symmetry

MDPI

Article

Developing an Image Manipulation Detection Algorithm Based on Edge Detection and Faster R-CNN

Xiaoyan Wei [1], Yirong Wu [1], Fangmin Dong [1], Jun Zhang [2] and Shuifa Sun [1,2,*]

[1] College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China; m15272137281@163.com (X.W.); yirongwu@gmail.com (Y.W.); fmdong@ctgu.edu.cn (F.D.)

[2] Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee, Milwaukee, WI 53201, USA; junzhang@uwm.edu

* Correspondence: watersun@ctgu.edu.cn

Received: 14 August 2019 Accepted: 19 September 2019; Published: 1 October 2019

Abstract: Due to the wide availability of the tools used to produce manipulated images, a large number of digital images have been tampered with in various media, such as newspapers and social networks, which makes the detection of tampered images particularly important. Therefore, an image manipulation detection algorithm leveraged by the Faster Region-based Convolutional Neural Network (Faster R-CNN) model combined with edge detection was proposed in this paper. In our algorithm, first, original tampered images and their detected edges were sent into symmetrical ResNet101 networks to extract tampering features. Then, these features were put into the Region of Interest (RoI) pooling layer. Instead of the RoI max pooling approach, the bilinear interpolation method was adopted to obtain the RoI region. After the RoI features of original input images and edge feature images were sent into bilinear pooling layer for feature fusion, tampering classification was performed in fully connection layer. Finally, Region Proposal Network (RPN) was used to locate forgery regions. Experimental results on three different image manipulation datasets show that our proposed algorithm can detect tampered images more effectively than other existing image manipulation detection algorithms.

Keywords: image manipulation detection; Faster R-CNN; edge detection; max pooling

1. Introduction

In modern society, image editing is becoming increasingly popular. With the simplification of image editing software, images can be edited even on one mobile phone. In various social networks, such as Facebook and Weibo, there are various kinds of edited images, some of which are maliciously distorted and tampered, making people unable to know the truth and even causing adverse effects. Therefore, accurate detection of tampered images is particularly important.

Roughly there are two types of image manipulation methods: image splicing and Copy–Move Forgery (CMF). Image splicing refers to pasting a part of an image into another image. CMF is to tamper image by pasting parts of an image into other areas of the image. Two examples of image manipulation are shown in Figure 1.

After image manipulation, post-processing operations are sometimes applied, such as gaussian smoothing, adding gaussian white noise, and JPEG compression. These post-processing operations make the tampering effect more realistic, which makes the task of identifying tampered images more challenging.

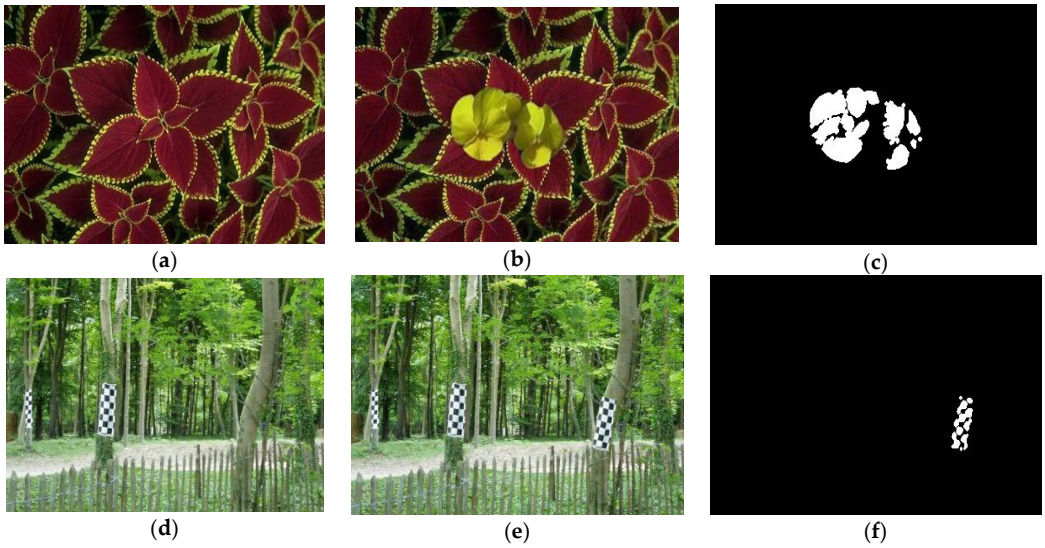

Figure 1. Example of image splicing and Copy–Move Forgery (CMF). (**a**) Authentic image of the splicing image. (**b**) The splicing image. (**c**) Ground-truth mask of the splicing image. (**d**) Authentic image of the CMF image. (**e**) The CMF image. (**f**) Ground-truth mask of the CMF image.

Tampered images can be identified by image forensics algorithms, which are mainly divided into active and passive forensics methods. Active forensics methods include watermarking [1,2] and digital signatures [3]. In active forensics methods, images have been pre-processed and imperceptibly marked. These changes can be easily detected. Passive forensics algorithms deal with completely unknown images with no prerequisites. Therefore, passive forensics is more challenging and has become an active research field for more than a decade.

Passive forensics algorithms, also known as image manipulation detection methods, are studied in this paper. Recent work on image manipulation detection algorithms focuses on certain features of tampered images, such as image compression features, image local noise features and Camera Filter Array (CFA) patterns, etc. Image compression features include block artifacts [4] and Block Artifact Grids (BAG) [5] generated by JPEG compression for checking image inconsistency. Then, the features of aligned double JPEG compression [6] and Non-Aligned double JPEG (NA-JPEG) compression [7] are introduced to detect image tampering. For image local noise features, the tamper traces are revealed based on the local noise variance models of wavelet filtering [8] and the kurtosis characteristics of the frequency sub-band coefficients [9] in the forged images. CFA demosaicing artifacts [10] were studied as digital fingerprint, and the existence of demosaicing artifacts can be measured even at the level of 2 × 2 blocks. These algorithms in recent work have limited applicability and low detection accuracy.

Recently, several new algorithms using deep learning were proposed to improve manipulation detection performance. Convolutional Neural Network (CNN) model [11] was developed to detect image splicing and CMF. A blind deep learning method based on CNN [12] was used to learn invisible discrimination artifacts from manipulated images. A constrained convolution layer [13] was proposed to adaptively learn manipulation detection features. CNN-LSTM architecture [14] was developed to detect image forgery by learning the edges of tampered and non-tampered areas. Leveraging Faster Region-based Convolutional Neural Network (Faster R-CNN) [15], a two-stream image manipulation detection model [16] was developed to detect different types of tampered images. However, it has modest detection performance for CMF images. In addition, it mainly aims to detect target tampering, while ignoring background tampering.

In this study, leveraging CNN-LSTM architecture [14] and Faster R-CNN two-stream image manipulation detection model [16], we develop an image manipulation detection algorithm based on Faster R-CNN model. Laplacian of Gaussian (LoG) operator [17] performs gaussian convolution filtering on the input image to overcome the influence of noise to some extent, but it may generate

false edges [18]. The Prewitt operator [19] can be used to remove some false edges. Therefore, our algorithm applies the edge detection LoG operator and Prewitt operator to images, obtains edge features, carries out end-to-end training, fuses the features in the bilinear pooling layer, and detects tampered images. Image tampering detection is different from object detection, since image tampering can be either background tampering or target tampering, which means that tampering may occur in any area of an image. Compared with the Faster R-CNN model, our algorithm adds an edge detection layer, finds inconsistency between tampered and non-tampered areas, removes RoI max pooling, uses bilinear interpolation method to fix the size of the interest area, avoids only extracting high-frequency information, and improves image tampering detection performance. Our algorithm performs well in Copy–Move Forgery Detection (CMFD) and image splicing detection. Specifically, the main contributions of this paper are as follows:

- The edge detection layer is added onto the Faster R-CNN model to extract edge features of original input images. Original input images and the images with edge features are put into the Faster R-CNN network in parallel for end-to-end training. We observe that the performance of image manipulation detection is improved on three benchmark datasets.
- We propose to remove the max pooling in the Region of Interest (RoI) pooling layer and fix the size of the RoI by bilinear interpolation, thus retaining more feature information. Experimental results show that our method is more accurate than the original max pooling approach in image manipulation detection.

2. Related Work

In the past ten years, many methods have been developed to detect low-level tamper artifacts in tampered images. Ye et al. [4] proposed a fast image manipulation detection method based on Discrete Cosine Transform (DCT) coefficient histogram, which detects a trace of forgery images by checking the inconsistency of block artifacts. Li et al. [5] developed a passive detection algorithm to detect tampered JPEG images using anomalous block artifact grids. Zhu et al. [6] proposed an improved double JPEG compression detection algorithm based on noise-free DCT coefficient mixture histogram model. Bianchi et al. [7] obtained a single feature from DCT coefficient statistics to detect NA-JPEG compression. This algorithm can effectively detect tampered JPEG images. Mahdian et al. [8] modeled the local noise variance by wavelet filtering, found local noise inconsistency, and identified forged images. Lyu et al. [9] proposed a method to detect regional splicing by revealing inconsistencies of local noise levels. This utilizes a particular regular property of the kurtosis of nature images in band-pass domains and the relationship between noise characteristics and kurtosis. Considering the traces left by CFA interpolation, Ferrara et al. [10] proposed a forensic tool to distinguish between the authentic area and the tampered area. These image manipulation detection algorithms only consider a limited set of features that tamper images, which hampers their detection performance.

Recently, CNN model [11] has been used to detect tampered images, which is the first example of using deep learning in image manipulation detection. Rota et al. [12] proposed a weak labeling method based on CNN, which can extract the tampered parts from a given forged image and generate segmentation region. Bayar et al. [13] developed a constraint convolution layer, which can suppress the content of the image and learn manipulation features adaptively. Bappy et al. [14] proposed a network model based on CNN-LSTM to detect different types of image manipulation, but the model only learns the boundary difference between manipulated and non-manipulated regions, ignoring other features of tampered images, which makes the detection performance not good enough. Salloum et al. [20] used the Multi-task Fully Convolutional Network (MFCN) framework to learn a ground-truth mask and spliced region boundary to predict tampered region mask for a specified image, which makes the tampering location more accurate. However, the detection accuracy is modest. Zhou et al. [16] proposed a two-stream Faster R-CNN model and trained it end-to-end to detect tampered regions in given images. One of the two streams is RGB stream, which aims to extract features from the RGB image to find tampering artifacts, such as strong contrast differences and unnatural tampering boundaries. The other is noise stream, which uses the noise features extracted

by Steganalysis Rich Model (SRM) filter [21] to find the noise inconsistency between the authentic areas and the tampered areas in manipulated images. However, this model is not ideal for detecting CMF images. Nevertheless, it provides an important basis for the construction of our algorithm. In this paper, the Faster R-CNN model is combined with the edge detection operators to learn the features of tampered images, and to perform image manipulation detection and forgery area localization.

3. The Faster R-CNN Model Combined with Edge Detection

The Faster R-CNN model [15] mainly comprises three parts: feature extractor, Region Proposal Network (RPN) and RoI pooling [22]. Leveraging the Faster R-CNN model, our model adds an edge detection layer to extract edge feature images, and input feature extractor in parallel with the original image to achieve feature mapping. We choose ResNet101, a residual network model [23] which is deeper than Visual Geometry Group (VGG) network [24], to learn the input image features. Different from the object detection [15,23], the forgery area of the manipulated image is likely to be the target region or the background region [13]. In the RoI pooling layer, max pooling operation extracts the high-frequency information in the image, which is more conducive to detect target tampering [22]. To overcome this problem, our algorithm removes the max pooling, uses bilinear interpolation to adjust the size of the region of interest and extracts more tampering information.

In this paper, we develop a Faster R-CNN-based image manipulation detection model (Figure 2). The detailed steps of our algorithm are described as follows: First, in the edge detection layer, original input images are convolved with the Prewitt operator and LoG operator to obtain the images with edge features. After the images with edge features are sent into the Faster R-CNN model in parallel with the original tampered images. ResNet101 network [23] is used to extract the features of tampered images, generating feature maps and edge feature maps of the original input images. They are sent to the RoI pooling layer to obtain both RoI features of the original input images and RoI features of the edge feature images. After two kinds of features are fused by bilinear pooling layer [25], the tampering classification is carried out in the fully connection layer. Finally, the features of the original input images are mapped to the RPN to locate the tampered areas [16]. The loss function of RPN network is described as follows [15]:

$$L_{RPN}(p_i, t_i) = \frac{1}{N_{cls}} L_{cls} \sum_i (p_i, p_i^*) + \lambda \frac{1}{N_{reg}} p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

where i represents the index value of anchors in a mini batch, p_i is the prediction probability of anchor i at the tampered area, and p_i^* is the ground-truth label where anchor i is positive. t_i, t_i^* are the vectors describing four parameterized coordinates of the bounding boxes and the ground-truth mask boxes, respectively. The classification loss L_{cls} denotes the cross-entropy loss of RPN network, and the regression loss L_{reg} denotes the smoothing L_1 loss of the proposed bounding boxes. L_{cls}, L_{reg} are normalized by N_{cls} and N_{reg} respectively, and weighted by the balance parameter λ (λ is set to 10 [15]). N_{cls} denotes the mini batch size of RPN network, and N_{reg} denotes the number of anchors.

We use the cross-entropy loss in tampering classification, and the bounding box regression for smoothing L_1 loss [16]. The total loss function is described as follows:

$$L = L_{RPN} + L_{tamper}(f, f_{edge}) + L_{bbox}(f) \quad (2)$$

where L is the total loss function. L_{RPN} represents the RPN loss function in RPN network. L_t represents the final cross-entropy classification loss, which is a bilinear pooling feature based on the original image input and edge feature image input. L_{bbox} represents the final bounding box

regression loss. f_r is the RoI features of the original input image. f_e is the RoI features of the edge feature image.

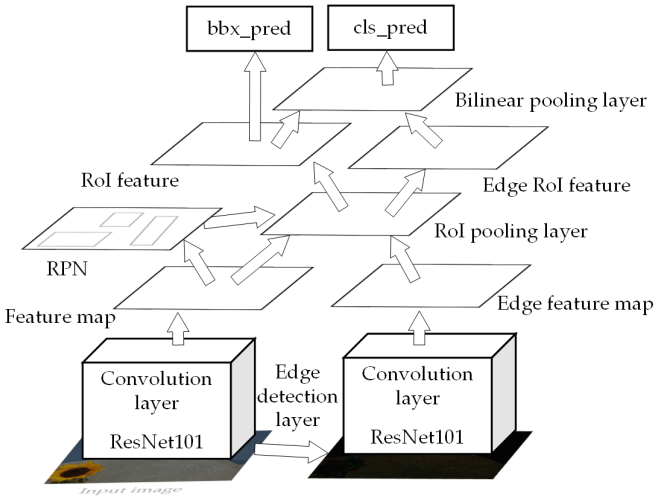

Figure 2. Faster Region-based Convolutional Neural Network (Faster R-CNN) image manipulation detection model.

3.1. Adding Edge Detection Layer

Whether image manipulation type is image splicing or CMF, it has visual inconsistency and obvious difference at the boundary of the tampered and non-tampered areas. Considering this information, edge features of the tampered area and tampered image are introduced to detect forgery image. The classical LoG operator and Prewitt operator in the field of edge detection are used to extract edge features from the original tampered images. They are then input into the Faster R-CNN model in parallel with the original tampered images for end-to-end training. Examples of edge feature images are shown in Figure 3.

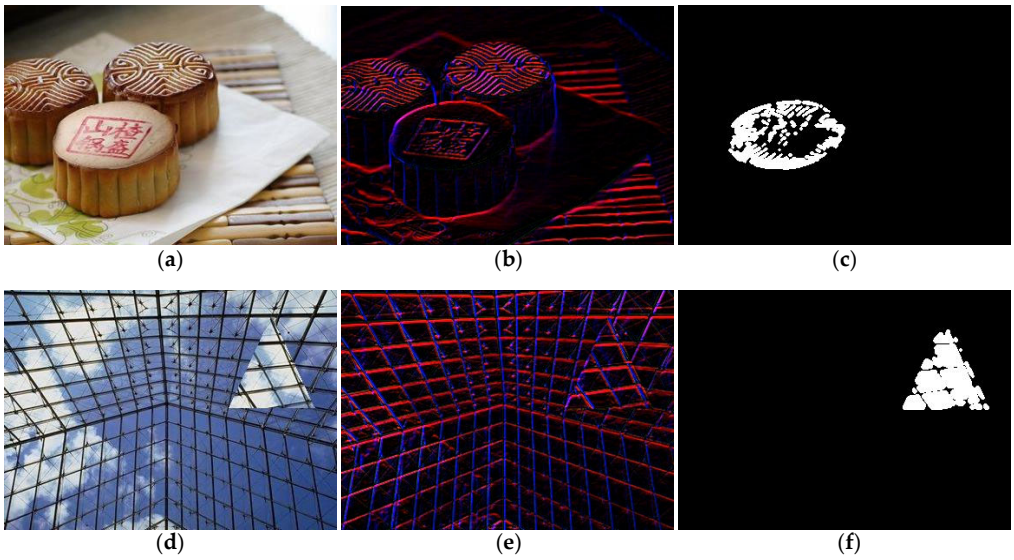

Figure 3. Examples of edge feature images. (**a**,**d**) Authentic images of the tampered images. (**b**,**e**) The tampered images. (**c**,**f**) Ground-truth masks of the tampered images.

LoG edge detection operator is expressed as:

$$LoG(f) = \frac{x^2+y^2-2\sigma^2}{\sigma^4}\exp(-\frac{x^2+y^2}{2\sigma^2}) \tag{3}$$

where σ is the standard deviation. The LoG edge detection operator performs gaussian convolution filtering for noise reduction of the input image, and then adopts Laplace operator for edge detection [17]. This procedure overcomes the influence of noise to some extent, but it may produce false edges [18]. If only one LoG operator is used to extract the edge features of the tampered images, the tampering detection performance is not ideal.

Prewitt edge detection operator is a first-order differential operator, which can be expressed as:

$$\begin{aligned} P_x = &\{f(i+1, j-1) + f(i+1, j) + f(i+1, j+1)\} \\ &-\{f(i-1, j-1) + f(i-1, j) + f(i-1, j+1)\} \end{aligned} \tag{4}$$

$$\begin{aligned} P_y = &\{f(i-1, j+1) + f(i, j+1) + f(i+1, j+1)\} \\ &-\{f(i-1, j-1) + f(i, j-1) + f(i+1, j-1)\} \end{aligned} \tag{5}$$

where P_x and P_y represent the Prewitt operator in horizontal and vertical directions, respectively. They use gray difference of pixel points at two sides to detect the edge, which can also be used to remove a part of the false edge [19]. There is a great difference in pixel grayscale at the boundary of the authentic and tampered regions in the forgery images. LoG operator and Prewitt operator in both horizontal and vertical directions can effectively extract the edge information of the tampered images. The mean average precision (mAP) values detected on the Common Objects in Context (COCO) synthetic tampering dataset are shown in Table 1.

Table 1. Mean average precision **(**mAP) of different edge detection operators.

Algorithms	mAP
Faster R-CNN + LoG	0.7400
Faster R-CNN + LoG + P_x	0.7635
Faster R-CNN + LoG + P_y	0.7582
Ours	**0.8100**

Three kernels are selected in the edge detection layer (increasing the number of kernels does not improve performance). Their weights are shown in Figure 4. We feed these three kernels directly into a pre-training network with input channel of 3. The kernel size in the edge detection layer is defined as $5*5*3$. The output channel of the edge detection layer is 3.

$$\frac{1}{16}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

LoG operator Prewitt operator

Figure 4. Three edge detection kernels used to extract edge features.

3.2. Improving RoI Pooling

RoI pooling layer can use max pooling to convert all regions of interest to rectangular windows with size of $H \times W$ ($H \times W$ is set to be 7×7 and 16 is feat_stride in this paper), where H and W are hyper-parameters. Each RoI window is represented as a four-tuple (r, c, h, w), specifying its

upper-left coordinate (r, c), its height h and width w. The goal of RoI max pooling is mainly to divide the RoI window of $h \times w$ into sub-windows of $H \times W$, and then collect the maximum value in each sub-window into the corresponding output grid cell through max pooling. The number of sub-windows is about $h/H \times w/W$ [22].

The max pooling operation focuses on the features of the target object only and ignores the features from background. However, in image manipulation detection, either the background tampering or the target tampering can occur. Therefore, in order to improve the detection accuracy and simplify the model, we propose to remove the max pooling, and convert the RoI area (r, c, h, w) into windows with the size of $H \times W$ by bilinear interpolation, which is then sent into the fully connection layer for manipulation detection. The workflow is shown in Figure 5.

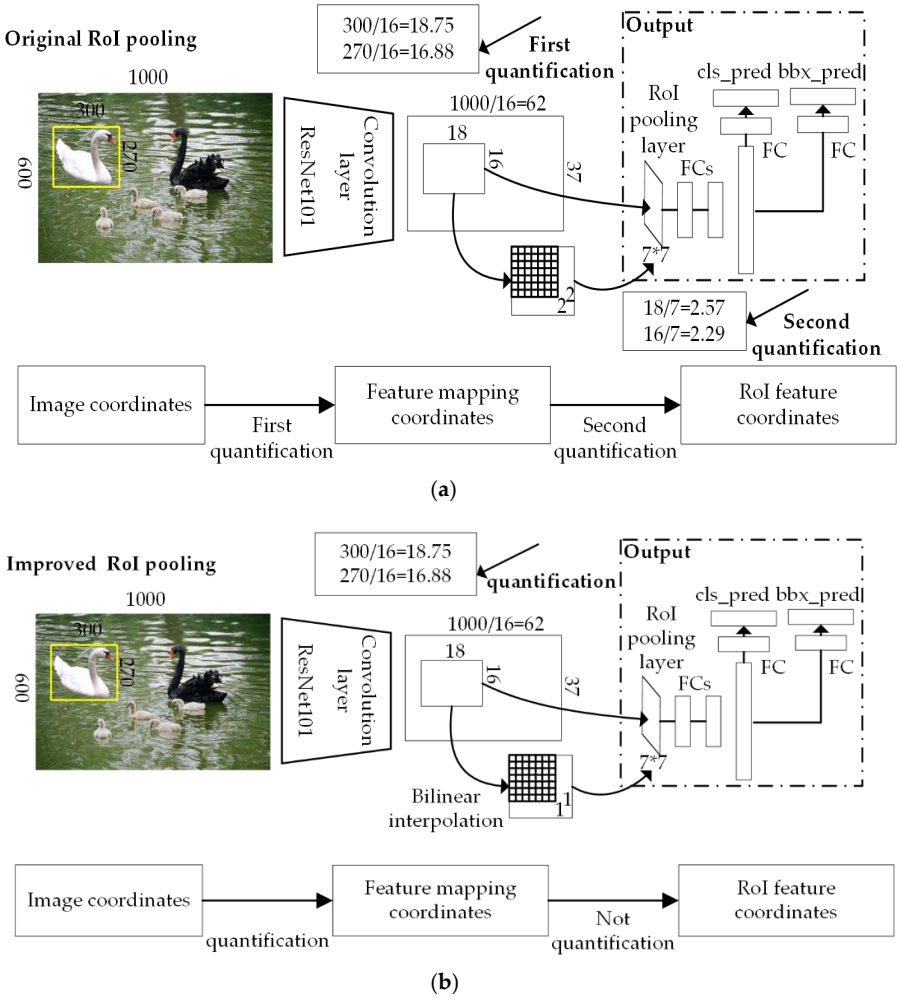

Figure 5. Flowchart of Region of Interest (RoI) pooling (an example of the input image size of 1000 × 600). (**a**) Original RoI pooling. (**b**) Improved RoI pooling.

4. Experimental Results and Analysis

4.1. Pre-Training Model

Currently, the number of published benchmark image manipulation datasets is small. When our model was trained, COCO synthetic tampering dataset [16] was chosen for pre-training. Table 2 shows the numbers of images for its training and testing.

Table 2. Distribution of pre-training training and testing images.

Datasets	Image Number
Training	11,650
Testing	1833

During the training process, most parameters were basically the same as Faster R-CNN. The scale parameters of anchors were 8, 16, 32, 64, the proportion parameters were 0.5, 1, 2, and the learning rate was set to 0.001 [16]. The learning rate was reduced to 0.0001 after 40 K steps. We trained our model for 120 K steps. Non-Maximum Suppression (NMS) was applied to the proposed regions to reduce the overlap. Table 3 shows the influence of test Intersection over Union (IoU) thresholds of NMS on experimental results. When the threshold is less than 0.5, mAP gradually increases, and when it exceeds 0.5, it shows a downward trend. Therefore, the test IoU threshold of NMS is set as 0.5. We used mAP as an evaluation index, which was consistent with the Faster R-CNN evaluation metric.

Table 3. The effect of Intersection over Union (IoU) threshold of Non-Maximum Suppression (NMS) on experimental results.

IoU Threshold	mAP
0.3	0.8066
0.4	0.8084
0.5	**0.8100**
0.6	0.8056
0.7	0.7966

In order to ensure the fairness of comparison, both RGB-N and our algorithm were trained on the same pre-training dataset with the same parameters. Their mAP values from the testing of COCO synthetic tampering dataset are shown in Table 4.

Table 4. mAP of COCO synthetic tampering dataset.

Models	Max Pooling	mAP
RGB-N [16]	√	0.7645
RGB-N [16]		0.7837
Ours	√	0.7459
Ours		**0.8100**

Table 4 shows that our algorithm produces a higher mAP than RGB-N method when the RoI max pooling is removed, but a lower mAP than RGB-N method when the RoI max pooling is used. This is because the RoI max pooling selects high-frequency features in tampering classification; the noise stream in the RGB-N method also extracts high-frequency information from the tampered images for training in the Faster R-CNN network. For our algorithm, the LoG operator in the edge detection layer removes part of noise interferences and high frequency components through gaussian filtering, so it performs better when removing the RoI max pooling.

4.2. Image Manipulation Detection

4.2.1. Datasets

In this paper, experiments and analysis were conducted on three publicly available benchmark image manipulation datasets: NIST2016 [26], Columbia [27] and CASIA [28]. NIST2016 is an image manipulation dataset provided by the Nimble challenge for postprocessing to hide visible traces and used as ground-truth masks for evaluation. Columbia dataset is uncompressed, containing only splicing images, without any post-processing. It also provides ground-truth mask. There are two

versions of CASIA dataset. Version 1.0 is a smaller dataset containing 921 tampered images, while Version 2.0 is a larger dataset containing 5123 tampered images. The CASIA dataset is post-processed and no ground-truth masks are provided. By comparing the difference between the authentic image and the tampered image, the labeling software 'LabelImg' was used to mark the tampered area. We used Version 2.0 as the training and Version 1.0 as the testing images. The distribution of training and testing images in three benchmark datasets is shown in Table 5.

Table 5. Training and testing images in three benchmark datasets.

Datasets	NIST2016	Columbia	CASIA
Training	404	-	5123
Testing	160	180	921

Data augmentation. Currently, the amount of data in the publicly available benchmark image manipulation datasets is small. The data volume was expanded by means of data enhancement to make the network-trained model more accurate. In this study, image flipping was used to double the original dataset. On those three benchmark datasets, our method's Recall, Precision, F1 scores were compared between the image flipping and no image flipping. The results show that image flipping can improve the performance of our algorithm, as shown in Table 6.

Table 6. Comparison of the results of our method between image flipping and no image flipping.

Datasets	Flipping	Recall	Precision	F1 scores
NIST2016		0.9563	0.9217	0.9387
	√	**0.9563**	**0.9503**	**0.9533**
Columbia		0.7333	0.7416	0.7374
	√	**0.7389**	**0.7644**	**0.7514**
CASIA		0.6566	0.4865	0.5589
	√	**0.6608**	**0.5158**	**0.5794**

4.2.2. Test Results and Analysis

Based on the performance of the pre-training model, our model was fine-tuned on three benchmark datasets. We also used F1 score, Recall and Precision as evaluation metrics. The highest F1 score was used as the final score for each image by changing different thresholds, following the same protocol in [16,20,29].

With the evaluation platform provided in [29], F1 scores of four traditional image manipulation detection algorithms on NIST2016, Columbia and CASIA datasets were obtained, which includes DCT [4], Noise (NOI1) [8], Block (BLK) [5] and Camera Filter Array (CFA1) [10]. MFCN results were obtained from the original literature [20]. RGB-N [16] results were obtained on the basis of the pre-training model. F1 scores, Recall and Precision of our method and existing image manipulation algorithms are shown in Tables 7–9.

Table 7. F1 scores of our algorithm and existing algorithms in benchmark datasets.

Algorithms	NIST16	Columbia	CASIA
DCT [4]	0.2756	0.5199	0.3005
NOI1 [8]	0.2850	0.5740	0.2633
BLK [5]	0.3019	0.5234	0.2312
CFA1 [10]	0.1743	0.4667	0.2073
MFCN [20]	0.5705	0.6117	0.5410
RGB-N [16]	0.9123	0.7467	0.5655
Ours	**0.9533**	**0.7514**	**0.5794**

Table 8. Recall of our algorithm and existing algorithms in benchmark datasets.

Algorithms	NIST16	Columbia	CASIA
DCT [4]	0.3625	0.4833	0.2237
NOI1 [8]	0.2938	0.5611	0.1748
BLK [5]	0.2562	0.4500	0.1705
CFA1 [10]	0.1500	0.6278	0.1857
RGB-N [16]	0.9437	**0.7944**	0.6515
Ours	**0.9563**	0.7389	**0.6608**

Table 9. Precision of our algorithm and existing algorithms in benchmark datasets.

Algorithms	NIST16	Columbia	CASIA
DCT [4]	0.2223	0.5626	0.4576
NOI1 [8]	0.2767	0.5875	0.5333
BLK [5]	0.3674	0.6254	0.3590
CFA1 [10]	0.2080	0.3714	0.2346
RGB-N [16]	0.8830	0.7044	0.4996
Ours	**0.9503**	**0.7644**	**0.5158**

It can be seen from Tables 7–9 that our method provided high performance than other image manipulation detection algorithms. This is because DCT, NOI1, BLK and CFA1 algorithms are mostly based on a single tamper feature, which only contains localized partial tamper information. This limits the detection performance of these algorithms. MFCN algorithm is trained on the ground-truth mask and the edge of the spliced region, indicating whether each pixel belongs to the boundary of the tampering region. It only pays attention to the boundary information, resulting in unsatisfactory detection performance. The addition of noise features in RGB-N algorithm is more effective for detecting splicing images, while for CMF images, due to tampering within an image, the noise difference generated is small. Therefore, RGB-N algorithm is not as good as our method on the NIST2016 and CASIA datasets. The Columbia dataset only contains splicing images, so the Recall of the RGB-N algorithm on this dataset is higher than that of our method.

Our algorithm combined the Faster R-CNN model with the edge detection method to learn not only the boundary features of tampered images, but also other features of tampered images, such as contrast difference, etc. Instead of the RoI max pooling, the bilinear interpolation method was used, such that the region of interest contains not only the target information but also the background information, which can be detected for both target tampering and background tampering. Consequently, the detection results of our algorithm are superior to other image tampering detection algorithms.

4.2.3. Robustness

To verify the robustness of our algorithm, we compared detection performance between our algorithm and other methods on CASIA dataset for different gaussian white noise (with mean of 0, variance of 5, 10, 15), different gaussian blurring (filter size 3 × 3, variance were 0.5, 1.0, 1.5) and different JPEG compression quality factor (85, 70, 55). Their F1 scores, Recall and Precision are shown in Figure 6.

The results in Figure 6 show that our method still has better detection performance than other image manipulation algorithms after adding gaussian white noise, gaussian blurring and JPEG compression attack. When gaussian white noise was added, RGB-N algorithm had a smaller decline in F1 scores than our method. The anti-noise performance of our method was not as good as that of RGB-N algorithm. After gaussian blurring processing, the F1 score of our algorithm declines more slowly than other image manipulation methods, which is better than other image manipulation detection methods. Under the attack of JPEG compression, the performance degradation of our method and RGB-N algorithm was basically the same. They are inferior to other traditional image manipulation detection algorithms.

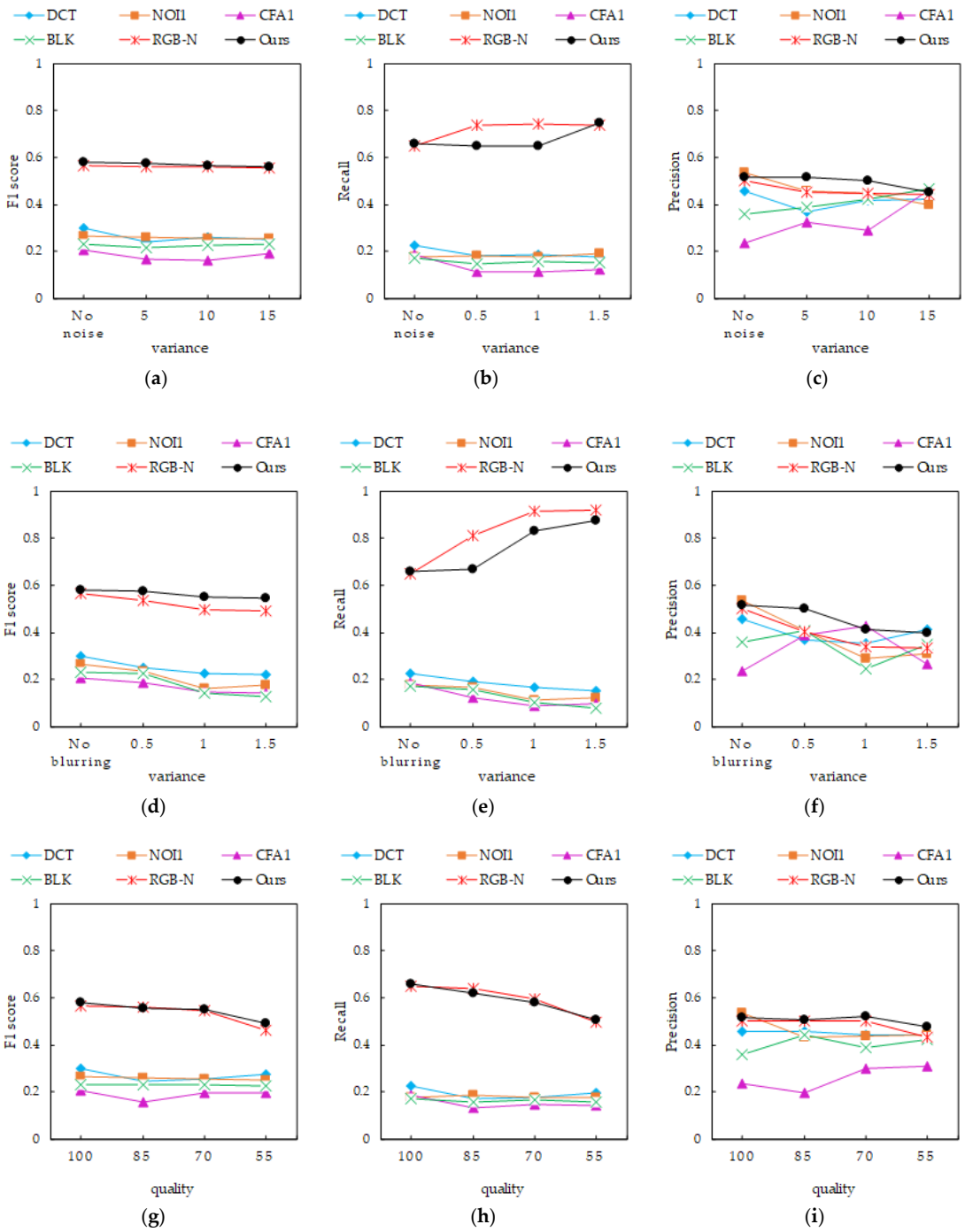

Figure 6. Algorithms comparison. (**a**–**c**) F1 score, Recall and Precision under different gaussian white noise attack. (**d**–**f**) F1 score, Recall and Precision under different gaussian blurring attack. (**g**–**i**) F1 score, Recall and Precision under different JPEG compression attack.

4.3. Localization of the Tampered Region

Figure 7 describes the localization performance of RGB-N algorithm and our method on tampered images. The images are from NIST2016 and CASIA datasets. Figure 7 shows that our algorithm is more accurate than RGB-N algorithm in locating tamper areas.

For CMF images, compared with RGB-N algorithm, our method can select the tampered regions accurately. In Figure 7a, RGB-N algorithm is imperfect to select tampered areas; the predicted bounding box of our method can frame the complete tampering area. According to the ground-truth mask Figure 7b, the tampering area is only the small boat on the left of the image. Our method can select the left side boat in this image correctly but RGB-N algorithm selects the tampered area incorrectly. RGB-N algorithm chooses the left side boat and the right-hand big ship in the image. Briefly, our method is more accurate than RGB-N algorithm in CMF.

For image splicing, the performance of our method is also better than of RGB-N algorithm. In Figure 7c, our method can box out the tampered lion completely, while RGB-N algorithm unboxes a small part of the lion's head and tail in the predicted bounding box. As shown in Figure 7d, RGB-N algorithm produces more untampered areas than our method. In one word, our method and RGB-N algorithm can correctly predict tampering areas, but the predicted bounding boxes of our method are closer to the ground-truth mask in splicing images.

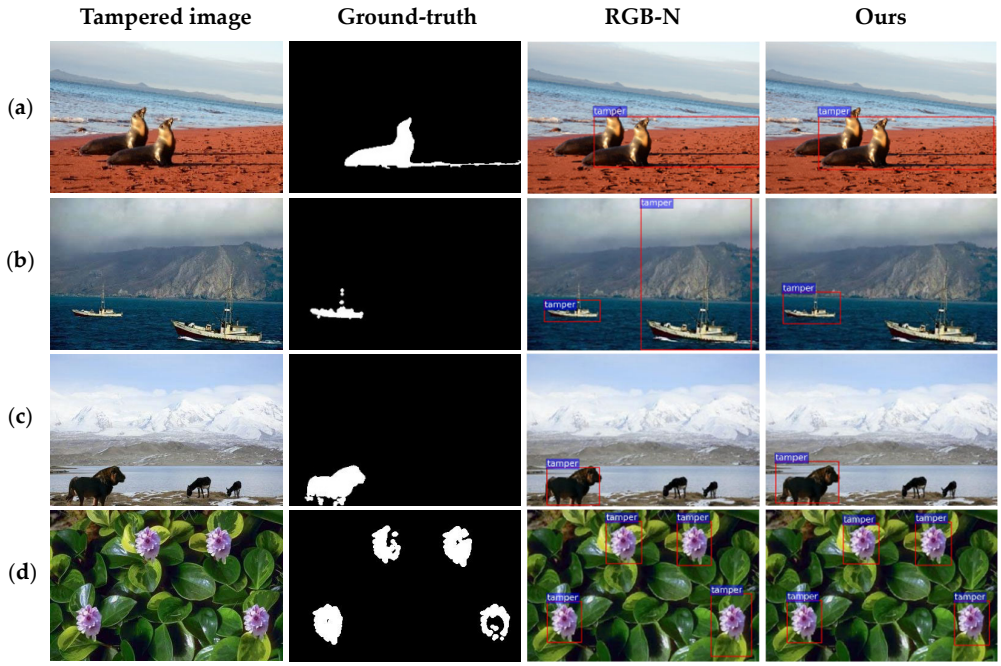

Figure 7. Performance comparisons between RGB-N and our algorithm. (**a**,**b**) CMF images. (**c**,**d**) Splicing images. The second column is ground-truth mask of (**a**–**d**) images from top to bottom. The third column and the last column are the tamper location results of RGB-N algorithm and our method respectively, and the tamper areas are predicted in red rectangular boxes.

5. Discussion and Conclusions

In this study, combining traditional edge detection algorithms with the Faster R-CNN model, a new image manipulation detection algorithm is developed to learn rich tampered image features for manipulation detection and localization. By using edge detection operators to extract features in the added edge detection layer, our algorithm can capture the edge inconsistency between the tampered area and the non-tampered area. In RoI pooling layer, our algorithm uses bilinear interpolation to adjust the size of the region of interest size instead of max pooling to extract more tampering information.

In this paper, we evaluate the performance of our proposed method through a series of experiments. Experiments on three benchmark datasets show that our proposed method is better than current image manipulation detection algorithms in detection accuracy. After adding gaussian white noise, gaussian blurring and JPEG compression attack, our method still has satisfactory

detection performance on the CASIA dataset. Our method is also more accurate than RGB-N algorithm for locating tampered regions, especially on CMF.

However, because the CASIA dataset has undergone different degrees of post-processing, such as smoothing the boundary traces between the unforged region and the forged region of the tampered image, the results of our algorithm on the CASIA dataset are still not ideal. Therefore, manipulation detection of post-processed images becomes the key to future research.

Our study demonstrates that traditional image manipulation detection algorithms possess limited applicability and low detection accuracy. Our proposed image manipulation detection algorithm leveraged by Faster R-CNN model combined with edge detection exhibits higher detection performance than traditional image manipulation detection algorithms. Multiple features of tampered images could be fused to find more tampering clues and improve image manipulation detection performance.

Author Contributions: Conceptualization, X.W.; formal analysis, X.W.; funding acquisition, F.D.; investigation, S.S.; methodology, X.W.; project administration, F.D.; resources, J.Z.; software, X.W.; supervision, S.S.; validation, X.W.; visualization, X.W.; writing—original draft, X.W.; writing—review and editing, Y.W. and S.S.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. U1703261 and No. 61871258)

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, X.; Zhang, H.; Wang, C. A Robust Image Watermarking Technique Based on DWT, APDCBT, and SVD. *Symmetry* **2018**, *10*, 77.
2. Hou, X.; Min, L.; Yang, H. A Reversible Watermarking Scheme for Vector Maps Based on Multilevel Histogram Modification. *Symmetry* **2018**, *10*, 397.
3. Schneider, M.; Chang, S.F. A robust content based digital signature for image authentication. In Proceedings of the 3rd IEEE International Conference on Image Processing, Lausanne, Switzerland, 19 September 1996; Volume 3, pp. 227–230.
4. Ye, S.; Sun, Q.; Chang, E.C. Detecting Digital Image Forgeries by Measuring Inconsistencies of Blocking Artifact. In Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, Beijing, China, 2–5 July 2007; pp. 12–15.
5. Li, W.; Yuan, Y.; Yu, N. Passive detection of doctored JPEG image via block artifact grid extraction. *Signal Process.* **2009**, *89*, 1821–1829.
6. Zhu, N.; Shen, J.; Niu, X. Double JPEG Compression Detection Based on Noise-Free DCT Coefficients Mixture Histogram Model. *Symmetry* **2019**, *11*, 1119.
7. Bianchi, T.; Piva, A. Detection of Nonaligned Double JPEG Compression Based on Integer Periodicity Maps. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 842–848.
8. Mahdian, B.; Saic, S. Using noise inconsistencies for blind image forensics. *Image Vis. Comput.* **2009**, *27*, 1497–1503.
9. Lyu, S.; Pan, X.; Zhang, X. Exposing Region Splicing Forgeries with Blind Local Noise Estimation. *Int. J. Comput. Vis.* **2014**, *110*, 202–221.
10. Ferrara, P.; Bianchi, T.; Rosa, A.D.; Piva, A. Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1566–1577.
11. Rao, Y.; Ni, J. A deep learning approach to detection of splicing and copy-move forgeries in images. In Proceedings of the 2016 IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi, UAE, 4–7 December 2016; pp. 1–6.
12. Rota, P.; Sangineto, E.; Conotter, V.; Pramerdorfer, C. Bad teacher or unruly student: Can deep learning say something in Image Forensics analysis? In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancún, Mexico, 4–8 December 2016; pp. 2503–2508.
13. Bayar, B.; Stamm, M.C. Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2691–2706.

14. Bappy, J.H.; Roy-Chowdhury, A.K.; Bunk, J.; Nataraj, L.; Manjunath, B.S. Exploiting Spatial Structure for Localizing Manipulated Image Regions. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4980–4989.
15. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149.
16. Zhou, P.; Han, X.; Morariu, V.I.; Davis, L.S. Learning Rich Features for Image Manipulation Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 1053–1061.
17. Marr, D.; Hildreth, E.C. Theory of Edge Detection. *Proc. R. Soc. Lond. Ser. B Biol. Sci.* **1980**, *207*, 187–217.
18. Ulupinar, F.; Medioni, G.G. Refining edges detected by a LoG operator. *Comput. Vis. Graph. Image Process.* **1990**, *51*, 275–298.
19. Prewitt, J.M. Object enhancement and extraction. *Pict. Process. Psychopictorics* **1970**, *10*, 15–19.
20. Salloum, R.; Ren, Y.; Kuo, C.C.J. Image Splicing Localization Using a Multi-Task Fully Convolutional Network (MFCN). *J. Vis. Commun. Image Represent.* **2018**, *51*, 201–209.
21. Fridrich, J.; Kodovsky, J. Rich Models for Steganalysis of Digital Images. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 868–882.
22. Girshick, R.B. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
24. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the ICLR 2015: International Conference on Learning Representations 2015, San Diego, CA, USA, 7–9 May 2015.
25. Lin, T.Y.; RoyChowdhury, A.; Maji, S. Bilinear CNN Models for Fine-Grained Visual Recognition. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1449–1457.
26. Nist Nimble 2016 Datasets. 2016. Available online: https://mig.nist.gov/NC2017/Resources.html (accessed on 30 October 2018).
27. Hsu, Y.F.J.; Chang, S.F. Detecting Image Splicing using Geometry Invariants and Camera Characteristics Consistency. In Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 1 January 2006; pp. 549–552.
28. Dong, J.; Wang, W.; Tan, T. CASIA Image Tampering Detection Evaluation Database. In Proceedings of the 2013 IEEE China Summit and International Conference on Signal and Information Processing, Beijing, China, 6–10 July 2013; pp. 422–426.
29. Zampoglou, M.; Papadopoulos, S.; Kompatsiaris, Y. Large-scale evaluation of splicing localization algorithms for web images. *Multimed. Tools Appl.* **2017**, *76*, 4801–4834.