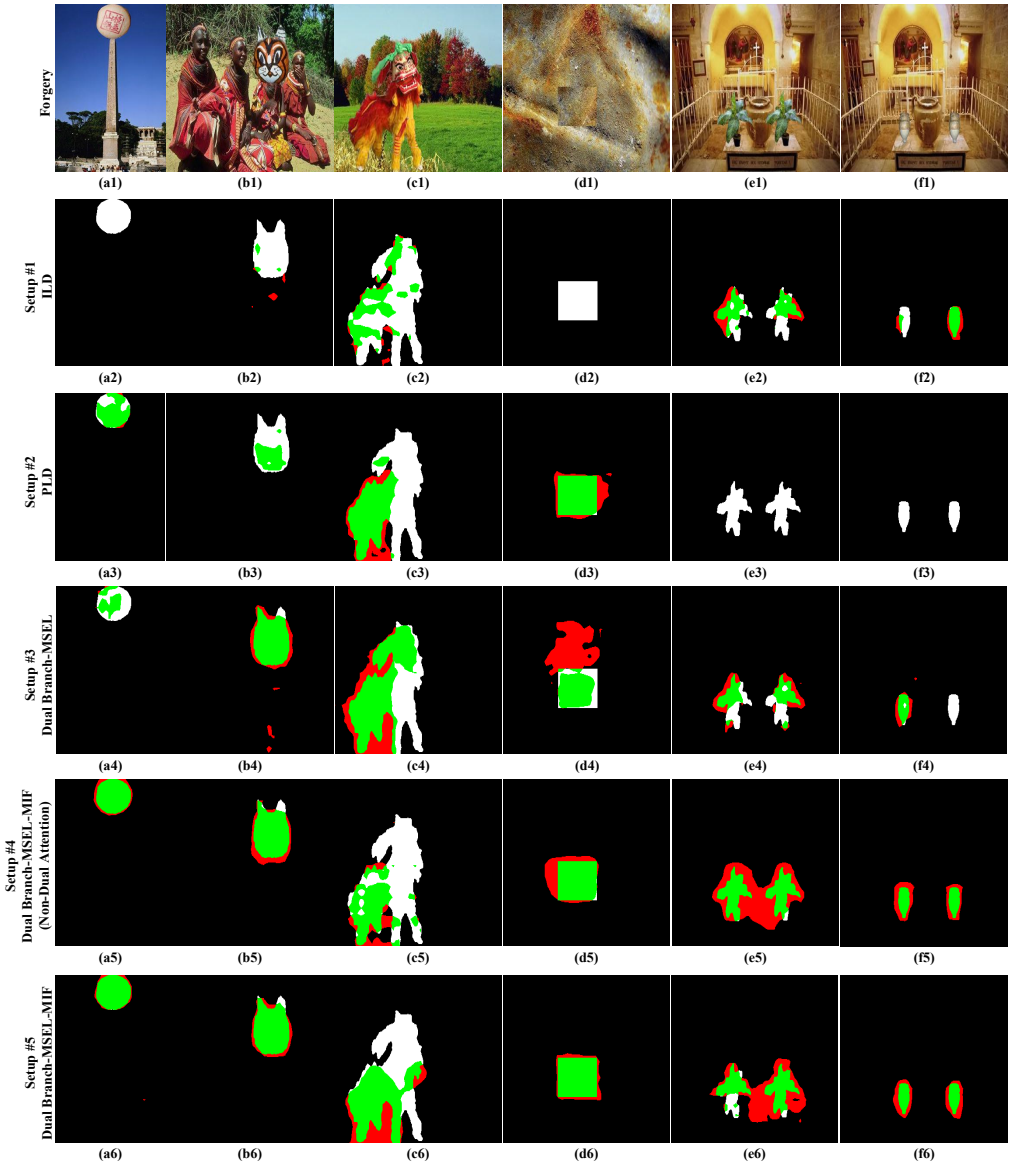
For more intuitively showing the detection effects of different setups, Figure 5 presents the results of these setups on several detection images at the pixel level. In this figure, there are the white, red, and green regions, which represent false negative, false positive, and true positive results, respectively. Below, we also use the same manner for result visualization. An outstanding detection performance should contain the areas with more green pixels but fewer white and red pixels. Compared with other setups, setups #1 and #2 have too many white pixels. Setup #3 detects more green pixels, but is accompanied by more red pixels, which makes a step forward. Setups #4 and #5 are better than the previous three. Setup #5 is the best, where it contains more green pixels, fewer red or white pixels.

4.4. Comparison with State-of-the-Art

We collected pixel-level forgery detection performance data for eight models across five datasets, including H-LSTM [14], ManTra-Net [15], HP-FCN [12], CR-CNN [17], GSR-Net [16], SPAN [9], CAT-Net [18], and MVSS-Net++ [2]. The experimental data for these models are all sourced from the paper by Dong et al. [2].

4.4.1. Pixel-Level Manipulation Detection

We also collected the source codes and corresponding training parameters of ManTra-Net [15] and CAT-Net [18], so we would show the detection results of the two algorithms and our proposed model, as shown in Figure 6. The meanings of the green, red, and white colors can refer to Section 4.3. In summary, the result with more green and less red or white is the best.



Note: The white, red, and green regions represent false negative, false positive, and true positive results, respectively, similarly hereinafter

Figure 5. The detection results of different setups. (**a1**–**f1**) is the detected forged images; (**a2**–**f2**) is the detected results of Setup #1 ILD, in corresponding to (**a1**–**f1**); (**a3**–**f3**) is the detected results of Setup #2 PLD, in corresponding to (**a1**–**f1**); (**a4**–**f4**) is the detected results of Setup #3 Dual Branch-MSEL, in corresponding to (**a1**–**f1**); (**a5**–**f5**) is the detected results of Setup #4 Dual Branch-MSEL-MIF (Non-Dual Attention), in corresponding to (**a1**–**f1**); (**a6**–**f6**) is the detected results of Setup #5 Dual Branch-MSEL-MIF, in corresponding to (**a1**–**f1**).

Observing Figure 6, ManTra-Net has too many incorrectly detected pixels, while the results of CAT-Net have too many forgery pixels that are not detected. Comparing with the two algorithms, the MMFD-Net obviously is better, which increases the ratio of detected forgery pixels while reducing false detections, thereby enhancing overall performance. This conclusion is further confirmed by the subsequent quantitative comparison.

Through the above qualitative comparison, we can intuitively see the detection effect of the algorithm. In the following, we verify the effectiveness of the algorithm through more quantitative comparisons.

In Table 4, the forgery detection performance of different models is evaluated using multiple datasets at the pixel level. Regarding evaluation metrics, we adopted the F1 score

of Pixel-Level Detection Metrics, where the best result on each dataset is highlighted with bold font. Meanwhile, using the mean value of F1 scores of six datasets, labeled as Mean, to comprehensively evaluate the overall performance of the model.

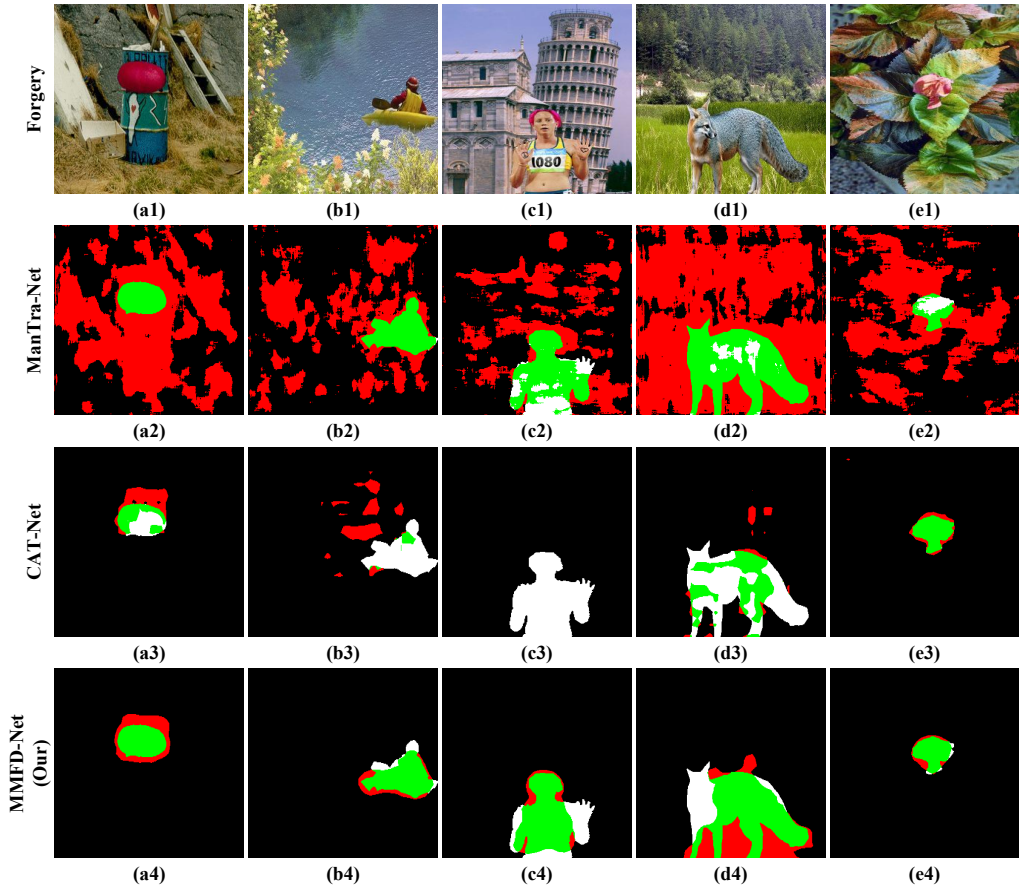


Figure 6. The examples of detection results using ManTra-Net, CAT-Net and MMFD-Net. (**a1**–**e1**) is the detected forged images; (**a2**–**e2**) is the detected results of ManTra-Net, in corresponding to (**a1**–**e1**); (**a3**–**e3**) is the detected results of CAT-Net, in corresponding to (**a1**–**e1**); (**a4**–**e4**) is the detected results of MMFD-Net, in corresponding to (**a1**–**e1**).

Table 4. Performance on Pixel-level Forgery Detection.

Method	Publication	NIST	CASIAv1	COVER	DEF-12k	IMD	Mean
H-LSTM	TIP19	**35.4**	15.4	16.3	5.9	19.5	18.5
ManTra-Net	CVPR19	0.0	15.5	28.6	**15.5**	18.7	15.66
HP-FCN	ICCV19	12.1	15.4	0.3	5.5	11.2	8.9
CR-CNN	ICME20	23.8	40.5	29.1	13.2	26.2	26.56
GSR-Net	AAAI20	28.3	38.7	28.5	5.1	24.3	24.98
SPAN	ECCV20	22.1	18.4	17.2	4.8	17.0	15.9
CAT-Net	WACV21	17.9	13.6	12.9	4.6	5.4	10.88
MVSS-Net++	TPAMI23	30.4	51.3	48.2	9.5	27.0	33.28
MMFD-Net (Our)		33.8	**53.7**	**50.4**	14.7	**29.8**	**36.48**

Note: Bold indicates the best performance on the corresponding dataset, similarly hereinafter.

In the COVER, CASIA v1, and IMD datasets, our proposed method achieves the best performance. In the NIST and DEF-12k datasets, we achieved the second-best performance, which was only 1.6% lower than H-LSTM and 0.8% lower than ManTra-Net, respectively. Compared with MVSS-Net++, the performance of our proposed method is almost the best in all datasets. The best performance of the ManTra-Net in DEK-12k is owed to the large-scale training data that comes from the COCO dataset. Compared with the models

training on the CASIA v2, i.e., MVSS-net++, GSR-Net, and CR-CNN, the proposed method is better than them on almost all test datasets, which also proves that our method has much better generalization in the different data settings. Overall, the proposed method performs the best in Mean values is to be expected.

4.4.2. Image-Level Manipulation Detection

In Table 5, we use four datasets to evaluate the image-level forgery detection performance of different models. Compared with Table 4, Table 5 does not use the NIST dataset to evaluate the performance of these models, because the NIST dataset does not provide authentic images for image-level assessment.

Table 5. Performance on image-level forgery detection.

Method	CASIAv1		COVER		DEF-12k		IMD	
	AUC	F_1	AUC	F_1	AUC	F_1	AUC	F_1
H-LSTM	0.498	0.0	0.500	0.0	0.499	0.2	0.501	0.0
ManTra-Net	0.500	0.0	0.500	0.0	0.543	0.0	0.500	0.0
CR-CNN	0.719	24.2	0.566	13.1	0.567	39.7	0.615	21.7
GSR-Net	0.500	0.0	0.515	0.0	0.456	0.2	0.500	0.0
SPAN	0.500	0.0	0.500	0.0	0.500	0.0	0.500	0.0
CAT-Net	0.647	38.0	0.557	41.5	0.543	46.5	0.586	41.1
MVSS-Net++	0.862	69.4	0.726	68.5	0.531	47.8	0.658	61.4
MMFD-Net (Our)	**0.873**	**71.5**	**0.739**	**69.7**	**0.618**	**48.7**	**0.673**	**63.2**

Note: Bold indicates the best performance on the corresponding dataset.

As listed in Table 5, the AUC of the proposed model is much closer to 1 in all four datasets, which means that the proposed model has a much greater ability to distinguish between the forgery and authentic images. Meanwhile, the F_1 scores of the proposed model are also the best among these compared models, which means that the proposed model has great performance in forgery image detection.

4.4.3. The Overall Performance

The overall performance, as measured by Com-F_1, which is computed from both pixel-level and image-level F_1 scores, is listed in Table 6. As shown in this table, the proposed model achieves the best performance, demonstrating that our method is more capable of adapting to real-world detection environments.

Table 6. The overall performance of four datasets.

Method	CASIAv1	COVER	DEF-12k	IMD	Mean
H-LSTM	0	0	0.4	0	0.1
ManTra-Net	0	0	0	0	0
CR-CNN	30.3	18.1	19.8	23.7	23.0
GSR-Net	0	0	0.4	0	0.1
SPAN	0	0	0	0	0
CAT-Net	20.0	19.7	8.4	9.5	14.4
MVSS-Net++	59.0	56.6	15.8	37.5	42.2
MMFD-Net (Our)	**61.3**	**58.5**	**22.6**	**40.5**	**45.7**

Note: Bold indicates the best performance on the corresponding dataset.

4.4.4. Comprehensive Analysis

MMFD-Net outperforms recent state-of-the-art methods such as MVSS-Net++ and CAT-Net. To elucidate why MMFD-Net surpasses these models, we analyze its key components and their contributions to the overall performance.

1. Multi-Stream Edge Feature Learning (MSEFL)

MMFD-Net introduces an MSEFL module that leverages both low-level edge features and high-level abstract features. By explicitly focusing on the boundaries of manipulated regions—where tampering traces are most likely to appear—the module enhances both detection and localization. Aggregating edge cues across multiple network stages enables the model to capture fine-grained as well as higher-level boundary information, which is particularly beneficial in challenging cases with subtle edges. Because edge cues are relatively content-agnostic, MSEFL also improves generalization across manipulation types and datasets, helping to mitigate overfitting.

2. Multi-Dimensional Information Fusion (MIF)

MMFD-Net employs an MIF module to integrate features from the color view and the noise view branches. Using a self-attention mechanism, the model dynamically reweights features and focuses on information most relevant to forgery detection. This fusion combines complementary cues—noise features are sensitive to inconsistencies introduced by tampering, while color features capture visual anomalies—yielding a richer and more comprehensive representation that improves discrimination between authentic and manipulated regions.

3. Joint Supervision Learning

We train MMFD-Net with joint supervision over multiple tasks, including image-level classification, pixel-level localization, and edge learning. This multi-task strategy encourages the model to learn semantics-agnostic forgery cues that are crucial for robust detection and generalization. By learning from multiple complementary objectives, the model excels at both pixel-level and image-level detection, reducing the risk of overfitting and improving performance across manipulation types and datasets.

Given the architectural similarities (both adopt color and noise branches and consider manipulation edges) of MMFD-Net and MVSS-Net++, we provide a focused comparison. MVSS-Net++ explores manipulated boundaries by applying a fixed Sobel operator to side outputs from multiple ResNet stages. As a handcrafted operator, Sobel is not adapted through learning and may be less flexible in capturing diverse manipulation patterns. In contrast, MMFD-Net learns edge features end-to-end, allowing the boundary extractor to adapt to the data distribution and better model complex, variable tampering artifacts.

The superior performance of MMFD-Net stems from the synergy among advanced edge-feature learning (MSEFL), multi-dimensional fusion (MIF), and joint supervision. Edge-centric learning is critical for reliable localization, while fusion and multi-task training further enhance robustness and versatility. Together, these components enable MMFD-Net to achieve state-of-the-art results in image forgery detection and localization.

4.4.5. Computational Complexity and Efficiency

We follow a deployment-oriented protocol: RTX 3090 (24 GB), CUDA 12, PyTorch 1.8, FP32, input resolution 512×512, and batch size one. Latency is measured with CUDA events and synchronization using 20 warm-up and 200 measured iterations. GPU memory is the peak allocated value obtained after resetting CUDA memory statistics. Model size is given in parameters (M) and computational cost in GFLOPs at 512^2 with a consistent counter.

Table 7 summarizes computational efficiency under our unified protocol. ManTra-Net is an extremely lightweight baseline (3.81 M parameters, 0.01 GFLOPs, 1.41 ms latency, 0.014 GB peak memory), but its detection accuracy in Tables 4–6 is weak across most datasets. CAT-Net is substantially heavier (114.26 M parameters, 59,907.14 GFLOPs), with a latency of 41.77 ms and 0.43 GB peak memory, and its overall Com-F1 remains limited.

Table 7. Computational complexity and efficiency.

Method	GFLOPs@512^2	Parameter (M)	Latency Mean (ms)	Peak Memory (GB)
ManTra-Net	0.01	3.81	1.41	0.014
CAT-Net	59,907.14	114.26	41.77	0.43
MMFD-Net (Our)	1667.97	154.59	30.90	0.58

Note: Latency is the per-image mean over 200 iterations after 20 warm-up runs (CUDA event timing, synchronized). Peak memory is the maximum allocated GPU memory during a single forward pass. GFLOPs are computed at 512^2 with a unified counter.

In contrast, MMFD-Net achieves a mean latency of 30.90 ms, which is 10.87 ms faster than CAT-Net (≈26.0% reduction), while using 0.58 GB peak memory (increasing 0.15 GB; ≈34.9%) and 154.59 M parameters (increasing 40.33 M; ≈35.3%). Notably, its computational count is 1667.97 GFLOPs, whereas CAT-Net reports 59,907.14 GFLOPs. Coupled with the accuracy advantages in Tables 4–6—MMFD-Net attains the highest mean Com-F_1 of 45.7—these results indicate a favorable cost–benefit trade-off: the multi-branch design introduces moderate memory/parameter overhead but delivers lower end-to-end latency and clearly superior detection performance.

For deployment, when accuracy and real-time performance are both required on a single GPU, MMFD-Net achieves a favorable accuracy–efficiency trade-off and is recommended as the default detector. When resources are severely constrained (edge/embedded scenarios), ManTra-Net can serve as a fast pre-filter, with positives rechecked by MMFD-Net. CAT-Net is heavier and slower under our setting while being less accurate; unless compatibility dictates otherwise, MMFD-Net is the better practical choice.

5. Conclusions

The proposed Multi-branch Multi-dimensional Forgery Detection Networks (MMFD-Net) effectively enhance the performance and generalization ability of digital image forgery detection by integrating image-level classification, pixel-level tampering region localization, and edge information. The design of MMFD-Net fully utilizes the advantages of the multi-branch structure. It enhances the model's perception of tampered regions through the Multi-dimensional Information Fusion (MIF) module and the Multi-Stream Edge Feature Learning (MSEFL) module, and further improves the robustness of the model through joint supervised learning. Experimental results demonstrate that MMFD-Net achieves excellent performance on multiple public datasets, especially in pixel-level and image-level detection tasks, where its comprehensive performance metric (Com-F_1) outperforms several existing state-of-the-art methods. Moreover, MMFD-Net demonstrates good generalization ability in handling complex scenarios and various types of forgery methods, proving its potential for practical applications.

Despite the commendable performance of MMFD-Net in the aforementioned experiments, its limitations persist in detecting highly compressed or low-resolution images, as well as forged images generated by GANs.

(1) Limitations in handling highly compressed and low-resolution images.

High compression ratios can markedly degrade discriminative image cues, making it challenging for any forgery detection model to identify manipulated regions. Like other deep learning models, MMFD-Net may be affected by detail loss and amplified artifacts introduced by heavy compression. Low-resolution images pose related difficulties, including reduced feature richness and coarse boundary evidence. Because the MSEFL module relies on edge cues, it can be harder to detect subtle boundaries at low resolution, which may diminish localization accuracy.

As potential remedies, future work may apply preprocessing such as super-resolution enhancement or denoising before inference, and/or train the model on more diverse datasets that explicitly include heavily compressed and low-resolution images to improve robustness to these conditions.

(2) Limitations in handling GAN-generated forgeries.

GANs can produce highly realistic forgeries that are difficult to detect. MMFD-Net, like other detectors, may struggle when inconsistencies are extremely subtle and when the adversarial generation process exploits model weaknesses to evade detection.

To address this, future work could incorporate adversarial training—i.e., training on GAN-generated forgeries—to improve the model's ability to recognize such manipulations.

Author Contributions: Conceptualization, H.Y., K.U. and J.W.; Software, J.W. and Z.G.; Validation, J.W.; Formal analysis, Z.G.; Investigation, H.Y. and K.U.; Data curation, Z.G.; Writing—original draft, H.Y.; Writing—review & editing, K.U. and J.W.; Visualization, Z.G.; Supervision, K.U. and J.W.. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: Jing Wang was employed by CEPREI Certification Body, Guangzhou CEPREI Certification Center Service Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Gafni, O.; Wolf, L. Wish you were here: Context-aware human generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7840–7849.
2. Dong, C.; Chen, X.; Hu, R.; Cao, J.; Li, X. Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 3539–3553. [CrossRef] [PubMed]
3. Bianchi, T.; Piva, A. Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1003–1017. [CrossRef]
4. Li, F.; Zhai, H.; Liu, T.; Zhang, X.; Qin, C. Learning Compressed Artifact for JPEG Manipulation Localization Using Wide-Receptive-Field Network. *ACM Trans. Multimed. Comput. Commun. Appl.* **2024**, *20*, 1–23. [CrossRef]
5. Xiao, B.; Wei, Y.; Bi, X.; Li, W.; Ma, J. Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering. *Inf. Sci.* **2020**, *511*, 172–191. [CrossRef]
6. 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), Munich, Germany, 8–14 September 2018; pp. 101–117.
7. He, Z.; Lu, W.; Sun, W.; Huang, J. Digital image splicing detection based on Markov features in DCT and DWT domain. *Pattern Recognit.* **2012**, *45*, 4292–4299. [CrossRef]
8. Anwar, M.A.; Tahir, S.F.; Fahad, L.G.; Kifayat, K. Image forgery detection by transforming local descriptors into deep-derived features. *Appl. Soft Comput.* **2023**, *147*, 110730. [CrossRef]
9. Hu, X.; Zhang, Z.; Jiang, Z.; Chaudhuri, S.; Yang, Z.; Nevatia, R. SPAN: Spatial pyramid attention network for image manipulation localization. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXI 16; Springer: Berlin/Heidelberg, Germany, 2020, pp. 312–328.
10. Salloum, R.; Ren, Y.; Kuo, C.C.J. Image splicing localization using a multi-task fully convolutional network (MFCN). *J. Vis. Commun. Image Represent.* **2018**, *51*, 201–209. [CrossRef]
11. 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, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1053–1061.
12. Li, H.; Huang, J. Localization of deep inpainting using high-pass fully convolutional network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8301–8310.
13. Xiang, Y.; Zhao, K.; Yu, Z.; Yuan, X.; Huang, G.; Tian, J.; Li, J. DFFormer: Capturing Dynamic Frequency Features to Locate Image Manipulation through Adaptive Frequency Transformer and Prototype Learning. *IEEE Trans. Circuits Syst. Video Technol.* **2025**, 1. [CrossRef]

14. 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 Trans. Image Process.* **2019**, *28*, 3286–3300. [CrossRef] [PubMed]
15. 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, Long Beach, CA, USA, 15–20 June 2019; pp. 9543–9552.
16. Zhou, P.; Chen, B.C.; Han, X.; Najibi, M.; Shrivastava, A.; Lim, S.N.; Davis, L. Generate, segment, and refine: Towards generic manipulation segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13058–13065.
17. Yang, C.; Li, H.; Lin, F.; Jiang, B.; Zhao, H. Constrained R-CNN: A general image manipulation detection model. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6.
18. Kwon, M.J.; Yu, I.J.; Nam, S.H.; Lee, H.K. CAT-Net: Compression artifact tracing network for detection and localization of image splicing. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 375–384.
19. Han, R.; Wang, X.; Bai, N.; Wang, Y.; Hou, J.; Xue, J. HDF-Net: Capturing homogeny difference features to localize the tampered image. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 10005–10020. [CrossRef] [PubMed]
20. Huang, Y.; Bian, S.; Li, H.; Wang, C.; Li, K. DS-UNet: A dual streams UNet for refined image forgery localization. *Inf. Sci.* **2022**, *610*, 73–89. [CrossRef]
21. Niu, Y.; Chen, P.; Zhang, L.; Tan, L.; Chen, Y. Image Forgery Localization via Guided Noise and Multi-Scale Feature Aggregation. *arXiv* **2024**, arXiv:2412.01622.
22. Zhu, J.; Li, D.; Fu, X.; Yang, G.; Huang, J.; Liu, A.; Zha, Z.J. Learning discriminative noise guidance for image forgery detection and localization. In Proceedings of the AAAI Conference on Artificial Intelligence, Singapore, 20–27 January 2024; Volume 38, pp. 7739–7747.
23. Wang, J.; Wu, Z.; Chen, J.; Han, X.; Shrivastava, A.; Lim, S.N.; Jiang, Y.G. Objectformer for image manipulation detection and localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 2364–2373.
24. Kwon, M.J.; Nam, S.H.; Yu, I.J.; Lee, H.K.; Kim, C. Learning jpeg compression artifacts for image manipulation detection and localization. *Int. J. Comput. Vis.* **2022**, *130*, 1875–1895. [CrossRef]
25. Guillaro, F.; Cozzolino, D.; Sud, A.; Dufour, N.; Verdoliva, L. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 20606–20615.
26. Xiang, Y.; Zhao, K.; Yin, H. SCCA-Net: A Novel Network for Image Manipulation Localization Using Split-Channel Contextual Attention. In Proceedings of the Computer Vision—ACCV 2024, Hanoi, Vietnam, 8–12 December 2024; Cho, M., Laptev, I., Tran, D., Yao, A., Zha, H., Eds.; Springer: Singapore, 2025; pp. 220–234.
27. Hao, Q.; Ren, R.; Niu, S.; Wang, K.; Wang, M.; Zhang, J. UGEE-Net: Uncertainty-guided and edge-enhanced network for image splicing localization. *Neural Netw.* **2024**, *178*, 106430. [CrossRef] [PubMed]
28. Lin, X.; Wang, S.; Deng, J.; Fu, Y.; Bai, X.; Chen, X.; Qu, X.; Tang, W. Image manipulation detection by multiple tampering traces and edge artifact enhancement. *Pattern Recognit.* **2023**, *133*, 109026. [CrossRef]
29. Sun, Z.; Jiang, H.; Wang, D.; Li, X.; Cao, J. Safl-net: Semantic-agnostic feature learning network with auxiliary plugins for image manipulation detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 22424–22433.
30. Ma, X.; Du, B.; Jiang, Z.; Hammadi, A.Y.A.; Zhou, J. IML-ViT: Benchmarking Image Manipulation Localization by Vision Transformer. *arXiv* **2023**, arXiv:2307.14863.
31. Xiang, Y.; Yuan, X.; Zhao, K.; Liu, T.; Xie, Z.; Huang, G.; Li, J. Image Manipulation Localization Using Dual-Shallow Feature Pyramid Fusion and Boundary Contextual Incoherence Enhancement. *IEEE Trans. Emerg. Top. Comput. Intell.* **2025**, *9*, 2858–2868. [CrossRef]
32. Bayar, B.; Stamm, M.C. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2691–2706. [CrossRef]
33. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
34. Wei, Q.; Li, X.; Yu, W.; Zhang, X.; Zhang, Y.; Hu, B.; Mo, B.; Gong, D.; Chen, N.; Ding, D.; et al. Learn to segment retinal lesions and beyond. In Proceedings of the 2020 25th International conference on pattern recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 7403–7410.
35. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
36. Dong, J.; Wang, W.; Tan, T. Casia image tampering detection evaluation database. In Proceedings of the IEEE China Summit and International Conference on Signal and Information Processing, Beijing, China, 6–10 July 2013; pp. 422–426.

37. Wen, B.; Zhu, Y.; Subramanian, R.; Ng, T.T.; Shen, X.; Winkler, S. COVERAGE—A novel database for copy-move forgery detection. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 161–165.
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 Proceedings of the IEEE Winter Applications of Computer Vision Workshops (WACVW), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 63–72.
39. 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, Montreal, BC, Canada, 11–17 October 2021; pp. 14185–14193.
40. Mahfoudi, G.; Tajini, B.; Retraint, F.; Morain-Nicolier, F.; Dugelay, J.L.; Pic, M. Defacto: Image and face manipulation dataset. In Proceedings of the 2019 27th European Signal Processing Conference (EUSIPCO), A Coruña, Spain, 2–6 September 2019; pp. 1–5.
41. 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 Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.