An Intelligent Forensics Approach for Detecting Patch-Based Image Inpainting

Xinyi Wang,1 He Wang,2 and Shaozhang Niu1

1Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing, China
2State Grid Jiangsu Electric Power Co. Ltd., Marketing Service Center, Nanjing, China

Correspondence should be addressed to Shaozhang Niu; szniu@bupt.edu.cn

Received 17 September 2020; Revised 10 October 2020; Accepted 18 October 2020; Published 28 October 2020

1. Introduction

With the popularity of digital cameras and smartphones, digital images have become increasingly widely used. However, a large amount of powerful, user-friendly image editing and processing software is making it easier to edit and modify digital images. People can easily modify digital images without having to learn professional and complex techniques and even use computers to synthesize realistic digital images, resulting in a large number of forgeries and composite images propagating on the network. It is difficult for ordinary people to visually observe the traces of these forged image modifications. Once those forged photos are used by people with bad purposes, there is no doubt that it will seriously threaten the stability and development of society. Therefore, the digital image forensics technology [1–4] has become essential. Passive image forensics methods designed to detect tamper traces without using prior knowledge, especially, have generated much research interest because they do not need any auxiliary information, for example, watermarks or signatures [5–8].

Passive image forensics is mostly targeted at specific tampering methods, such as double JPEG compression [9–11] and median filtering [12, 13]. Among them, the object removing operation is one of the most concerned malicious tampering methods. Since the reduction of image objects can mask important targets, the content of the image is changed to a large extent and affects the viewer’s cognitive understanding. Currently, object removing operations are mainly implemented in two ways: copy-move [14, 15] and image inpainting methods [16–18].

This paper focuses on image tampering forensics based on inpainting techniques. Image inpainting is a significant study domain in computer vision and has attracted many researchers over the years [19–22]. Its main purpose is to use the information of the known area of the image to repair the damaged or removed area and to make the inpainted image keep the consistency on texture and structure as much as possible. We...
can obtain a realistic visual effect so that the observer is unable to detect that the image was once edited. It can also be utilized to eliminate image semantic objects for malicious purposes. The basic symbols of image inpainting can be seen in Figure 1. In this sketch map, $I$ is the original image and $\Lambda$ represents the undamaged part, where $\Omega$ indicates the area to be repaired. Many image inpainting approaches fill the damaged area $\Omega$ using the undamaged part $\Lambda$. The patch-based methods [23, 24] are the representative studies of image restoration and many methods have been improved on this basis. However, these methods are often used to remove objects to change the semantics of images. As shown in Figure 2, this is an example of an image tampering operation using the inpainting algorithm, in which Figure 2(a) is an original image and Figure 2(b) is an inpainted image.

At present, there is much conventional research on inpainting forensics algorithms, which have various limitations and deficiencies. As a first attempt, Wu et al. [25] were the first to propose an image detection algorithm based on sample synthesis restoration. In their paper, the idea of zero connectivity to select suspicious regions was presented. Moreover, the block ambiguity level was used to distinguish the repaired patches. Since their approach required prior selection of the areas and used a full search strategy to find suspicious blocks, the computational complexity was high and could not meet the needs of applications. On the basis of this, Bacchuswar et al. [26] presented a jump patch-block matching method. Although some simplifications can save a certain amount of time, it was still a half-automatic detection method. While Zhang et al. [27] presented a faster forensics method using central pixel mapping (CPM). However, the rate of misrecognition was still high. These are conventional methods of inpainting detection, which rely on approximate characteristics between image blocks to exploit the difference between inpainting and noninpainting areas. They all have a large time complexity to extract features and low robustness to postprocessing operations. To complete the forensic task of the repair area, we need to extract features from the images and then distinguish the pixels in the image into two categories: inpainting and noninpainting. However, the inpainting operation often leaves no obvious traces, so it is difficult to obtain features with high discrimination by conventional methods. To overcome these issues, we employ an improved network based on the Mask Regional Convolutional Neural Network (Mask R-CNN) [15] to detect the inpainting manipulation and identify the inpainting localization.

In this paper, our main contributions are as follows. First, the backbone of the Mask R-CNN is applied to detect and locate the manipulated areas under complex backgrounds successfully. Use the prior information at the pixel level of the inpainted area to guide and supervise the training of this deep neural network. Then, we adjust and improve the network according to the shape of the inpainted data, including the size of the anchor scales and an improved method using the threshold of nonmaximum suppression, so that the model can generate the more accurate area of interest. Finally, the deep neural network works well in our self-made dataset.

The rest of this paper is organized as follows. Section 2 reviews the background of the patch-based inpainting method. In Section 3, we describe the deep learning architecture for our inpainting forensics. Section 4 gives the extensive experimental results on the datasets. Finally, we conclude the paper in Section 5.

2. Patch-Based Inpainting

Criminisi et al. [28] proposed the inpainting approach, combined both structure and texture features, which do not divide the image, and simultaneously processed the image texture and structure. So far, it is the most popular exemplar-based image inpainting algorithm. This algorithm first searches for the pixel block that best matches the area to be repaired and then fills the damaged area with the obtained pixel block. This inpainting method can obtain better results. Many of the subsequent studies on these image inpainting techniques were improved under the framework of the Criminisi algorithm. Therefore, we use the Criminisi algorithm as a representative example to introduce the patch-based inpainting principle in detail.

Figure 3 shows the patch-based inpainting process of the Criminisi algorithm. First, the user specifies a target area that needs to be repaired or removed as shown in Figure 3(a). There has been an image with the damaged area $\Omega$ and the known area $\Phi$, and the aim of Criminisi’s inpainting is to restore the target area (damaged area $\Omega$) with the image information of the source area (known area $\Phi$). The boundary area is represented by $\partial \Omega$. The primary steps of the Criminisi algorithm for image restoration are as follows:

Step 1: compute the priorities of the points on the interior $\partial \Omega$ and find the point $p$ with the highest priority. Then, the image patch $\Psi p$ centered at the point $p$ is chosen as a target patch to be inpainted, which is shown in Figure 3(b).

Step 2: search the whole known region $\Phi$ for the reference block $\Psi q$ which is the most similar to $\Psi p$ as shown in Figure 3(c). Minimizing the perceived distance $d (\Psi p, \Psi q)$, which is used to measure the similarity between $\Psi p$ and $\Psi q$, it is defined as follows:

$$d (\Psi p, \Psi q) = \sum_{i} \sum_{j} |\Psi p(i, j) - \Psi q(i, j)|^2.$$  \hspace{1cm} (1)

Step 3: search for the corresponding pixels of $\Psi q$ to repair the damaged area in $\Psi p$, and keep the priority between $\Psi p$ and $\partial \Omega$ constantly refreshed, which can be seen in Figure 3(d).
Step 4: cycle from the first to the third step until the damaged area is completely inpainted.

The Criminisi algorithm and the improved algorithm based on it have some dissimilarities in the methods of calculating the priority and finding the closest block that can match, but using these methods to fill the damaged block introduces the abnormal similarity. They differ from texture-like blocks in nature [12] because they are not generated by normal imaging algorithms. This will cause an unusual distribution of pixel values. Moreover, the block-filled area based on the patch will be different from the area of natural imaging. This leaves traceable traces for image forensics. We can mine these features to detect patch-based inpainting tampering.

3. Methodology

The backbone network in this paper is based on the deep learning framework: Mask R-CNN [29]. It is a small and flexible general object instance segmentation framework that can achieve the best experimental results. Therefore, Mask R-CNN can be thought of as a Faster R-CNN [30] boundary box detection model with a small Fully Convolution Network (FCN) [31]. Mask R-CNN is an extension above Faster R-CNN, adding a layer for predicting the segmentation mask on each region of interest (ROI), called the mask branch. It can effectively detect the object in the image and can also generate a high-quality segmentation mask for each instance, so it is equivalent to multitask learning. Since the mask layer only adds a small amount of computation to the entire system, this method can simultaneously obtain object detection and instance segmentation.

This deep neural network convolves the entire input image to obtain the feature map. After the candidate region is generated by the RP network, the candidate frame is filtered by the method of improved nonmaximum suppression (NMS), and the candidate region corresponds to the high-dimensional feature vector on the feature map. Then, we use the full-convolution network to obtain the category of pixels in the image, calculate the score in the detection network, and finally output the detection and location results of the inpainted regions.

3.1. Architecture of the Proposed Method. This section describes the proposed method for patch-based image inpainting detection. The architecture of it is given in Figure 4. In this structure, the backbone network utilizes the 101-layer deep residual network ResNet and Feature Pyramid Network (FPN). ResNet is a framework for remnant learning that reduces the burden of network training. The network has a deeper level than the networks used before. It uses this layer to associate with the input layer to learn the residual function. FPN extracts ROI features from different levels of features based on feature size and corrects each ROI using ROI Align. After getting the feature map of each ROI region, the classification and bounding box of each ROI are predicted. Each ROI uses the designed FCN framework to predict the category of each pixel in the ROI region. Finally, a segmentation result of the image inpainted region is obtained.
The designed idea of the algorithm is as follows: manipulated localization, manipulated recognition, and semantic segmentation work together to achieve the purpose of enhancing object detection. Feature extraction is performed by the first backbone convolutional network ResNet, which obtains RoIAlign and is different from the RoIPool obtained by Faster R-CNN for the next object detection process. Then, the obtained RoIAlign is divided into two branches: one branch gains the boundary box and the object class through the fully connected network and the other branch gains the pixel-level instance segmentation result of the inpainted area through the calculation of the convolutional layer. The two branches are combined to gain the ultimate manipulated detection result. The adjunction of instance segmentation also improves the precise manipulated recognition, and the obtained detection information is more useful.

This process is simulated by a fully convoluted convolutional neural network. The network of RPN [30] is very small and it is shown in Figure 5, and the red box is a window that slides on the feature pyramid of the above trunk. Fully concatenate with the input convolutional feature map and map the sliding window to a low-dimensional vector, usually to 256 or 512 dimensions. The vector is put into two fully connected layers for classification and border regression. When the number of anchors is chosen to be 3, the effective area of the input image is the largest. At the position of each sliding window, we simultaneously predict $k$ proposal regions, and the box regression layer will generate $4k$ coordinate codes of $k$ boxes. The object recognition layer for each recommended area or the nonobject likelihood frame recognition layer will generate 2$k$ scores. The $k$ proposals are parameterized and are referred to as anchors. Each anchor is concentrated in the middle of the sliding window in the identification problem. Usually, $k$ scales and 3 aspect ratios are used, resulting in $k = 9$ anchors at each sliding window position. For size $W \times H$ (usually less than 2400), there are typically $WHk$ anchors. An important feature of this method is that it has translation invariance and has this property in both the computational anchor and the function of calculating correlations and anchors. In contrast, multiframes using the K-means method to generate anchors do not have translation invariance. If the object position in an image changes, the proposed object should also change, so the prediction of the proposal object could be at any position.

### 3.2. Improved NMS.

The sizes of anchor scales are adjusted to (16, 32, 64, 128, 256) due to the inpainted image data. We use Soft-NMS [19] instead of the original NMS. Because it uses a scoring system to score a lower score than the threshold of IOU, instead of deleting it directly.

After the detection frame is generated in the object detection, the traditional NMS [28] can eliminate the repeated bounding box. Generally, the detection box is sorted according to the score, the box with the highest score will be selected, and other boxes with the coincidence degree exceeding the threshold will be removed. However, this method of uniformly zeroing the detection frames that exceed the threshold is easy to make objects that are closer and have occlusions not detected. Therefore, we use an improved nonmaximum suppression (NMS) [32] instead, which uses a scoring system to evaluate scores below the Intersection over Union (IoU) threshold, rather than directly deleting them.

The improved NMS is shown in the following equation:

$$
s_i = \begin{cases} 
  s_i, & \text{IoU}(M, b_i) < N_i, \\
  s_i(1 - \text{IoU}(M, b_i)), & \text{IoU}(M, b_i) \geq N_i.
\end{cases}
$$  

Among them, $N$ is the threshold, $M$ is the box with the highest current score, $s_i$ is the corresponding score, and $b_i$ is the box to be processed. The improved NMS does not directly remove the detection frame that exceeds the threshold with the current highest score frame but reduces the score of the detection frame. When the degree of coincidence of the detection frame is greater, its score will also decay more. If there is only a small part of the overlap, it will not be much impact on the original detection score. In this way, the prediction frame will not be deleted by mistake when the objects of the same type are highly overlapped, and the improved NMS algorithm does not introduce any hyperparameters during the training. The hyperparameters used to adjust the improved NMS algorithm will only appear in the test or demonstration phase and will not cause an increase in computational complexity.

### 3.3. Loss Construction.

The loss function of the RPN is defined as the proposed object should also change, so the prediction of the proposal object could be at any position:

$$
L_t = \frac{1}{N_{\text{cls}}} \sum_t L_{\text{cls}}(s_t, s'_t) + \lambda \frac{1}{N_{\text{reg}}} \sum_t s_t^r c(t, t'_t).
$$  

\[L_t = \sum_t \left( \log p_t - \log \frac{p_t}{\sum_t p_t} \right) + \lambda \sum_t \sqrt{s_t^r c(t, t'_t)}.
\]
The RPN network consists of two parts: one is to judge whether the bounding box is the foreground or the background and the other is to predict the regression of the bounding box. The loss function of the corresponding RPN network is also composed of two parts, as $L(\{s_i\}, \{t_i\})$, where $L_{cls}$ is the partial loss function of the classification, $L_{reg}$ is the regression loss function of the bounding box, $N_{cls}$ is the minibatch size, $N_{reg}$ is the anchor location, and the weighted sum of the two partial loss functions is the total loss function of the RPN.

Here, $i$ is the anchor index, $s_i$ and $s_i^*$ are the probability that the bounding box is predicted to be the foreground and the probability that the prediction box is the foreground, $t_i$ and $t_i^*$ are the coordinates of the predicted border and the coordinates of the ground-truth border, respectively, and $s_i^* L_{reg}$ indicates that the frame is the foreground. It performs regression calculations.

The model initialized by ImageNet training is finetuned end-to-end by region proposal. Then, it uses the proposal generated before to train the detection network through Fast R-CNN segmentation. It is also initialized by the ImageNet training model. The third step is to initialize the regional proposal training network through the object detection network, but only the shared convolution layer and fine-tuned region proposal network. Now the two networks share the convolution layer.

Finally, the shared convolution layer is fixed, so that the two networks share the convolution layer to form a unique network. For pixel-level segmentation, it is parallel to the above box regression and object recognition. The multitask loss function is defined for the region of interest:

$$L_{Total} = L_{cls} + L_{box} + L_{mask}.$$ (4)

Due to the fact that the prediction of masks is depending on the region proposals, the loss of mask must be added to the total loss function. This can make region proposals more precise. In general, the prediction of masks and the region proposals complement each other and ultimately improve the accurate positioning of the inpainted boundary.

4. Experimental

In this paper, all experiments are performed on Ubuntu 16.04 of NVidia GeForce GTX 2080 Ti, operating in an Intel Core CPU i7-9700K. The sizes of anchor scales are adjusted to (16, 32, 64, 128, 256) due to the inpainted image dataset.

4.1. Dataset and Evaluation Metrics

4.1.1. Dataset. We select two typical image databases for experiments in our inpainting detection network. First, in the COCO [33] dataset, we randomly selected $2 \times 10^4$ color images and selected all 2,000 color images in the UCID [34] dataset; the size is cropped into $256 \times 256$. The inpainted images generated in the dataset are all repaired using the Criminisi algorithm of [28], which have different tampering areas in sizes and shapes. Half of the images are selected with some regular masks, resulting in tampering rates of 5%, 10%, and 20% but randomly selected for tampering. For the remaining half of the image, the inpainting area has an irregular shape, and the inpainting ratio varies between 1% and 50%. Several sample images are shown in Figure 6 (masked area is marked in green). A ground-truth tag matrix for repairing an image is formed depending on the tamper region used. Finally, the inpainted images with the corresponding ground-truth are split into two datasets: the part containing 80% of the images are training and validation dataset and 20% of the images are the testing dataset. Separate the training and the testing set to make sure that equal background and inpainted operation do not occur between them.

4.1.2. Evaluation Metrics. We choose True Positives Rate (TPR), False Positives Rate (FPR), and Accuracy Precision (AP) as evaluation metrics standard. Compare the performance of their detection in the same dataset with the approaches in [35, 36]. One of the algorithms for selecting contrast is that the traditional method has a better effect, and the other is the latest method of using the deep learning algorithm.

4.2. Forgery Detection. Figure 7 shows the detection results of the inpainted images (the first two column of Figure 6) using our method. We can see the detection bounding box, class and confidence score in the resulting image. This proves that the method of this paper can accurately detect and locate the tampering area (inpainted by the method in [28]).
Some examples of image inpainting detection can be seen in Figures 8 and 9, we find these three test algorithms from left to right: [35, 36] and our method can basically distinguish the inpainted area which has been listed in Figure 6. The last column of this figure shows that we can get more accurate pixel-level positioning in the detection using our method in different inpainted shapes. The other two methods have different degrees of false alarm pixels, and our method basically does not show false alarm pixels. Compared with [35, 36], our deep neural network provides complete and accurate detection on each measured image (closer to the ground truth in Figure 6).

For these three forensic methods, Table 1 summarizes three evaluation metrics averaged on the test datasets. The experimental results show that our network performs the highest True Positives Rate of 96.7%, the lowest False Positives Rate of 1.9%, and the tampering rate of more than 30% on the test images with irregular tampering areas. At the same time, we can also see that the detection performance decreases at the slow pace as the tampering ratio decreases. Moreover, our method still achieves less than 1.6% FPR with the images without restoration (extreme case). This demonstrates that our intelligent neural network can definitely capture the inpainting features. Similar experiments are carried out on circular or rectangular inpainted areas, our network shows slightly less effect under the regular shape mask. We suspect this is because our network has obtained the shape characteristics of the inpainted areas, which may have an adverse effect. Among those different shapes of inpainted areas we tested in the experiments, the performance of our method in FPR and AP is obviously better than that of other methods [35, 36].

4.3. Experimental Results. Considering tampering images are often attacked by JPEG compression and image scaling, we test the robustness of our proposed approach in Tables 2 and 3. Actually, we randomly select 3000 images from COCO [33] dataset. Then, the images with irregular tamper area and tamper rate greater than 30% are generated. In order to get the tempered images for robustness testing, we compress those images with 65%, 70% 80%, 90%, and 95% quality factor (QF), respectively. For the image scaling, these selected images are scaled in the range of 0.5, 0.75, and 1.5, respectively. The comparisons of ROC curve under two different attacks are shown in Figure 10.
Figure 8: The above regular inpainted images (1st column) and the corresponding tampered region detection results obtained by the methods proposed in [35] (2nd column) and [36] (3rd column), and our method (4th column).

Figure 9: Some irregular inpainted images (1st column) and the corresponding tampered region detection results obtained by the methods proposed in [35] (2nd column), and [36] (3rd column), and our method (4th column).
According to the experimental results (Table 2 and the ROC curve in Figure 10), compared with the test image without JPEG compression, the detection accuracy of the method used in the paper is slightly less than 96.7%, and the low-quality compression strength at 65% also has an accuracy rate of more than 80%, which is better than method [35].

Table 1: Performance experiments of three methods with different evaluation metrics standards.

<table>
<thead>
<tr>
<th>Region shape</th>
<th>Regular</th>
<th>Irregular</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tamper ratio</td>
<td>TPR</td>
<td>FPR</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Method [35]</td>
<td>81.2</td>
<td>87.8</td>
</tr>
<tr>
<td>FPR</td>
<td>3.1</td>
<td>8.5</td>
</tr>
<tr>
<td>AP</td>
<td>94.5</td>
<td>91.4</td>
</tr>
<tr>
<td>Method [36]</td>
<td>85.3</td>
<td>82.5</td>
</tr>
<tr>
<td>FPR</td>
<td>2.3</td>
<td>2.5</td>
</tr>
<tr>
<td>AP</td>
<td>94.5</td>
<td>95.2</td>
</tr>
<tr>
<td>Proposed</td>
<td>TPR</td>
<td>89.3</td>
</tr>
<tr>
<td>FPR</td>
<td>1.2</td>
<td>1.5</td>
</tr>
<tr>
<td>AP</td>
<td>97.5</td>
<td>96.4</td>
</tr>
</tbody>
</table>

Table 2: The accuracy performance of different methods under five JPEG compression (with quality of 65, 70, 80, 90, and 95).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>44.5</td>
<td>76.8</td>
<td>82.7</td>
</tr>
<tr>
<td>70</td>
<td>53.8</td>
<td>85.4</td>
<td>87.4</td>
</tr>
<tr>
<td>80</td>
<td>69.5</td>
<td>89.7</td>
<td>92.1</td>
</tr>
<tr>
<td>90</td>
<td>82.4</td>
<td>91.3</td>
<td>94.2</td>
</tr>
<tr>
<td>95</td>
<td>92.4</td>
<td>93.8</td>
<td>96.7</td>
</tr>
</tbody>
</table>

Table 3: The accuracy performance of three methods under different scaling conditions (with scaling factors 0.5, 0.75, and 1.5).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>61.2</td>
<td>87.9</td>
<td>90.1</td>
</tr>
<tr>
<td>0.75</td>
<td>75.3</td>
<td>90.6</td>
<td>93.4</td>
</tr>
<tr>
<td>1.5</td>
<td>84.6</td>
<td>91.3</td>
<td>92.7</td>
</tr>
</tbody>
</table>

Figure 10: ROC curve comparison of three forensic methods under different attack operations. (a) ROC curve comparison (95% JPEG compression quality) and (b) ROC curve comparison (scaling ratio 1.5).
[36] under different compression attacks. The traditional detection method [35] is greatly affected by JPEG compression. When the compression factor is lower than 80%, the accuracy rate does not exceed 60%. Experimental results show that our proposed network can maintain the robustness of JPEG compression. Both Table 3 and the ROC curve show that when the scaling factor is gradually away from 1, AP is less affected in our proposed method, method [36] has a certain effect, and method [35] has the greatest effect. Therefore, our method is insensitive to scaling. In conclusion, our method based on the deep neural network can keep robust to JPEG and scaling attacks.

Although the traditional detection method [35] uses a search algorithm based on weight transformation to speed up the detection and reduces the search time for matching blocks, it usually takes several minutes. The intelligent detection algorithms using the deep learning method, our proposed method, and method [36], in addition to consuming some time during training, generally only take a few seconds during detection and can basically be detected in real time.

5. Conclusion

This paper proposes an effective and intelligent image inpainting detection approach using the deep neural network. In order to distinguish and obtain different features between the inpainted and noninpainted regions, this paper applies the improved region detection network here, converts the object detection problem to a pixel-level segmentation, and uses the mask to create the bounding box. The nonmaximum suppression, especially, is defined by using the overlap ratio of the mask area instead of the bounding box to filter the output result. Adjust the anchors scale and the step size and use the improved NMS of the area suggestion network according to the object morphological in order to generate a more accurate region of interest. This paper proves that the state-of-the-art instance segmentation model can obtain the differences of features in inpainted images. In the future, we will optimize the network and increase the tampering dataset, which may improve the effectiveness of different image tampering forensics.

Data Availability

The tables and figures' data used to support the findings of this study are included within the article. Previously reported tables and figures' data were used to support this study and are available in the relevant references. These prior studies (and datasets) are cited at relevant places within the text as references [31–36]. The data supporting this research are from previously reported studies and datasets, which have been cited. The processed data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61370195 and U1536121).

References


