Noise and Edge Based Dual Branch Image Manipulation Detection

Zhongyuan, ZZ, Zhang∗
School of Computer Science and Technology, Xi’an Jiaotong University, Xi’an, China
zhangzhongyuan@stu.xjtu.edu.cn

Yi, YQ, Qian
School of Computer Science and Technology, Xi’an Jiaotong University, Xi’an, China
yqian@mail.xjtu.edu.cn

Yanxiang, YZ, Zhao
School of Computer Science and Technology, Xi’an Jiaotong University, Xi’an, China
yanxiangzhao@stu.xjtu.edu.cn

Lin, LZ, Zhu
China Mobile Research Institute, China Mobile, Beijing, China
zhulinyj@chinamobile.com

Jinjin, JW, Wang
China Mobile Research Institute, China Mobile, Beijing, China
wangjinjin@chinamobile.com

Juan, JZ, Zhao
China Mobile Research Institute, China Mobile, Beijing, China
zhaojuan@chinamobile.com

ABSTRACT
Unlike ordinary computer vision tasks that focus more on the semantic content of images, the image manipulation detection task pays more attention to the subtle information of image manipulation. In this paper, the noise image extracted by the improved constrained convolution is used as the input of the model instead of the original image to obtain more subtle traces of manipulation. Meanwhile, the dual branch network, consisting of a high-resolution branch and a context branch, is used to capture the traces of artifacts. In general, most manipulation leaves artifacts on the manipulation region boundary. A specially designed manipulation edge detection module is constructed based on the dual branch network to identify these artifacts. We add a distance factor to the self-attention module to better describe the correlation between pixels. Experimental results on publicly available image manipulation datasets demonstrate the effectiveness of our model.

CCS CONCEPTS
• Computing methodologies; • Computer vision.

KEYWORDS
Image forensics, Image manipulation detection, Image noise extraction, Edge detection

1 INTRODUCTION
People can easily use image editing software to obtain a manipulated image. It is often difficult for human eyes to discern traces of manipulation. While image editing technology brings convenience, it also brings some problems. So, a general image manipulation detection method is needed.

Image manipulation detection pays more attention to the details left by manipulation operations. Recent studies on image manipulation detection are mainly based on convolutional neural networks (CNN). The CNN architecture tends to learn features representing the semantic content of images, which is not entirely consistent with the purpose of the image manipulation detection task. To solve this problem, the image is processed by the constrained convolution [1] to obtain the corresponding noise image. The noise image is semantically independent. Nevertheless, due to the constrained process, the weights of constrained convolution are prone to drastic changes in practical training. So, we propose an improved constrained convolution to obtain the corresponding noise image.

To capture more subtle manipulation information, the features in the CNN need to maintain high resolution. Due to the local characteristics of CNN, a certain degree of downsampling is indispensable to obtain sufficient image context information. We build a dual branch network to obtain both kinds of information. One of the branches is used to obtain the context information of the image, and the other branch maintains a certain resolution to avoid losing too many details.
Figure 1: The overall structure of the NEDB-Net. The model consists of a high-resolution branch and a context branch. The noise image is used as the manipulation detection cue instead of the original image.

The manipulation edge is critical information for the manipulation detection task since the three fundamental manipulation operations of copy-move, splicing, and removal will leave artifacts on the manipulation region border. In the previous works [2–4], edge prediction is added to the detection model to enhance detection performance. These methods cannot fully utilize the features generated by different layers. We propose an edge fusion module, which can adaptively fuse the information contained in shallow features and deep features.

Global correlation information is necessary for manipulation detection. The non-local module [5] is used in CNN to obtain the correlations between each pixel and all other pixels. However, the non-local module can break the distance limit to calculate the correlation between pixels but also ignore their distance relationship simultaneously. Therefore, we add a distance metric to the non-local module so that it can better describe the global correlations among pixels.

Overall, the main contributions of this work are as follows:

- We optimize the constrained convolution process to make constrained convolution more stable and easier to train;
- We design a dual branch network and the corresponding manipulation edge detection module;
- We add the distance factor to the non-local module so that the non-local module can better capture the global correlations among pixels;
- The results of on publicly datasets demonstrate that our model has advantages over the SOTA.

2 PROPOSED METHOD

As shown in Figure 1, we propose a model named NEDB-Net. The improved constrained convolution processes the input image to obtain the corresponding noise image. The noise image is then fed into a dual branch network with ResNet-34 [6] as the backbone. The high-resolution branch can maintain the resolution of the feature to obtain more details, and the context branch is used to obtain richer correlations among pixels. The edge extraction block (EEB) extracts corresponding manipulation edge information from features output from each layer in the model. And then all the edge information is fused by the edge merge block (EMB) to get the final prediction. Features of the context branch are optimized by the attention modules and fused with the feature of the high-resolution branch. Then the fused feature is upsampled and convolved to obtain the final mask of manipulation regions.

2.1 Improved Constrained Convolution

Ordinary CNN tends to learn features representing semantic information of an image rather than manipulation details. Some works use constrained convolution to get the noise image which is independent of semantic content. Constrained convolution imposes constraints after kernels update weights, making the weights distribution of kernels similar to the high-pass filter. For a convolution kernel, the constraints can be achieved by the following steps:

1. Calculate the sum of the weights of the non-central positions;
2. Divide the weights of the non-center positions by the sum in Step 1;
3. Set the center position weight to -1.

Since the weights before constraints may be positive or negative, their sum may be negative, and the absolute value of the sum may be relatively small. At this time, according to the constraint rules, dividing the weights of all non-central positions by this sum will cause three problems:

1. Dividing the weight by this relatively small sum causes the weight to be amplified a lot, but the center position weight is still -1;
2. If the sum is negative, the division operation will make the positive weights negative and the negative weights positive;
3. There is an order of magnitude difference between the weights of the constrained convolution and the weights of the subsequent normal convolution.

To address the above issues, we set new constraints on convolution. The improved constrained convolution training process is shown in Algorithm 1. Firstly, we use a fixed method to initialize the convolution kernel to avoid the impact of random initialization. The weight of the non-center position is related to its Euclidean distance from the center position.

Secondly, we calculate the sum of the absolute values of weights $S_k$ in Line 7 of Algorithm 1. This can not only ensure the accumulation of weights, but also avoid that the sum of weights is a relatively small negative value. The divisor of the division in Line 8 is the $S_k$. Thirdly, in Line 9, after the division operation, small positive values and all negative values are set to 0.001, making their impact small but not useless. Finally, in Line 10, the weight of the center position is not strictly equal to -1 but is equal to the negative sum of the weights of all non-center positions. With these constraints, the convolution can better extract the image noise and slow down fluctuations during training.

### Algorithm 1 Training Algorithm for Improved Constrained Convolutional Layer

initialize $S_k$'s with the Laplacian-like weight

| $i = 1$ |
| --- |
| while $i < max\_\text{iter}$ do do feedforward pass update filter weights through stochastic gradient descent and backpropagate errors set $w_k(c, c) = 0$ for all K filters calculate the sum of the absolute values of the non-central position weights for all K filters, $S_k = \sum_{m, n \neq c} |w_k(m, n)|$ let $w_k(m, n) = \frac{w_k(m, n)}{S_k}$ to regularize $w_k$ set $w_k(m, n) = 0.001$ if $w_k(m, n) \leq 0.001$ set $w_k(c, c) = -S_k$ for all K filters $i = i + 1$ end

2.2 Dual Branch Network and Edge Prediction Module

We design a dual branch network with ResNet-34 as the backbone to obtain more subtle information. As shown in Figure 1, the high-resolution branch maintains the feature’s height and width at 1/8 of the input image for more detailed information. The context branch changes the feature’s height and width to 1/32 of the input image through multiple convolutions to obtain richer contextual information. Finally, the features of the two branches are fused to obtain information with sufficient context and details.

Since each layer in the CNN learns different contents, we utilize the edge extraction block (EEB) to extract the edge information from the features output by each layer of the network. The flow of EEB is shown in Figure 2. To reduce the calculation and fully use the feature information, we first use $1 \times 1$ convolution to reduce the number of features’ channels and then construct residual learning.

Finally, $1 \times 1$ convolution is used to reduce the number of channels to 32.

Different edges extracted by EEB contain different information. The edge of shallow feature contains rich detail information, while the edge of deep feature has more overall information. As shown in Figure 3, we refer to [7] to design an edge fusion module (EMB) to adaptively fuse different edges. For the two edge information $edge_1$ and $edge_2$, directly use element-wise addition to obtain the fused information and then use global average pooling to compress the width and height of the fused information into $1 \times 1$:

$$Fuse_{avg} = \text{AvgPooling}(edge_1 + edge_2)$$ (1)

Where $Fuse_{avg}$ represents the pooled fused information. In order to further learn the pooled information and reduce the amount of calculation, we use $1 \times 1$ convolution to reduce the channel dimension of $Fuse_{avg}$ to 1/2 of the original:

$$C = \text{Conv}_{1 \times 1}(Fuse_{avg})$$ (2)

Where $C$ represents compressed information. Because there are two edge information to be fused, we use two $1 \times 1$ convolutions to learn the weight of each edge in future fusions from the compressed information and restore the number of channels:

$$\begin{cases} P_1 = \text{Conv}_{1 \times 1}(C) \\ P_2 = \text{Conv}_{1 \times 1}(C) \end{cases}$$ (3)

Where $P_1$ and $P_2$ represent the learned fusion weights. To make the sum of two weights be 1, Softmax is computed over them. Then the two weights are multiplied by the corresponding edge information to indicate that the corresponding information is selected from the original edge. Finally, the selected edge information is added to obtain the fused edge information:

$$edge = \frac{e^{P_1}}{e^{P_1} + e^{P_2}} \times edge_1 + \frac{e^{P_2}}{e^{P_1} + e^{P_2}} \times edge_2$$ (4)

Only the fusion of two edges is illustrated here. If there are multiple edges to be fused, the process is the same as above.

2.3 Self-attention Mechanism with the Distance Factor

The global information is essential for image manipulation detection. Therefore, the non-local module is applied to CV tasks, which can break through the distance limitation of convolution and obtain the correlations between each pixel and other pixels.

However, ignoring the channel dimension, the image is 2-dimensional and contains much richer spatial information than 1-dimensional natural language. The influence of one pixel on other
pixels in an image is closely related to distance. The closer two pixels are in the distance, the stronger their correlation is (pixels of the same class are positively correlated, and pixels of different classes are negatively correlated).

The non-local module calculates the correlations among pixels across distances while ignoring the effect of distance factors on the correlation. To reflect the distance relationship between pixels, we use Euclidean distance to construct the distance matrix of each pixel. The structure of the non-local module with the distance factor is shown in Figure 4.

3 EXPERIMENT

3.1 Experimental Setup

Implementation Details. We implement our NEDB-Net using the Pytorch framework. The high-resolution branch and the context branch of the model are initialized with the weights of ResNet-34 pre-trained on ImageNet [8]. The width and height of the input image are uniformly adjusted to 512. Image pixel values are divided by 255 for normalization. Then the input is standardized by subtracting the mean and dividing by the variance. The mean values of the three channels of BGR are 0.406, 0.456, and 0.485. The variances are 0.225, 0.224, and 0.229. Flipping and mirroring are used to perform simple data augment. The training batch size is set to 48. We train the model for 12K steps. The learning rate is initially set to 0.01 and then reduced to 0.0075, 0.005, and 0.0025 after step 5K, 7.5K, and 10K, respectively. All experiments are performed on a single NVIDIA Tesla V100 GPU with 32GB memory.

Evaluation Criteria. We evaluate the prediction results using pixel-level precision, recall, and F1. Because in the actual manipulation image, the most appropriate threshold cannot be predicted, we use the median value of 0.5 as the threshold to determine the positive and negative classes.

3.2 Comparison with the State of the art

Datasets. We select CASIAv1 [9], CASIAv2 [10], COVERAGE [11], COLUMBIA [12], NIST16 [13] these publicly available image manipulation datasets as the experimental datasets.

The composition of the manipulation image types of datasets is shown in Table 1. It is worth noting that some of the manipulation images lack the corresponding ground-truth or do not match the shape of the ground-truth. We only count the correct manipulation images here.

Evaluation method. When comparing effects in many studies, the model will be trained on other large datasets and then tested on the above datasets. Another comparison method is to select a portion of each dataset for fine-tuning and the rest portion for testing. We do not think these comparison methods are particularly plausible because they do not adequately demonstrate the model’s generality against unknown data. So, we adopt the same evaluation method as [4], let the model train only on the CASIAv2 datasets, and then directly test it on the remaining dataset. This approach directly reflects whether the model has learned how to detect manipulation rather than just fitting the dataset.

Baselines. The baseline models we chose are as follows: Fully Convolutional Networks (FCN) [14], High-Pass Fully Convolutional Network (HP-FCN) [15], Manipulation Tracing Network (ManTra-Net) [17], Constrained R-CNN (CR-CNN) [18], Generate, Segment, Refine Network (GSR-Net) [3], Multi-View Multi-Scale Supervision Network (MVSS-Net) [4], and Dense Fully Convolutional Network (D-FCN) [16].

Among them, HP-FCN and ManTra-Net directly use the model weights provided by the authors due to the lack of training codes and private datasets. For D-FCN, the authors provide the weight trained on the private manipulation dataset and 10% of the NIST16 dataset. We train it on the CASIAv2 dataset based on this pretrained weight and report two experimental results, where D-FCN (pre) means directly using the author’s pre-trained weight for testing, and D-FCN (re) means using our retrained model for testing. Other methods either follow the same evaluation protocol or retrain on the CASIAv2 dataset. The experimental results are shown in Table 2 (part of the experimental results are obtained from [4]).

Manipulation detection effect comparison. As can be seen from Table 2, our NEDB-Net outperforms other models on other datasets, except that it is worse than D-FCN (pre-train) and MVSS-Net on the NIST16 dataset. Especially the results on COLUMBIA and CASIAv1 are much ahead of other models. The experimental results verify the detection effect of our model and show that our model can also achieve good results on unknown datasets. In order
Table 1: The composition of manipulation image datasets

| Datasets  | Copy-move | Splicing | Removal | Total |
|-----------|-----------|----------|---------|-------|
| CASIAv1   | 459       | 461      | 0       | 920   |
| CASIAv2   | 3263      | 1843     | 0       | 5106  |
| COVERAGE  | 100       | 0        | 0       | 100   |
| COLUMBIA  | 0         | 180      | 0       | 180   |
| NIST16    | 68        | 288      | 208     | 564   |

Table 2: The composition of manipulation image datasets

| CASIAv1 | COVERAGE | COLUMBIA | NIST16 |
|---------|----------|----------|--------|
| FCN     | 0.441    | 0.199    | 0.223  | 0.167  |
| HP-FCN  | 0.154    | 0.003    | 0.067  | 0.121  |
| Mantra-Net | 0.155 | 0.286    | 0.364  | 0.000  |
| CR-CNN  | 0.425    | 0.300    | 0.519  | 0.252  |
| GSR-Net | 0.387    | 0.285    | 0.613  | 0.283  |
| MVSS-Net | 0.452 | 0.453    | 0.658  | 0.292  |
| D-FCN(pre) | 0.007 | 0.194    | 0.376  | 0.402  |
| D-FCN(re) | 0.331    | 0.260    | 0.233  | 0.136  |
| NEDB-Net | 0.511    | 0.463    | 0.753  | 0.291  |

Figure 5: Some localization results for example images in COLUMBIA dataset (Row 1–3), CASIAv1 dataset (Row 4–6). The first and second columns are the manipulation images and ground-truths, respectively. The other columns are the prediction results obtained by different methods.

to compare the effects of each model more intuitively, we selected three manipulation images from COLUMBIA dataset and CASIAv1 dataset respectively. The detection results of each model are shown in Figure 5. From this figure, it is observed that the results output by our model are more consistent with the ground-truths. For example, our model has higher confidence in the prediction of manipulation regions. Moreover, our model is more accurate for the edge detection of the manipulation region.

Robustness evaluation. To compare the robustness of the model to compression and blur, we apply standard OpenCV built-in functions `imencode` (JPEG compression) and `GaussianBlur` (JPEG GaussianBlur) on CASIAv1, respectively. The experimental
results are shown in Figure 6. As can be seen from this figure, our NEDB-Net does not perform very well with a small Gaussian blur kernel. But as the Gaussian kernel increases, the anti-blurring ability of our model continues to increase. Except for Mantra-Net, other models are relatively weak against large blur kernels. In addition, GSR-Net performs relatively well for the interference of compressed images. The ability of our model to fight compression is similar to other models.

4 CONCLUSION

In this paper, we propose an image manipulation detection model based on image noise and manipulation edge. In our NEDB-Net, the improved constrained convolution can better extract the noise information of the image while addressing the problem of training stability. The non-local module ignores the distance relationship among pixels while computing the correlations among pixels spanning distances. We add the distance factor to it to better capture the global information among pixels. The dual branch network composed of the high-resolution branch and the context branch can fully preserve the detailed information of manipulation. Rich edge information extracted based on the dual branch network and the adaptive edge merge module greatly improve the overall manipulation detection effect. Experiments on public manipulation datasets demonstrate the state-of-the-art detection performance of our model.

ACKNOWLEDGMENTS

This work was supported by the MoE-CMCC "Artificial Intelligence" Project (No.MCM20190701). The authors would like to thank the Jiutian deep learning platform of China Mobile. This work used the platform for project construction and model training.

REFERENCES

[1] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," IEEE Transactions on Information Forensics and Security, vol. 13, no. 11, pp. 2691–2706, 2018.

[2] R. Salloum, Y. Ren, and C.-C. J. Kuo, "Image splicing localization using a multi-task fully convolutional network (mfcn)," Journal of Visual Communication and Image Representation, vol. 51, pp. 201–209, 2018.

[3] P. Zhou, B.-C. Chen, X. Han, M. Najibi, A. Shrivastava, S.-N. Lim, and L. Davis, "Generate, segment, and refine. Towards generic manipulation segmentation," in Proceedings of the AAAI conference on artificial intelligence, 2020, pp. 13 058–13 065.

[4] X. Chen, C. Dong, J. Ji, J. Cao, and X. Li, "Image manipulation detection by multi-view multi-scale supervision," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2021, pp. 14 185–14 193.

[5] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2018, pp. 7794–7803.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[7] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2019, pp. 510–519.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2009, pp. 248–255.

[9] J. Dong, W. Wang, and T. Tan, "Casia image tampering detection evaluation database, 2010, http://forensics.idealtest.org.

[10] J. Dong, W. Wang, and T. Tan, "Casia image tampering detection evaluation database," in IEEE China Summit and International Conference on Signal and Information Processing, 2013, pp. 422–426.

[11] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, "Coverage—a novel database for copy-move forgery detection," in IEEE international conference on image processing (ICIP), 2016, pp. 161–165.

[12] J. Hsu and S. Chang, "Columbia uncompressed image splicing detection evaluation dataset," 2006, https://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataset/AuthSplicedDataset.htm.

[13] H. Guan, M. Kozaik, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kherytkhah, J. Smith, and J. Fiscus, "Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation," in IEEE Winter Applications of Computer Vision Workshops (WACVW), 2019, pp. 63–72.

[14] J. Long, E. Shlensker, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440.

[15] H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 8301–8310.

[16] F. Zhuang, H. Li, S. Tan, B. Li, and J. Huang, "Image tampering localization using a dense fully convolutional network," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2986–2998.

[17] Y. Wu, W. AbdAlmageed, and P. Natarajan, "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 (CVPR), 2019, pp. 9543–9552.

[18] C. Yang, H. Li, F. Lin, B. Jiang, and H. Zhao, "Constrained r-cnn: A general image manipulation detection model," in 2020 IEEE International Conference on Multimedia and Expo (ICME), 2020, pp. 1–6.