Effective Image Tampering Localization via Semantic Segmentation Network

Haochen Zhu, Gang Cao, Member, IEEE, and Mo Zhao

Abstract—With the widespread use of powerful image editing tools, image tampering becomes easy and realistic. Existing image forensic methods still face challenges of low accuracy and robustness. Note that the tampered regions are typically semantic objects, in this letter we propose an effective image tampering localization scheme based on deep semantic segmentation network. ConvNeXt network is used as an encoder to learn better feature representation. The multi-scale features are then fused by Upernet decoder for achieving better locating capability. Combined loss and effective data augmentation are adopted to ensure effective model training. Extensive experimental results confirm that localization performance of our proposed scheme outperforms other state-of-the-art ones.

Index Terms—Image forensics, Tampering localization, Encoder and decoder, ConvNeXt, Semantic segmentation.

I. INTRODUCTION

With the development of image processing techniques, forgery images become rather realistic and difficult to be distinguished from real ones. Malicious manipulations, such as splicing, object copy-move and removal, typically change the semantic expression of images. Misuse of the tampered images would cause great harm to public security and judicial forensics. It is significant to develop forensic techniques for identifying forgery images and locating the tampered regions therein. The image tampering localization techniques can help us discover more refined forensic information, such as the answers of how and where the malicious alteration occurs.

In recent years, a number of image tampering localization schemes have been proposed by leveraging on deep learning architecture. Region- or pixel-level tampering localization has been achieved to some extent. Huh et al. [1] use image EXIF metadata as a monitoring signal to learn self-consistency by convolution neural network (CNN). The algorithms Noiseprint [2] and ForSim [3] work by predicting whether two image patches come from the same source by measuring the distance of learned features. Such forensic schemes rely on checking consistency of specific clues, and might be greatly affected by postprocessing. Some other schemes treat the image tampering localization as a pixel classification task [4-11], which is solved by leveraging on an end-to-end deep learning framework. The specific clue features and their regional inconsistency are no longer pursued. The tampering localization network is trained by massive tampered image samples with pixel-level tampering annotations. The deep learning architectures developed in prior works include deep CNN [4-7], two stream region-CNN [8], long short-term memory (LSTM) network [9, 10] and dense fully convolutional network (DFCN) [11], etc. Besides, spatial regional and boundary features extracted by DeepLab network [12] and reinforced semantic information [13] are also applied. However, accuracy and robustness of such tampering localization algorithms are still insufficient for real applications.

In order to attenuate the deficiency of existing works, here we propose an effective image tampering localization scheme based on semantic segmentation encoder-decoder framework. Note that tampered regions in real-world images are typically related with some complete and distinct semantic objects. To capture such inherence of latent tampered regions, the precise pixel-level localization capability of deep semantic segmentation network is transferred and exploited. In the encoder, rich and hierarchical tampering features are extracted by ConvNeXt backbone [14], which is one of the state-of-the-art representation learning networks. The decoder built on Upernet [15] fuses the features at multiple levels and estimates the tampering probability for each pixel. Combined loss is adopted to guide effective model training, and data augmentation is used to enhance the generality and robustness. Performance advantages of the proposed tampering localization scheme have been verified extensively.

The rest of this letter is organized as follows. The proposed image tampering localization scheme is described in Section II, followed by experimental results and discussions in Section III. We draw the conclusions in Section IV.

II. PROPOSED SCHEME

In this section, the proposed image tampering localization scheme is presented in detail. The encoder-decoder localization network based on deep CNNs is first introduced and discussed. Then the applied data augmentation strategy and loss function are described. An overview of the scheme is shown in Fig. 1.
A. Encoder-Decoder Localization Network

Let an input color image be denoted by \( I \in \mathbb{R}^{H \times W \times 3} \), where \( H, W \) are the image height and width, respectively. \( I \) is first input to a convolutional layer of \( 4 \times 4 \) kernel with stride 4 for yielding an appropriate feature embedding \( X_0 \in \mathbb{R}^{H/4 \times W/4 \times C} \), where \( C = 128 \) is the number of feature map channels. The backbone encoder network owns four modules stacked with 3, 3, 27, 3 ConvNeXt blocks [14], respectively. Output of the \( i \)-th module \( E_i \) can be formulated as

\[
X_i = E_i(X_{i-1})
\]

where \( X_i \in \mathbb{R}^{H/2^i \times W/2^i \times 2^i \times C} \), \( i = 1, 2, 3, 4 \). Three downsampling blocks with rate \( 1/2 \) are integrated into \( E_3 \sim E_4 \), respectively.

As shown in the bottom left of Fig. 1, a ConvNeXt block consists of 7×7 depthwise and 1×1 general convolutions. The former mainly mixes spatial domain information, and the latter extends and condenses the feature maps in channel dimension. The large convolutional kernel could provide larger perceptual field for capturing large-scale and attentive tampering traces. Such complex large-kernel convolution operations are enforced with fewer channels for reducing model parameters. Efficient \( 1 \times 1 \) convolutional layers are used to deepen the channels. As such, both global and local tampering trace features could be extracted cost-effectively by ConvNeXt encoder.

According to the hierarchical nature of the encoder network, low and high modules tend to capture local and large-scale complex patterns, respectively. The combined use of such low-level pixelwise and high-level semantic features would benefit to tampered region identification. To this end, the Upernet [15] based decoder is designed to reconstruct a pixel-level tampering localization map. The multi-scale features \( \{X_i\} \) are adaptively fused via feature pyramid network (FPN) [16], which output a set of feature maps \( Y_i \), \( i = 4, 3, 2, 1 \), with the same dimension as \( X_i \). Specifically, \( X_i \) is fed to a pyramid pooling module [17] and converts into \( Y_i \). As illustrated in the bottom middle of Fig. 1, the high-level feature map \( Y_{i+1} \) is upsampled by two times and then concatenated into \( X_i \) by the \( i \)-th lateral connection \( L_i \). The corresponding generated feature map \( Y_i \) is

\[
Y_i = L_i(Y_{i+1}, X_i)
\]

where \( i = 3, 2, 1 \). The upsampling and \( 1 \times 1 \) convolution operations are used to align the two input arguments \( Y_{i+1}, X_i \). A \( 3 \times 3 \) convolution is followed to eliminate aliasing effect.

Lastly, all scales of decoded feature maps \( Y_i, i = 4, 3, 2, 1 \), are fused by performing bilinear upsampling alignment to \( Y_1 \) and concatenation. A \( 3 \times 3 \) convolutional layer is subsequently enforced to further interface such fused feature and reduce its channel dimension. Finally, the tampering localization map \( I' \) is yielded by predicting each pixel of \( I \) as tampered or pristine.

B. Data Augmentation

Sufficient sample images are rather important for training the proposed tampering localization network. However, there lack enough annotated real-world forgery images. The recent works [6, 9] have proposed to automatically synthesize plenty of forgery images based on object segment of COCO images [18]. In consideration of cost-effective computation, the object regions are randomly composited into pristine images. In order to simulate real-world post-processing, we propose to apply data augmentation operations to such composite sample images. Inspired by computer vision community, resizing, random cropping and flipping are applied as geometric transformation. Then Gaussian noising, blurring, and photometric distortions of brightness, contrast, saturation, hue are simulated to retouch the image appearance. Lastly, middle or high level post JPEG compression is enforced. Such augmentations are expected to make the training samples approach the distribution of real-world forgeries. As a result, generalization ability and robustness of the forensic scheme could be enhanced.

C. Loss Function

Since malicious tampering typically occurs locally within an image, the tampered region is usually much smaller than the pristine one. Under such an imbalanced distribution, the standard cross-entropy (CE) loss used for tampered/pristine pixel classification concentrates on most negative samples. It may lead to low true positive rate due to misclassification of tam-
pered pixels. In order to attenuate such class imbalance problem, we exploit the focal loss [19] defined as Eq. (3), which can be considered as a general CE loss.

\[ L_{\text{Focal}}(y_i, p_i) = -\sum (1-p_i)^{\gamma} y_i \log(p_i) - \sum (1-\alpha)p_i^{\gamma}(1-y_i)\log(1-p_i) \]  

(3)

where \( y_i \) and \( p_i \) are the ground truth and prediction labels for each pixel of the sample image, respectively. The hyperparameters \( \alpha, \gamma \) are the weights for making the model focus on learning from more difficult samples, and control the gradient of different imbalanced samples to a certain extent.

Intersection of Union (IOU) is another popular evaluation metric in tampering localization tasks. In order to reinforce the focus loss, we apply an optimized IOU measure, i.e., Lovasz-softmax loss [20], as the second loss function term. That is,

\[ L_{\text{lovasz}}(y_i, p_i) = \overline{\Delta J}(\max(1-p_i y_i, 0)) \]  

(4)

where \( \overline{\Delta J} \) denotes the Lovasz extension to traditional IOU loss [20]. Overall, the final loss function set is as

\[ L(y_i, p_i) = \lambda_1 L_{\text{Focal}}(y_i, p_i) + \lambda_2 L_{\text{lovasz}}(y_i, p_i) \]  

(5)

where \( \lambda_1, \lambda_2 \) are the weights. The focal loss addresses the pixel class imbalance and difficult samples. The lovazs-softmax loss guides the network trained to locate tampered region accurately.

III. EXPERIMENTS

In this section, extensive experiments are performed to test the performance of our proposed tampering localization scheme.

A. Experimental Setup

Datasets and Metrics. The training and validation sample images are collected from a public synthetic image dataset [9] and CASIAv2 [21]. The 55848 synthetic images [9] are created by copying and pasting arbitrary COCO object regions [18] into pristine Dresden [22] and MFC18 [23] images. The CASIAv2 dataset owns 5123 realistic forgery images. The total 60971 synthetic images with tampered pixel location maps are randomly divided into training and validation sets by 9:1. The following data augmentation operations are enforced subsequently to each sample image: resizing with a factor in [0.5, 2], cropping to 512×512 pixels, random horizontal flipping, Gaussian noise and blurring, random adjustment of brightness, contrast, saturation and hue, JPEG compression with Q=71-95. Since sample images have different resolutions, the resizing and cropping are always performed. Each of the other operations is applied with the probability 0.5. To keep consistent with the same of-the-art works [1, 2-4], four public image tampering datasets including Columbia [24], CASIAv1 [21], DSO [25] and NIST [26] are used as testing sample sets. The AUC, IOU and F1 metrics [11] are employed to measure the accuracy of tampering localization by comparing the predicted class with corresponding ground-truth for each pixel in a test image.

Implementation Details. The proposed tampering localization network is implemented based on MMSEgmentation framework [27]. All experiments run on a PC with one NVIDIA TitanXP GPU. Model parameters for the backbone ConvNeXt-

| Encoder | ResNet | ConvNeXt |
|---------|--------|---------|
| Decoder |        |         |
| Loss function | Cross entropy | Combined loss |
| Augmentation |        |         |
| AUC     | 73.1   | 84.6    | 83.1   | 81.9   | 88.5  | 79.7   | 85.3  |
| F1      | 31.8   | 54.8    | 53.1   | 54.5   | 53.4  | 51.5   | 58.1  |
| IOU     | 29.1   | 50.7    | 50.0   | 51.5   | 50.0  | 48.7   | 54.8  |

B are initialized by the weights pretrained on ImageNet-22K dataset [28], AdamW optimizer is adopted. The initial learning rate is set as 1×10^-4 using linear warmup schedule with 1500 iterations and reduced by a poly policy. The maximum iteration is 160K and batch size is 4.

B. Ablation Studies

Performance via encoder-decoder configurations. As for the encoder, ResNet-101 [29] which has a similar architecture to ConvNeXt is compared. In Table I, the test results on CASIAv1 dataset show that ConvNeXt outperforms ResNet-101 by 26.3% and 25.7% in F1 and IOU, respectively. In the decoder, fusing the features from all ConvNeXt blocks, i.e., \( X_1, X_2, X_3, X_4 \), behaves the best. The performance degrades when merely part of or none features are fused. Such results validate effectiveness of the proposed encoder-decoder network.

Selection of loss function. The influence of loss function on the tampering localization performance is tested. The results in Table I show that the performance of combined loss is higher than that of CE loss. It verifies the advantage of combined loss for tampering localization.

Performance via data augmentation. The scheme without applying our proposed data augmentation is also tested. Its F1 and IOU values are 51.5%, 48.7%, which are lower than those of the scheme with augmentation by 6.6%, 6.1%, respectively.

C. Comparison with Other Localization Algorithms

Performance of the proposed tampering localization scheme is compared with other state-of-the-art ones, i.e., ManTraNet [4], Noiseprint [2], ForSim [3], DFCN [11] and OSN [6]. The results in Table II indicate that our proposed scheme achieves the highest localization accuracy on Columbia, CASIAv1 and NIST datasets. The performance measurements of compared algorithms on public datasets are referred from the literature [6]. On the average, F1 and IOU achieved by the proposed scheme are respectively 52.7% and 48.8%, which are higher than the second best one, i.e., OSN, by 3.1% and 7.9%. Such two algorithms behave rather better than the other ones. It should be mentioned that our scheme meets challenge on DSO dataset, which consists of close-up human face or body spliced images. The unromantic performance might attribute to the mismatching between distributions of training and testing samples, which
TABLE II

PERFORMANCE COMPARISON OF DIFFERENT TAMPERING LOCALIZATION ALGORITHMS ON DIFFERENT DATASETS. METRIC VALUES ARE IN PERCENTAGE, AND THOSE OF COMPARED ALGORITHMS ARE CITED FROM [6]. "-" INDICATES NOT APPLICABLE.

| Algorithm   | Columbia AUC | Columbia F1 | CASIAv1 AUC | CASIAv1 F1 | NIST AUC | NIST F1 | DSO AUC | DSO F1 | Average AUC | Average F1 | Average IOU |
|-------------|--------------|-------------|-------------|-------------|----------|---------|---------|---------|-------------|-------------|-------------|
| ManTraNet [4] | 74.7         | 35.7        | 77.6        | 13.0        | 63.4     | 8.8     | 74.5    | 8.5     | 73.8        | 23.0        | 16.3        |
| Noiseprint [2] | 84.0         | 36.2        | 64.0        | 16.9        | 64.4     | 18.8    | 79.6    | 48.7    | 73.8        | 36.2        | 26.8        |
| ForSim [3]    | 73.1         | 47.4        | 55.4        | 16.9        | 64.4     | 18.8    | 79.6    | 48.7    | 73.8        | 36.2        | 26.8        |
| DFCN [11]     | 78.9         | 39.5        | 65.4        | 19.2        | 77.8     | 25.0    | 72.4    | 30.3    | 73.6        | 32.2        | 23.6        |
| OSN [6]       | 86.2         | 70.0        | 50.9        | 25.8        | 78.3     | 32.2    | 85.4    | 43.6    | 84.3        | 49.6        | 40.9        |
| Proposed      | 93.6         | 88.5        | 85.3        | 58.1        | 74.3     | 37.0    | 80.1    | 27.0    | 83.3        | 52.7        | 48.8        |

TABLE III

ROBUSTNESS COMPARISON OF DIFFERENT TAMPERING LOCALIZATION ALGORITHMS AGAINST POSTPROCESSING OF DIFFERENT ONLINE SOCIAL NETWORKS (OSNS). METRIC VALUES ARE IN PERCENTAGE, AND THOSE OF COMPARED ALGORITHMS ARE CITED FROM [6]. "-" INDICATES NOT APPLICABLE.

| Algorithm   | OSNs  | Columbia AUC | Columbia F1 | CASIAv1 AUC | CASIAv1 F1 | NIST AUC | NIST F1 | DSO AUC | DSO F1 | Average AUC | Average F1 | Average IOU |
|-------------|-------|--------------|-------------|-------------|-------------|----------|---------|---------|---------|-------------|-------------|-------------|
| ManTraNet [4] | Facebook | 62.6        | 10.3        | 76.3        | 10.2        | 65.2     | 9.5     | 63.8    | 10.9    | 73.8        | 23.0        | 16.3        |
| Noiseprint [2] |            | 66.4        | 17.5        | 57.5        | 13.6        | 58.0     | 5.4     | 30.3    | 6.1    | 64.3        | 11.3        | 6.9        |
| ForSim [3]    | Weibo   | 68.3        | 71.4        | 65.4        | 19.0        | 70.5     | 20.7    | 67.3    | 23.8    | 83.1        | 27.6        | 18.0        |
| DFCN [11]     | Wechat  | 92.9        | 87.3        | 85.3        | 53.4        | 73.8     | 36.2    | 80.1    | 26.0    | 82.6        | 50.7        | 46.6        |
| OSN [6]       |         | 62.0        | 10.3        | 75.4        | 9.9         | 67.1     | 8.8     | 60.6    | 5.7     | 63.8        | 3.6         | 66.7        |
| Proposed      |         | 66.4        | 17.5        | 57.5        | 13.6        | 58.0     | 5.4     | 30.3    | 6.1     | 64.3        | 11.3        | 6.9        |
| ManTraNet [4] | Weibro  | 61.0        | 43.3        | 54.2        | 16.5        | 58.1     | 15.0    | 56.8    | 26.0    | 57.5        | 25.7        | 16.8        |
| Noiseprint [2] |            | 67.6        | 45.8        | 65.3        | 19.1        | 70.6     | 19.2    | 63.9    | 22.7    | 66.8        | 26.7        | 17.5        |
| ForSim [3]    | Wechat  | 88.3        | 72.4        | 65.4        | 20.2        | 78.0     | 29.4    | 80.8    | 37.0    | 83.2        | 46.3        | 38.0        |
| OSN [6]       |         | 91.4        | 88.2        | 83.1        | 52.8        | 73.5     | 35.7    | 79.7    | 26.2    | 82.4        | 50.8        | 46.9        |
| Proposed      |         | 61.3        | 19.9        | 12.5        | 8.0         | 65.4     | 9.5     | 58.2    | 7.6     | 64.3        | 11.3        | 6.9        |
| Noiseprint [2] |            | 63.9        | 20.2        | 12.4        | -           | -        | -       | -       | -       | -           | -           | -           |
| ForSim [3]    |         | 65.0        | 49.6        | 35.4        | 15.3        | 58.1     | 13.6    | 56.4    | 24.7    | 58.2        | 25.8        | 16.8        |
| DFCN [11]     |         | 67.6        | 48.7        | 65.1        | 19.3        | 70.1     | 17.6    | 65.3    | 22.1    | 67.0        | 26.9        | 17.9        |
| OSN [6]       |         | 88.3        | 72.7        | 63.1        | 40.5        | 78.4     | 28.6    | 82.3    | 36.6    | 82.6        | 44.6        | 36.4        |
| Proposed      |         | 93.3        | 88.2        | 85.4        | 35.8        | 73.8     | 34.9    | 79.5    | 23.0    | 80.5        | 45.5        | 41.6        |

Fig. 2. Tampering localization maps on six example forgery images. From left to right: test images, ground-truth, and results of different localization methods.

As qualitative evaluation, tampering localization results on six example forgery images are illustrated in Fig. 2. We can see that most of tampered regions could be identified accurately by our method, which outperforms the other counterparts evidently in terms of completeness and false alarm errors of the identified tampered regions.

D. Robustness Analysis

As mentioned in [6, 7], various lossy operations adopted by online social networks pose a great challenge for robust image forgery forensics. We evaluate the robustness of different localization algorithms against the postprocessing incurred by online social network platforms, such as Facebook, Weibo and Wechat. Table III verifies that the comparative performance advantage of our scheme could still be kept as that without postprocessing. For instance, the average F1 of ours (50.7%, 50.8%, 45.5%) are always above those of the OSN algorithm [6] (48.8%, 46.3%, 44.6%) on the three platforms, respectively. Furthermore, the overall performance reduction of our scheme is approximately the smallest in different cases. That could be illustratedly confirmed by the results on DSO dataset, where F1 of ours always keeps above 23% while that of other methods fluctuates visibly. Although AUC values of our scheme are slightly lower than those of OSN, the overall outperformance among the state of the art algorithms is still evident. It should be noted that AUC might not be a reasonable metric for tampering localization task due to its severe nonlinearity and neglect of class imbalance.

IV. CONCLUSION

In this work, we have proposed a novel image tampering localization scheme based on semantic segmentation encoder-decoder network. Hierarchical tampering features are extracted by the ConvNeXt encoder. Such features are fused properly by FPN for accurately estimating the tampering probability of each pixel. Extensive evaluations have verified the effectiveness and robustness of our proposed scheme. The localization performance gains achieved by the applied data augmentation and combined loss have also been validated.
