Edge-aware Regional Message Passing Controller for Image Forgery Localization

Dong Li*, Jiaying Zhu*, Menglu Wang, Jiawei Liu†, Xueyang Fu, Zheng-Jun Zha
University of Science and Technology of China, China
{dongli6, zhujy53, vault}@mail.ustc.edu.cn, {jwliu6, xyfu, zhazj}@ustc.edu.cn

Abstract

Digital image authenticity has promoted research on image forgery localization. Although deep learning-based methods achieve remarkable progress, most of them usually suffer from severe feature coupling between the forged and authentic regions. In this work, we propose a two-step Edge-aware Regional Message Passing Controlling strategy to address the above issue. Specifically, the first step is to account for fully exploiting the edge information. It consists of two core designs: context-enhanced graph construction and threshold-adaptive differentiable binarization edge algorithm. The former assembles the global semantic information to distinguish the features between the forged and authentic regions, while the latter stands on the output of the former to provide the learnable edges. In the second step, guided by the learnable edges, a region message passing controller is devised to weaken the message passing between the forged and authentic regions. In this way, our ERMPC is capable of explicitly modeling the inconsistency between the forged and authentic regions and enabling it to perform well on refined forged images. Extensive experiments on several challenging benchmarks show that our method is superior to state-of-the-art image forgery localization methods qualitatively and quantitatively.

1. Introduction

The forged or manipulated images pose risks in various fields, such as removing copyright watermarks, generating fake news, and even falsifying evidence in court [32]. The growing technology of forgery will cause a crisis of trust and affect social equity. Therefore, the detection of image forgery is of great significance. The crucial aspect of the detection is to model the inconsistency between the forged and authentic regions and locate the forged regions on the suspicious image, i.e., image forgery localization (IFL). However, as post-processing techniques such as GAN [16, 26, 63], VAE [27, 44] and homogeneous manipulation [7, 33] are wildly utilized, images can be easily tampered in a visually imperceptible way. These techniques constantly aim to couple the forged and authentic regions’ features, making image forgery localization challenging. Therefore, in order to accurately locate the image forgery region, it is particularly essential to decouple the features between forged and authentic regions.

In recent years, deep-learning techniques have attracted more attention [23, 57, 58, 65, 66]. Due to the development of deep learning, image forgery localization has achieved remarkable results. For example, ManTra-Net [55] treats the forgery localization problem as a local anomaly detection problem and proposes a novel long short-term memory solution to assess local anomalies. To discriminate between heterologous regions, SPAN employs CNNs to extract anomalous local noise features from noise maps. MVSS-Net [5] learns a multiview feature with multi-scale...
supervised networks to jointly exploit the noise view and the boundary artifacts. However, these methods do not decouple the features between the forged and authentic regions, making it difficult to locate the tampered regions for elaborately forged images accurately. As shown in Figure 1, in the previous method, the features of the forged region are coupled with some of the features of the authentic region, resulting in wrong localization.

In this work, we propose a novel method to avoid feature coupling of the two regions (forged and authentic) for image forgery localization. One of the keys of this method is to construct a dynamic graph, where the edges between the forged and authentic regions participate in its construction. We control the message passing of regions inside and outside the edges (i.e., forged and authentic) by reconstructing the adjacency matrix of nodes inside and outside the edges, thus achieving effective disentanglement of the features between the forged and authentic regions. Based on its functionality, the edge-aware dynamic graph convolution is named the regional message passing controller (RMPC).

In order to use the proposed RMPC for image forgery localization, it is necessary to obtain the edge information between the forged and authentic regions, which is the other key of this method. Therefore, an edge reconstruction (ER) module was developed, including a context-enhanced graph (CEG) and a threshold-adaptive differentiable binarization module. We specially design an adjacency matrix learner in the CEG, which encodes global information along the nodes to assemble the global semantic information. Inspired by the Sigmoid function, we develop the threshold-adaptive differentiable binarization edge algorithm, which stands on the output of the CEG to provide the learnable edges.

In summary, in this work we propose a new two-step framework named Edge-aware Regional Message Passing Controller (ERMPC) for Image Forgery Localization, including RMPC and ER. The ERMPC could effectively control the message passing between the forged and authentic regions to achieve effective disentanglement of the two regions, thus boosting the performance of image forgery localization. We take edge information as the main task and use it as a basis to explicitly model the inconsistency between two regions. To the best of our knowledge, this work is the first attempt to explicitly weaken the message passing between the forged and authentic regions. Our contributions are as follows:

- We propose ERMPC, a novel two-step coarse-to-fine framework for image forgery localization, explicitly modeling the inconsistency between the forged and authentic regions with edge information.

- We propose an edge-aware dynamic graph, also known as RMPC, to control the message passing between two regions (forged and authentic) in the feature map.

- We develop an edge reconstruction module containing a context-enhanced graph and a threshold-adaptive differentiable binarization module to obtain the desired edge information.

- We conduct extensive experiments on multiple benchmarks and demonstrate that our method is qualitatively and quantitatively superior to state-of-the-art image forgery localization methods.

2. Related Works

2.1. Image Forgery Localization

Most early works propose to localize a specific type of forgery, including splicing [2, 3, 9, 10, 24, 28, 37, 51, 60], copy-move [8, 14, 25, 48, 52, 54], and removal [1, 50, 56, 64]. While these methods demonstrate satisfactory performance in detecting specific types of forgery, they exhibit limitations in practical applications due to the prevalence of unknown and diverse forgery types. Consequently, recent studies have emphasized the need for an approach to tackle multiple forgery types with one model. RGB-N [62] proposes a dual-stream Faster R-CNN network. The first stream is designed to extract RGB features and identify tampering artifacts, while the second stream utilizes noise features to model the noise inconsistency between tampered and authentic regions, thereby localizing image forgery with enhanced accuracy. ManTra-net [55] is an end-to-end network that performs both detection and localization, which treats the problem as anomaly detection and introduces a long short-term memory solution to assess local anomalies. SPAN [22] attempts to model the spatial correlation by constructing a pyramid of local self-attention blocks. MVSS-Net [5] has designed an edge-supervised branch that uses edge residual blocks to capture fine-grained boundary detail in a shallow to deep manner. It is worth noting that edge information is important for image forgery detection and localization, as the tampered regions are commonly surrounded by unnatural artifacts. Nevertheless, most studies only utilize edges in a supervised strategy, such as MVSS-Net [5], MFCN [41], GSR-Net [61], and CAIFL [47]. PSCCNet [35] uses a progressive spatial-channel correlation module that uses features at different scales and dense cross-connections to generate operational masks in a coarse-to-fine fashion. ObjectFormer [46] captures forgery traces by extracting high-frequency parts of the image and combining them with RGB features. In this work, we explicitly employ learnable edges as guiding information to impede the message passing between the forged and untouched regions and employ refined edge reconstruction to achieve inconsistent modeling of the two regions to localize manipulation artifacts.
2.2. Graph Reasoning

Graph-based approaches have attracted growing attention from the computer vision community these days and have shown to be an effective relational reasoning method with a robust capacity for non-local feature aggregation. Specifically, CDGCNet [21] utilizes a class-wise learning strategy and designs a class-wise graph convolution, which avoids heavy graph concatenation and facilitates feature learning. To overcome the problem that most graph-based approaches assign a fixed number of neighbors for each query item, DAGL [39] proposes a dynamic attentive graph learning model to explore the dynamic non-local property for image restoration. CTL [34] learns a context-reinforced topology to construct multi-scale graphs which have a robust representational capacity by considering both global contextual information and physical connections of the human body. On other tasks in computer vision, e.g., object detection [43], multi-label image recognition [6], and skeleton-based action recognition [36, 59], graph convolutional neural networks have also achieved impressive performance. In contrast, we propose an improved edge-guided graph attention model to obstruct the message passing between forged and untouched regions with dynamic construction for forgery region localization. As graph convolution is an effective relational reasoning method that is very suitable for detecting forgery traces, this paper applies it to the task of image forgery localization for the first time. An improved edge-guided graph attention model is proposed for forgery region localization, which obstructs the message passing between forged and untouched regions with dynamic construction.

3. Methodology

This section details the scheme of image forgery localization based on edge-aware message passing control. Sec. 3.1 describes an overview of the framework. One of the keys of the scheme is to control the message passing with edge and thus model the inconsistency between the forged and authentic regions (Sec. 3.2). Another key lies in reconstructing accurate edges from coarse features (Sec. 3.3). In addition, following [5, 22, 55, 62], we use the noise branch and fuse it with RGB branch at the end of the network (Sec. 3.4). The optimization is introduced in Sec. 3.5.
3.1. Overview

Figure 2 is an overview of the framework. The input image is represented as \( X \in \mathbb{R}^{H \times W \times 3} \), where \( H \) and \( W \) represent the height and width of the image. First, we use two branches to handle RGB and noise separately, obtaining \( G_r \in \mathbb{R}^{H \times W \times C_r} \) and \( G_n \in \mathbb{R}^{H \times W \times C_n} \) respectively. Then, we use ResNet-50 pretrained on ImageNet [11] as the backbone network. Following [4], we adopt atrous spatial pyramid pooling(ASPP) together with ResNet-50 to capture the long-range contextual information. The coarse features extracted from RGB branches are converted into edges through the edge reconstruction block. Meanwhile, the coarse features are constructed as graph structures, guided by the reconstructed edge information. Finally, the RGB features after the graph convolutional network are fused with noise information through dual attention [5] to output the predicted forgery localization map.

3.2. Region Message Passing Controller

Most forged images are carefully processed to hide tampering artifacts, making it challenging to model inconsistencies in the RGB branch. To overcome this problem, the edges between the forged and authentic regions are utilized to control the message passing explicitly.

The edge feature \( G_e \in \mathbb{R}^{H_e \times W_e \times 1} \) comes from the edge reconstruction block (described in detail in the next subsection), where \( H_e = H_s \) and \( W_e = W_s \). First, we calculate the relationship between two node features \( P_i, P_j \) of the \( G_E \) using an algorithm similar to XNOR gate:

\[
XN(P_i, P_j) = \begin{cases} 
0 & \text{Inside and outside the edges} \\
1 & \text{On the same side of edges.}
\end{cases}
\]  

If two nodes are inside and outside the edge respectively, then their \( XN \) is set to 0. For each of the \( N = H_e \times W_e \) nodes features, we calculate its \( XN \), thus generating a matrix \( A_e \in \mathbb{R}^{N \times N} \).

Next, we apply graph learning to process the \( G_r \in \mathbb{R}^{H_r \times W_r \times C_r} \). Following GAT [45], we calculate the similarity between two nodes as attention coefficients:

\[
\alpha_{i,j} = \psi(x_i)^T \psi'(x_j),
\]

where \( \psi, \psi' \) denotes two learnable linear transformations. Specifically, we utilize \( \psi = WX \) and \( \psi' = W'x \), where \( W \in \mathbb{R}^{C_r \times C_r} \) and \( W' \in \mathbb{R}^{C_r \times C_r} \) are both weight matrices. To make coefficients easier to compare across different nodes, we normalize them using the softmax function:

\[
A_{r_{ij}} = \frac{\exp(\alpha_{i,j})}{\sum_{j=1}^{N} \exp(\alpha_{i,j})},
\]

where \( A_r \in \mathbb{R}^{N \times N} \) is the preliminary adjacency matrix. It reflects the relationship between any two nodes in the feature map. Larger values represent a greater flow of information between two nodes. In order to realize message passing control better, a method of dynamically adjusting the adjacency matrix is adopted. Specifically, if two nodes are on and off the boundary respectively, their adjacency is broken due to the dynamic adjustment of the adjacency matrix. In practice, the adjacency matrix is recalculated as follows:

\[
A'_r = A_r \odot A_e,
\]

where \( \odot \) is the Hadamard product. \( A'_r \in \mathbb{R}^{N \times N} \) re-models the weights between nodes in the feature map and severs the connections between the forged and real regions.

Once the adjacency matrix is obtained, it is weighted by the learnable attention weights. Then the original nodes are updated with the following:

\[
Z_r = \text{ReLU}(A'_r G'_r W_z),
\]

where \( Z_r \in \mathbb{R}^{N \times C_r} \) is RGB features after graph reasoning, \( W_z \in \mathbb{R}^{C_r \times C_r} \) is the learnable parameter and \( G'_r \in \mathbb{R}^{N \times C_r} \) is the graph representation transformed by \( G_r \).

It is worth noting that this work not only introduces the idea of controlling the message passing in image forgery localization for the first time, but also the approach implemented is different from previous research [12, 20]. BFP [12] uses directed acyclic graphs (DAG) for feature propagation and introduces boundary information into the propagation to control the message passing between different segments. However, it has to scan the image pixel by pixel and requires a lot of loops, causing it difficult to implement in real-world applications. BGC [20] emphasizes reducing the weights of the edges and does not really focus on the message passing between regions.

3.3. Edge Reconstruction

As described in the previous section, the Edge-aware Message Passing Control Graph requires accurate edge information. Designing such an edge-access network is non-trivial. The main challenge is how to learn edge information from the coarse feature. To this end, a novel approach to edge reconstruction is proposed. Specifically, we started by using the Sobel layer [5] to enhance the edge-related patterns in \( G_r \in \mathbb{R}^{H_r \times W_r \times C_r} \):

\[
G_c = G_r \odot \sigma \left( \text{Norm} \left( \text{Sobel}(G_r) \right) \right),
\]

where \( \text{Norm} \) is L2 Normalization, \( \sigma \) is Sigmoid and Sobel is the SobelConv [5].

Then, we explore a Context-Enhanced Graph (CEG), extracting the local and global features separately for \( G_c \in \mathbb{R}^{H_c \times W_c \times C_c} \). Specifically, the local information is extracted by the convolutional layers [5]. For global information, the contextual information of the feature map is encoded as the adjacency matrix \( A_c \in \mathbb{R}^{N \times N} \) in a simple and
L2 Normalization is applied to each row of the feature vector into the $N \times N$ adjacency matrix. In addition, L2 Normalization is applied to each row of $A_c$ to facilitate stable optimization. Next, we use this adjacency matrix to complete the graph reasoning. In this way, we can obtain a global feature map. The process can be written as

$$A_c = \text{Norm}(\text{Conv}(\text{Conv}(G'_c))),$$

where $G'_c \in \mathbb{R}^{N \times C_e}$ is reshaped by $G_c$. Given the node features of $G'_c$, we first squeeze the feature dimensions by a $1 \times 1$ convolution layer. Then, another convolution layer with $1 \times 1$ kernel is used to transfer the $N$-dimension feature vector into the $N \times N$ adjacency matrix. In addition, L2 Normalization is applied to each row of $A_c$ to facilitate stable optimization. Next, we use this adjacency matrix to complete the graph reasoning. In this way, we can obtain a global feature map. The process can be written as

$$\text{Global}(G'_c) = A_cG'_cW_c,$$

where $W_c \in \mathbb{R}^{C_e \times C_s}$ is the learnable parameter. To match with the local information dimension, we reshape $\text{Global}(G'_c) \in \mathbb{R}^{N \times C_e}$ back to $\text{Global}'(G'_c) \in \mathbb{R}^{H_s \times W_s \times C_s}$. Then combining the local and global information, we can obtain the edge probability map $G_p \in \mathbb{R}^{H_s \times W_s \times 1}$,

$$G_p = \sigma(\mathcal{C}(\text{Cat}(\text{Local}(G_c), \text{Global}'(G'_c)))),$$

where $\sigma$ denotes the Sigmoid, $\mathcal{C}$ is a $1 \times 1$ convolution layer, and $\text{Local}$ contains ReLU and two convolutional layers.

Furthermore, in order to determine the edges, a threshold needs to be determined to binarize the probability map. Most previous studies have used fixed thresholds, and the process is not differentiable. Inspired by [30], we utilize a variable threshold map which is adaptive for each point on the probability map $G_p$. Besides, we propose a variant of the Sigmoid function to complete the binarization, which is capable of participating in the backpropagation involved. Binarize threshold adjustment is achieved by translating the Sigmoid function along the x-axis. Therefore, we explore the threshold-adaptive differentiable binarization (TDB) for edge reconstruction. It is computed as follows:

$$G_c = \frac{1}{1 + e^{-k(G_p - \tau(G_p))}},$$

where $\tau$ denotes the learnable transformation, which in practice is $3 \times 3$ convolution operators, and $k$ is the amplifying factor. In particular, $k$ is set to 500 empirically.

### 3.4. Branch Fusion

Following most studies [5,22,55,62], we also employ the noise branch. However, this is not the focus of this work, so we have used some common methods. As in Figure 2, the noise is extracted with BayarConv [55]. For the fusion of the two branches, we follow [5] and adopt the Dual Attention (DA) [15]. DA includes channel attention module (CAM) and position attention module (PAM). It can effectively fuse two branches. The process can be written as

$$G_o = DA(G_z, G_n),$$

where $G_z \in \mathbb{R}^{H_s \times W_s \times C_s}$ is reshaped by the $Z$. Finally, we transform $G_o \in \mathbb{R}^{H_s \times W_s \times 1}$ with bilinear upsampling into the final predicted mask $G_{out} \in \mathbb{R}^{H \times W \times 1}$.

### 3.5. Optimization

As shown in Figure 2, we calculate the loss function for three components: the final prediction $G_{out} \in \mathbb{R}^{H \times W \times 1}$, the binary edge prediction $G_e \in \mathbb{R}^{H \times W \times 1}$ and the edge probability map $G_p \in \mathbb{R}^{H_s \times W_s \times 1}$. For edge loss, the ground-truth edge $\mathcal{E} \in \mathbb{R}^{H \times W \times 1}$ is downsampled to a smaller size $\mathcal{E}' \in \mathbb{R}^{H_s \times W_s \times 1}$ to match $G_e, G_p$. This strategy outperforms upsampling $G_e, G_p$ in terms of computational cost and performance. The overall loss function can be written as:

$$\mathcal{L} = \lambda_1 \mathcal{L}(Y, G_{out}) + \lambda_2 \mathcal{L}(\mathcal{E}', G_e) + \lambda_3 \mathcal{L}(\mathcal{E}', G_p),$$

where $\mathcal{L}$ denotes the Dice loss [5], $Y \in \mathbb{R}^{H \times W \times 1}$ is the ground-truth mask, and $\lambda_1, \lambda_2, \lambda_3$ are the parameters to balance the three terms in loss function ($\lambda_1 + \lambda_2 + \lambda_3 = 1$). In our setting, they are set as 0.50, 0.25 and 0.25 respectively.

### 4. Experiments

#### 4.1. Experimental Setup

**Pre-training Data** We create a sizable image tampering dataset and use it to pre-train our model. This dataset includes three categories: 1) splicing, 2) copy-move, 3) removal. For splicing, we use the MS COCO [31] to generate spliced images, where one annotated region is randomly selected per image and pasted into a different image after several transformations. We use the same transformation as [5].
including the scale, rotation, shift, and luminance changes. Since the spliced region is not always an object, we create random outlines using the Bezier curve [38] and fill them to create splicing masks. For copy-move, the datasets from MS COCO and [53] are adopted. For removal, we adopt the SOTA inpainting method [29] to fill one annotated region that is randomly removed from each chosen MS COCO image. We randomly add Gaussian noise or apply the JPEG compression algorithm to the generated data to resemble the visual quality of images in real situations.

**Testing Datasets** Following [46], we evaluate our model on CASIA [13] dataset, Coverage [49] dataset, Columbia [19] dataset, NIST16 [17] dataset and IMD20 [40] dataset. Specifically, CASIA [13], which contains two types of tampered images (splicing and copy-move), is widely used in the image forgery field. The database V1.0 has fixed size tampered images, which are generated only by using crop-and-paste operation under Adobe Photoshop. The database v2.0 is more comprehensive and challenging, and most tampered examples are generated with post-processing. COVER [49] provides 100 images, and all of them are generated by copy-move tampering technique. Columbia [19] consists of 180 splicing images, whose size ranges from $757 \times 568$ to $1152 \times 568$. In detail, the tampered images in Columbia are all uncompressed and without any post-processing. NIST16 [17] is a high-quality dataset. This dataset contains three types of tampering, and some regions of manipulation are difficult for humans to identify. IMD20 [40] collects real-life manipulated images from the Internet, and involves all three manipulations as well. We apply the same training/testing splits as [22, 46, 62] to fine-tune our model for fair comparisons.

**Evaluation Metrics** To quantify the localization performance, following previous works [22, 46], we use pixel-level Area Under Curve (AUC) and F1 score on manipulation masks. And we adopt the Equal Error Rate (EER) threshold to binarize masks because binary masks are necessary to calculate F1 scores.

**Implementation Details** The input images are resized to $512 \times 512$. In this work, the backbone is ResNet-50 [18], pre-trained on ImageNet [11]. Implemented by PyTorch, our model is trained with GeForce GTX 3090. We use Adam as the optimizer, and the learning rate decays from $10^{-4}$ to $10^{-7}$. We train 100 epochs with a batch size of 8, and the learning rate decays by 10 times every 30 epochs.

### 4.2. Comparison with the State-of-the-Art Methods

Following SPAN [22] and ObjectFormer [46], our model is compared with other state-of-the-art tampering localization methods under two settings: 1) training on the synthetic dataset and evaluating on the full test datasets, and 2) fine-tuning the pre-trained model on the training split of test datasets and evaluating on their test split. The pre-trained model will demonstrate each method’s generalizability, and the fine-tuned model will demonstrate how well each method performs locally once the domain discrepancy has been significantly reduced.

**Pre-trained Model** Table 1 shows the localization performance of pre-trained models for different methods on four standard datasets under pixel-level AUC. We compare our model ERMPC with MantraNet [55], SPAN [22], PSCCNet [35], and ObjectFormer [46] when evaluating pre-trained models. The pre-trained ERMPC achieves the best localization performance on Coverage, CASIA, NIST16 and IMD20, and ranks second on Columbia. Especially, ERMPC achieves 94.4% on the copy-move dataset.
Figure 4. Visualization of the predicted manipulation mask by different methods. From top to bottom, we show forged images, GT masks, predictions of ManTraNet, SPAN, PSCC-Net, and ours.

<table>
<thead>
<tr>
<th>Variants</th>
<th>CASIA AUC</th>
<th>CASIA F1</th>
<th>NIST16 AUC</th>
<th>NIST16 F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>71.6</td>
<td>38.3</td>
<td>77.1</td>
<td>52.6</td>
</tr>
<tr>
<td>w/o RMPC</td>
<td>76.9</td>
<td>45.6</td>
<td>86.4</td>
<td>60.7</td>
</tr>
<tr>
<td>w/o CEG</td>
<td>85.1</td>
<td>51.5</td>
<td>93.4</td>
<td>75.3</td>
</tr>
<tr>
<td>w/o TDB</td>
<td>88.6</td>
<td>57.3</td>
<td>98.2</td>
<td>81.9</td>
</tr>
<tr>
<td>Ours</td>
<td>90.4</td>
<td>58.6</td>
<td>99.7</td>
<td>83.6</td>
</tr>
</tbody>
</table>

Table 4. Ablation results on CASIA and NIST16 dataset using different variants of ERMPC. AUC and F1 scores (%) are reported.

COVER, whose image forgery regions are indistinguishable from the background. This validates our model owns the superior ability to control message passing between two regions (forged and authentic). We fail to achieve the best performance on Columbia, falling behind PSCCNet 1.4% under AUC. We contend that the explanation may be that the distribution of their synthesized training data closely resembles that of the Columbia dataset. This is further supported by the results in Table 2, which show that ERMPC performs better than PSCCNet in terms of both AUC and F1 scores. Furthermore, it is important to note that ERMPC attains decent results with less pre-training data.

**Fine-tuned Model** The fine-tuned models, which are initiated with the network weights of the pretrained model, will be trained on the training split of Coverage, CASIA, and NIST16 datasets, respectively. We evaluate the fine-tuned models of different methods in Table 2. As for AUC and F1, our model achieves significant performance gains. This validates that ERMPC could capture subtle tampering artifacts by controlling the message passing between two regions (forged and authentic) in the feature map.

**4.3. Robustness Evaluation**

We degrade the raw manipulated images from NIST16 using the distortion settings in [46] to analyze the robustness of ERMPC for forgery localization. These distortions types include scaling images to various scales (Re-
size), applying Gaussian blur with a kernel size $k$ (GaussianBlur), adding Gaussian noise with a standard deviation $\sigma$ (GaussianNoise), and performing JPEG compression with a quality factor $q$ (JPEGCompress). We compare the forgery localization performance (AUC scores) of our pre-trained models with SPAN [22] and ObjectFormer on these corrupted data and report the results in Table 3. ERMPC demonstrates more resistance to various distortion techniques. It is worth noting that JPEG compression is commonly performed when uploading images to social media. Our model ERMPC performs significantly better on compressed images than other methods.

4.4. Ablation Analysis

The Region Message Passing Controller (RMPC) module of our method ERMPC is designed to weaken the message passing between the forged and authentic regions. The Context-Enhanced Graph (CEG) encodes global information along the node to obtain a better edge probability map, while the Threshold-adaptive Differentiable Binarization (TDB) adaptively performs the binarization process for learnable edges. In order to evaluate the effectiveness of RMPC, CEG and TDB, we remove them separately from ERMPC and evaluate the forgery localization performance on CASIA and NIST16 datasets.

Table 4 presents the quantitative outcomes. The baseline denotes that we just use ResNet-50. It can be seen that without TDB, the AUC scores decrease by 2.0% on CASIA and 1.5% on NIST16, while without CEG, the AUC scores decrease by 5.9% on CASIA and 6.3% on NIST16. Furthermore, when RMPC is discarded, serious performance degradation in Table 4, i.e., 14.9% in terms of AUC and 22.2% in terms of F1 on CASIA can be observed.

In Figure 5, we show the different values of parameters $k$ in threshold-adaptive differentiable binarization to verify its effect over three datasets. With its increasing, the curve of binarization becomes steeper. Moreover, being smaller is inadequate for weakening message passing, while being larger will destroy the adaptive capacity of the network. It is obvious that the setting of 500 is the optimal solution.

4.5. Visualization Results

Qualitative results. As shown in Figure 4, We provide predicted forgery masks of various methods. Since the source code of ObjectFormer [46] is not available, their predictions are not available. The results demonstrate that our method could not only locate the tampering regions more accurately but also develop sharp boundaries. It benefits from the explicit modeling of inconsistencies and the full exploitation of edges by our method.

Visualization of message passing controller. To verify the usefulness of the region message passing controller (RMPC), we show the change of features before and after the controller in Figure 6. It is clear that RMPC facilitates the learning of forgery features and prevents false alarms. Specifically, the network without RMPC will make false judgments about objects that are similar to the forgery.

Visualization of edge reconstruction. To verify the effect of the edge reconstruction (ER) module, the change of features before and after EG is shown in Figure 7. The results demonstrate that the EG can effectively acquire accurate edges, thus helping our model to perform well.

5. Conclusion

In this paper, we propose a novel Image Forgery Localization framework with a two-step Edge-aware Regional Message Passing Controlling strategy. In detail, the first step is to account for fully exploiting the edge information. In the second step, guided by the learnable edges, an edge-aware dynamic graph is devised to weaken the message passing between the forged and authentic regions. Our paper provides a new research strategy to solve the misjudgment problem in the field of IFL. Extensive experimental results on several benchmarks demonstrate the effectiveness of the proposed algorithm.
References


[32] Xun Lin, Shuai Deng, Ying Fu, Xiao Bai, Xincui Chen, Xiaolei Qu, and Wenzhong Tang. Image manipulation detection by multiple tampering traces and edge artifact enhancement. Pattern Recognition, 133:109026, 2023. 1


