

Article

A Multiscale Fusion Lightweight Image-Splicing Tamper-Detection Model

Dan Zhao ^{1,2,3} and Xuedong Tian ^{1,2,3,*} ¹ School of Cyber Security and Computer, Hebei University, Baoding 071002, China² Hebei Machine Vision Engineering Research Center, Hebei University, Baoding 071002, China³ Institute of Intelligent Image and Document Information Processing, Hebei University, Baoding 071002, China

* Correspondence: xuedong_tian@126.com

Abstract: The easy availability and usability of photo-editing tools have increased the number of forgery attacks, primarily splicing attacks, thereby increasing cybercrimes. Because of an existing image-splicing tamper-detection algorithm based on deep learning with high model complexity and weak robustness, a multiscale fusion lightweight model for image-splicing tamper detection is proposed. For the above problems and to improve MobileNetV2, the structural block of the classification part of the original network structure was removed, the stride of the sixth largest structural block of the network was changed to 1, the dilated convolution was used instead of downsampling, and the features extracted from the second and third large structural blocks in the network were downsampled with maximal pooling; then, the constraint on the backbone network was increased by jumping connections. Combined with the pyramid pooling module, the acquired feature layers were divided into regions of different sizes for average pooling; then, all feature layers were fused. The experimental results show that it had a low number of parameters and required a small amount of computation, achieving 91.0% and 96.4% precision on CASIA and COLUMB, respectively, and 83.2% and 88.1% F-measure on CASIA and COLUMB, respectively.

Keywords: image-splicing tamper detection; multiscale fusion; lightweight network; MobileNetV2; pyramid pooling module



Citation: Zhao, D.; Tian, X. A Multiscale Fusion Lightweight Image-Splicing Tamper-Detection Model. *Electronics* **2022**, *11*, 2621. <https://doi.org/10.3390/electronics11162621>

Academic Editors: Leandros Maglaras, Helge Janicke and Mohamed Amine Ferrag

Received: 26 July 2022

Accepted: 19 August 2022

Published: 21 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the emergence of various image-editing programs, compositing images has become increasingly simple [1]. When this is used for fake news, fake propaganda, or fake evidence, it can have a bad effect. Therefore, it is urgent to identify and combat malicious image forgery.

Digital image-tampering methods generally include operations such as copying, splicing, and removing images. Image splicing refers to stitching a part of an image onto another image [2], so that the human eye cannot distinguish between true and false images, as shown in Figure 1.

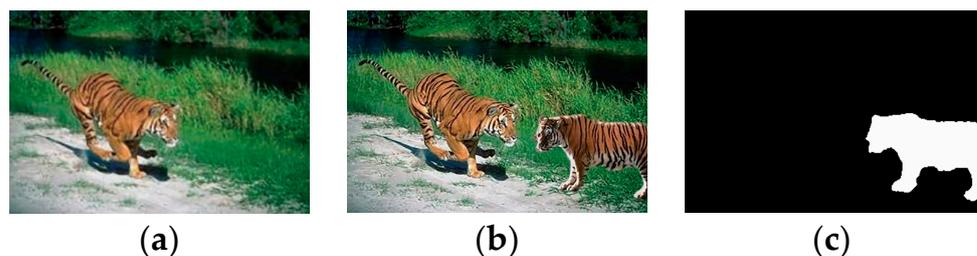


Figure 1. Example of image splicing tampering. (a) Original image; (b) splicing image; (c) ground truth.

Image-tampering detection methods can be divided into traditional feature-extraction-based detection methods and deep-learning-based detection methods.

Most of the traditional tampering detection algorithms focus on the statistical information and physical characteristics of the image itself, and use image feature extraction methods to detect the tampered regions. These features are manually selected, so they are also called manual feature extraction methods. These include looking for statistical anomalies related to color filter arrays (CFAs) [3–6], double JPEG compression [7–9], sensor noise [10,11], and the distribution inconsistencies of light sources [12,13]. Most of the traditional tampering detection technologies are premised on the existence of forgery and were only designed for a certain attribute of an image. The way of manually determining features leads to a final detection rate that is not very high and poor robustness.

In recent years, convolutional neural networks have achieved great success in the field of computer vision. Zhang et al. [14] tried to locate tampered regions with a CNN, but the detected regions could only be shown as inaccurate rough regions composed of some square white blocks. The authors in [15,16] attempted to use nonoverlapping image patches as input to a CNN; however, patches that were too large or too small failed to detect the tampered regions. Huh et al. [17] used automatically recorded photo EXIF metadata as a monitoring signal to train a model to determine whether the images were self-consistent; however, matching localization requires much computation and consumes many resources. Zhou et al. [18] performed feature extraction in the RGB-N dual-stream CNN framework, and then performed feature fusion, which improved the tamper detection accuracy. Wu et al. [19] proposed an end-to-end detection network that defined the splice localization problem as a local anomaly-detection problem. Remya and Wilsy [20] used a pretrained convolutional neural network to extract deep texture features from a rotation-invariant local binary pattern (RI-LBP) map of chroma images, and then trained a quadratic support vector machine (SVM), which is a classifier that improves the detection accuracy of fake images. El-Latif et al. [21] proposed an image-stitching detection algorithm based on deep learning and wavelet transform. Nath et al. [22] proposed a blind image-stitching detection technique that uses a deep convolutional residual network structure as the backbone, and then uses full connections to classify true and false images with high accuracy. Wang et al. [23] proposed an image-splicing tamper-detection method based on deep learning and an attention mechanism that could effectively improve the accuracy of image-stitching tampering detection and locate the tampered area. Jaiswal et al. [24] proposed a noise-inconsistency-based technique to detect and localize false regions in images with high detection accuracy, but it complicated the process. In [25], the authors proposed an end-to-end fully convolutional neural network containing RGB and DCT streams, with each stream considering multiple resolutions to handle various shapes and sizes of stitched objects.

Most of the above deep-learning networks have one thing in common: deep-learning-based algorithms have strong representation capabilities for complex data, which, to a certain extent, disposes of the dependence of traditional methods on hand-designed features. However, in the process of improving the accuracy of splicing detection, the complexity of the model and the equipment requirements increase.

To solve the above problems, this paper proposes a lightweight network, Mobile-Pspnet, based on MobileNetV2 [26] multiscale spatial information fusion; for JPEG images that may be spliced, a mask that can locate the parts of the image that may be tampered with is generated.

2. Related Work

2.1. Depthwise Separable Convolution

Depthwise separable convolution [27] was proposed by Google, and consists of depthwise and pointwise convolutions. The feature extraction method of depthwise separable convolution is different from that of traditional convolution, as shown in Figure 2.

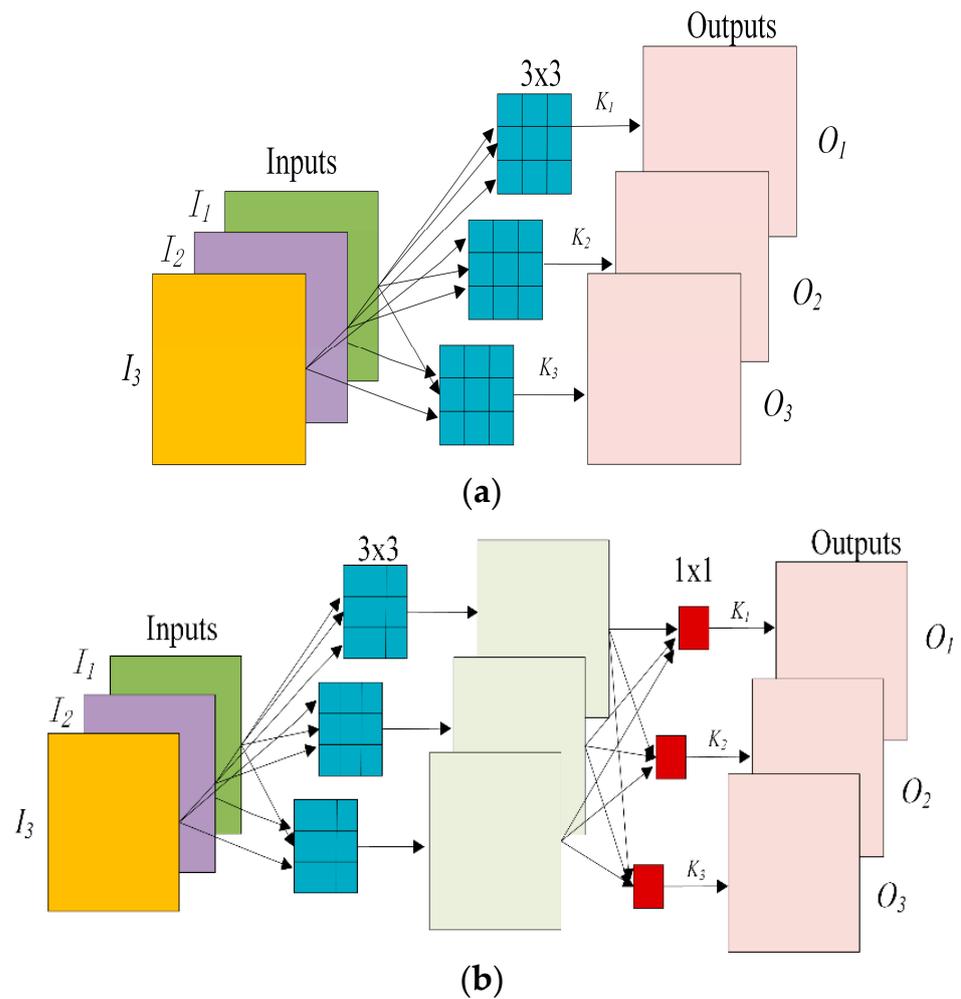


Figure 2. Contrast diagram of (a) traditional and (b) depthwise separable convolutions.

Suppose the dimensions of the input matrix are $N_F \times D_F \times D_F \times C_F$, the traditional convolutional kernel size is $N_K \times D_K \times D_K \times C_K$, and the two convolutional kernel sizes in the depthwise separable convolution are $C_F \times D_K \times D_K \times 1$, $N_F \times 1 \times 1 \times C_K$. The calculation amounts of traditional and depthwise separable convolutions are shown in Formulas (1) and (2), respectively.

$$F_1 = D_K \times D_K \times C_F \times N_K \times D_F \times D_F \tag{1}$$

$$F_2 = D_K \times D_K \times C_F \times D_F \times D_F + C_F \times N_K \times D_F \times D_F \tag{2}$$

When the size of convolutional kernel $D_K \times D_K \times C_K$ is $3 \times 3 \times C_K$, the computational complexity of depthwise separable convolution can be significantly reduced to that of standard convolution 1/8–1/9, as shown in Formula (3).

$$\frac{F_2}{F_1} = \frac{1}{D_K^2} + \frac{1}{N_K} \sim \frac{1}{D_K^2} \tag{3}$$

2.2. Lightweight MobileNetV2

After the depthwise separable convolution had been proposed, Google proposed MobileNetV2 [26]. When stride = 1, the architecture of MobileNetV2 is an inverted residual structure with a linear bottleneck. When stride = 2, because the sizes of input and output are different, no shortcut structure is added, and the network adopts depthwise (DW) convolution combined with pointwise (PW) convolution to extract features. The residual

structure that it introduces could significantly reduce the memory footprint required during inference while enhancing gradient propagation.

2.3. Pyramid Pooling Module

For segmentation tasks, contextual information and multiscale fusion are very effective for segmentation accuracy. The pyramid pooling structure [28] considers the target features under multiple receptive fields in parallel. It divides the acquired feature layers into regions of different sizes, and performs average pooling within each region to aggregate the context information of different regions to obtain different scales of average feature-layer pooling areas, thereby improving the ability to obtain global information.

3. Proposed Method

The Mobile-Pspnet network proposed in this paper is shown in Figure 3.

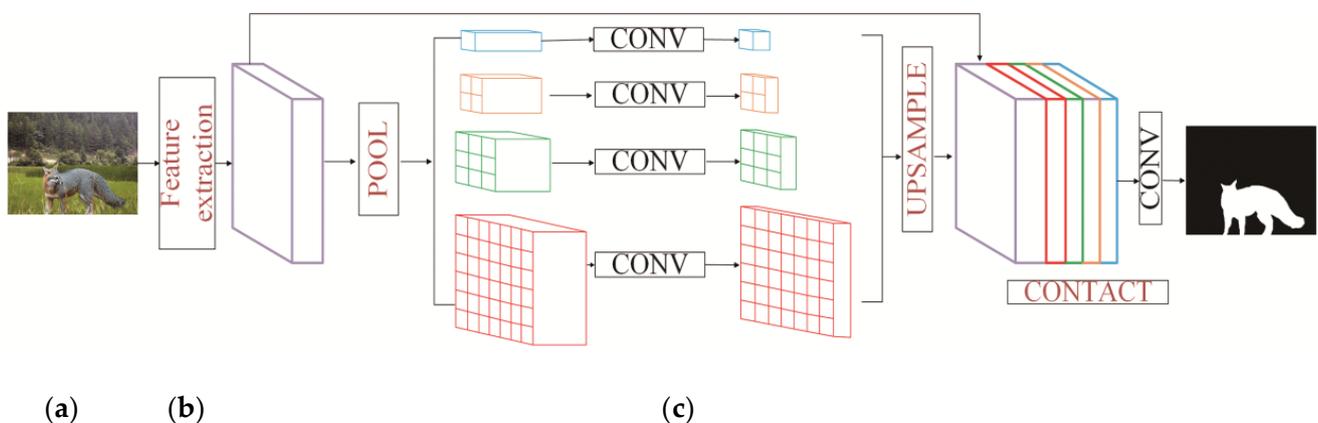


Figure 3. Overall structure of Mobile-Pspnet. (a) Input iamge; (b) Feature map; (c) Final prediction.

Mobile-Pspnet is composed of a feature extraction module and a pyramid pooling module [28]. The improved MobileNetV2 was used as the backbone network to reduce the number of parameters and operations as much as possible. The relative bottom-level features extracted by the second and third largest structural blocks were downsampled in a maximal pooling manner. The number of channels were adjusted and performed skip connections with the features obtained from the backbone network to enhance the richness of features, reduce information loss, and send the obtained features to the pyramid pooling module. Then, it was fused with the features extracted by the feature extraction module, and the obtained features were lastly classified at the pixel level to obtain pixel-level positioning.

3.1. Improving MobileNetV2

The original MobileNetV2 is suitable for image classification tasks, but not for segmentation tasks. Different from the structure of the original paper, we performed some changes on the MobileNetV2 structure used for feature extraction. The structural blocks of the classification part of the original network structure were removed, and the stride of the sixth largest building block of the network was changed to 1, the expansion coefficient of the last two largest structural blocks was set to 2, and dilated convolution was used instead of downsampling, which improved the receptive field while keeping the size of the feature map unchanged. The modified network parameters are compared in Table 1 below.

Table 1. Comparison of network parameters before and after modification.

Layer	Operation (Before)	Operation	Stride (Before)	Stride
1	Block	Block	1	1
2	Block	Block	2	2
3	Block	Block	2	2
4	Block	Block	2	2
5	Block	Block	1	1
6	Block	dilated block	2	Stride = 1, dilate rate = 2
7	Block	dilated block	1	Stride = 1, dilate rate = 2

3.2. Feature Extraction Module

The features extracted from the second and third largest structural blocks in the improved MobileNetV2 network structure were downsampled with the maximal pooling method with windows of 4×4 and 2×2 , respectively, and then combined with the features extracted by the backbone feature extraction network. The features were connected, which could strengthen the extraction of feature textures and reduce the influence of useless information. The structure of the specific feature extraction block module is shown in Figure 4.

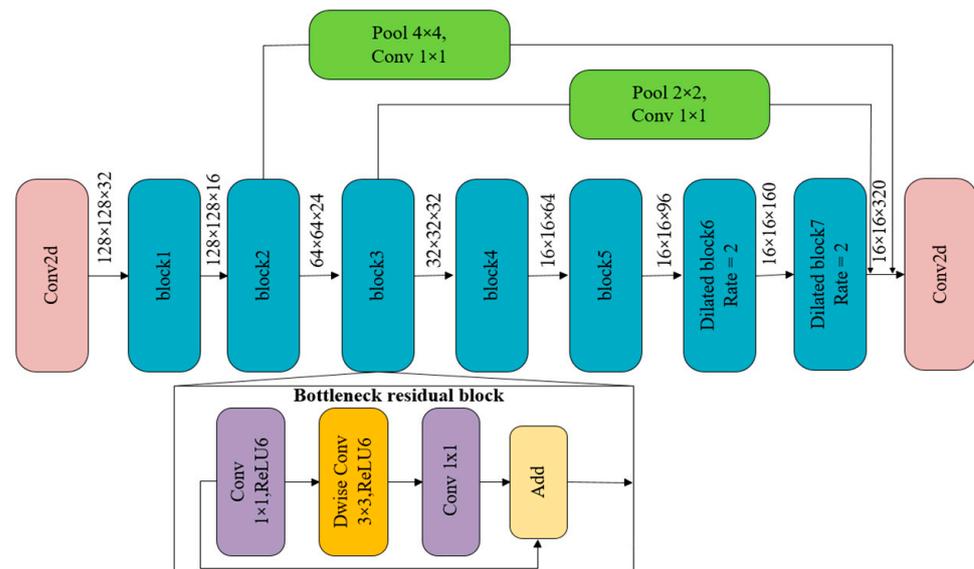


Figure 4. Structure diagram of the feature extraction module.

4. Experimental Results and Discussion

For better feature extraction, improved feature-extraction network MobileNetV2 was trained on the PASCAL VOC [29] dataset to obtain a pretrained model of the backbone network for more accurate and faster feature extraction in the training process. To better train the model, all input images were resized to 384×256 . The batch size used for network training was 32. Random quality factor compression, the addition of Gaussian noise, and image flipping were used for data augmentation. The initial learning rate was set to 0.001 and was dynamically reduced until it was 0, and the momentum was 0.9. Cross-entropy [30] was used as the loss function. All experiments were run on a computer with NVIDIA RTX 2080Ti GPU.

4.1. Dataset Introduction

The CASIA [31] and COLUMB [32] datasets were selected from four major public datasets for evaluation. Most of the images in CASIA have been compressed, and are the images obtained by stitching smaller objects to a certain part of the original image. COLUMB is a historical dataset used for tampering detection, and its stitched regions are

simple, large, and meaningless regions. The binary label images of CASIA used in this paper are from a third party [33]. In CASIA, 100 sets of images were randomly selected as the test set, and the rest were used as the training and validation sets. Since the COLUMB dataset only provides label images in RGB format, appropriate image processing was performed on it to obtain binary label images. Similarly, in COLUMB, 44 sets of data were randomly selected as test sets, and the rest were used as the training and validation sets. To better train Mobile-Pspnet, the image sizes in both the training and the test sets were adjusted to 384×256 . Then, data augmentation was achieved using random Gaussian noise addition, random quality factor compression, and random flipping, rendering the entire dataset five times larger. The distribution of datasets based on CASIA and COLUMB in the experiment is shown in Table 2.

Table 2. Experiments based on CASIA and COLUMB datasets.

Sets	Cases	Parameters	Range	Step	CASIA	COLUMB
Training Set	Augmented splicing	—	—	—	20,000	625
	Source image	—	—	—	4000	125
Validation Set	Plain splicing	—	—	—	1500	110
	Source image	—	—	—	300	11
Testing Set	Plain splicing	—	—	—	100	44
	Source image	—	—	—	100	44
	JPEG compression	Quality factor	50–90	10	500	220
	Noise corruption	Variance	0.002~0.01	0.002	500	220

4.2. Evaluation Index

The pixelwise authenticity discrimination of tampered images is a binary classification task that is similar to a segmentation task, that is, marking each pixel in the image as tampered or real. The evaluation metrics are the number of correctly detected tampered pixels (TP), the number of correctly detected nontampered pixels, and the number of falsely detected tampered pixels (FP). Precision, recall, and F-measure were used to evaluate the pixel-level performance of the proposed tamper detection method. Their definitions are shown in Formulas (4)–(6).

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6)$$

4.3. Comparative Approach

Three classical feature-extraction-based detection methods, CFA [4], DCT [7], NOI [3]; two semantic-segmentation-based methods: C2R-Net [17] and DFNet [16]; and three CNN-based methods, FCN [34], RRU [35], and MCNL-Net [36] were used for the comparative experiments. CFA [4] measures the presence of artifacts in the local horizontal direction and, on the basis of a new statistical model, derives the tampering probability for each image patch. DCT [7] is a quantized table of power spectrum estimation based on the histogram of DCT coefficients, which locates forgery regions by checking for inconsistent blocking artifacts. NOI [3] detects tampered regions by detecting various noise levels in the image. Since DFNet [16] uses 64×64 image patches as input, the detection effect on CASIA is general, so it is not shown in the experimental results.

4.4. Experimental Results

4.4.1. Ablation Experiment

To verify the simplicity and effectiveness of the combined model of MobileNetV2 and the pyramid pooling module, the backbone networks MobileNetV2 and Mobile-Pspnet were tested separately on the CASIA dataset, and the intersection over union (IoU) was selected as the evaluation index, which is defined as the ratio of the intersection and union of the true label of a certain type of sample and the predicted value, as shown in Formula (7). The parameters, calculations, and IoU values in the experimental results are shown in Table 3. The data in the table show that, although Mobile-Pspnet had a slightly higher number of parameters and computations on the CASIA dataset than MobileNetV2 did, the IoU value was increased by 6.63%. Therefore, we combined MobileNetV2 with the pyramid pooling module, which was valid.

$$IoU = \frac{TP}{TP + FP + FN} \quad (7)$$

Table 3. Performance of two backbone networks.

Methods	Para (MB)	GFLOPs	IoU (%)
MobileNetV2	2.41	1.68	70.12
Ours	2.53	1.72	76.75

4.4.2. Common Splicing-Detection Result

Table 4 shows the performance of the proposed method and selected comparative methods on benchmark datasets CASIA and COLUMB.

Table 4. Test results of common splicing forgery.

Methods	Precision	CASIA		F-Measure	COLUMB	
		Recall	F-Measure		Precision	Recall
CFA [4]	0.057	0.846	0.108	0.574	0.469	0.517
DCT [7]	0.349	0.871	0.498	0.365	0.633	0.463
NOI [3]	0.079	0.088	0.083	0.321	0.015	0.028
DF-Net [16]	-	-	-	0.528	0.468	0.496
C2R-Net [17]	0.417	0.424	0.42	0.576	0.097	0.166
FCN [34]	0.509	0.173	0.259	0.859	0.443	0.584
MCNL-Net [36]	0.909	0.828	0.866	0.839	0.715	0.772
RRU-Net [35]	0.848	0.834	0.841	0.961	0.873	0.915
Ours	0.910	0.801	0.832	0.964	0.852	0.881

Table 4 shows that, among the selected comparison algorithms, the proposed algorithm had precision values of 0.910 and 0.964 on the CASIA and COLUMB datasets, respectively. Although the recall of Mobile-Pspnet was a little worse than that of DCT, from a subjective perspective, DCT almost lost effectiveness. Although the recall and F-measure of Mobile-Pspnet were a little worse than those of RRU, Mobile-Pspnet had higher precision, which ranked first among all the testing methods.

Table 5 shows the algorithmic complexity comparison between the proposed method and several selected classical deep-learning methods. Since both DF-Net and C2R-Net methods take smaller image patches as input, for the sake of fairness, we did not count the computation amount of these two methods.

Table 5. Complexity of different algorithms.

Methods	Para (MB)	GFLOPs
DF-Net [16]	22.22	-
C2R-Net [17]	54.81	-
FCN [34]	95.01	10.20
MCNL-Net [36]	18.63	7.06
RRU-Net [35]	8.33	4.18
Ours	2.53	1.72

Table 5 shows that the parameter amount of the algorithm in this paper was only 2.53 MB, which was 2.3 times lower than that of the RRU-Net algorithm with the smallest parameter amount in the comparison algorithm, and the FCN algorithm with the highest parameter amount was 47 times that of our algorithm. The average parameter amount of the comparison algorithm was 39.8 MB, which was 15.7 times the number of parameters of our algorithm. The calculation amount of the algorithm in this paper was the least among the comparison algorithms, namely, 0.4, 0.24, and 0.16 times the calculation amounts of RRU-Net, MCNL-Net, and FCN, respectively. So, the model complexity of the Mobile-Pspnet was small.

In general, compared with the comparison algorithms, the proposed algorithm had fewer parameters and required less computation when the accuracy is high. From the test results, it is easy to draw the following conclusions:

- Mobile-Pspnet generally outperformed the traditional image splicing detection methods.
- Compared with the detection methods based on deep-learning comparison methods, the algorithm in this paper had higher accuracy and lower model complexity.

4.4.3. Comparative Experiment of Anti-Interference Detection

To evaluate the robustness of Mobile-Pspnet, the following studies were conducted on the CASIA-based test set:

- (1) The CASIA test set was compressed with different JPEG quality factors of 50, 60, 70, 80, and 90 for splicing detection. The results of F-measure, precision, and recall are shown in Figure 5a–c, respectively.
- (2) After adding different degrees of noise to the CASIA test set (variances were 0.02, 0.04, 0.06, 0.08, and 0.1), splicing detection was performed. Figure 6a–c show the F-measure, precision, and recall in the experimental results, respectively.

Under the influence of different quality factor compressions on the test set, the precision of the Mobile-Pspnet was the highest among the seven other methods. When the compression quality factor was lower than 90, the recall of Mobile-Pspnet was a little lower than that of CFA, but was the highest among the other comparison methods. Under the influence of different degrees of noise added to the test set, the F-measure and precision of Mobile-Pspnet were the highest among all comparison methods. Figures 5 and 6 show that the proposed method was under the influence of noise interference or a different quality factor compression, and the line changes of the three graphs were relatively gentle compared to those of the comparison methods. We can conclude that the detection effect of Mobile-Pspnet was relatively stable, and the proposed model had good robustness.

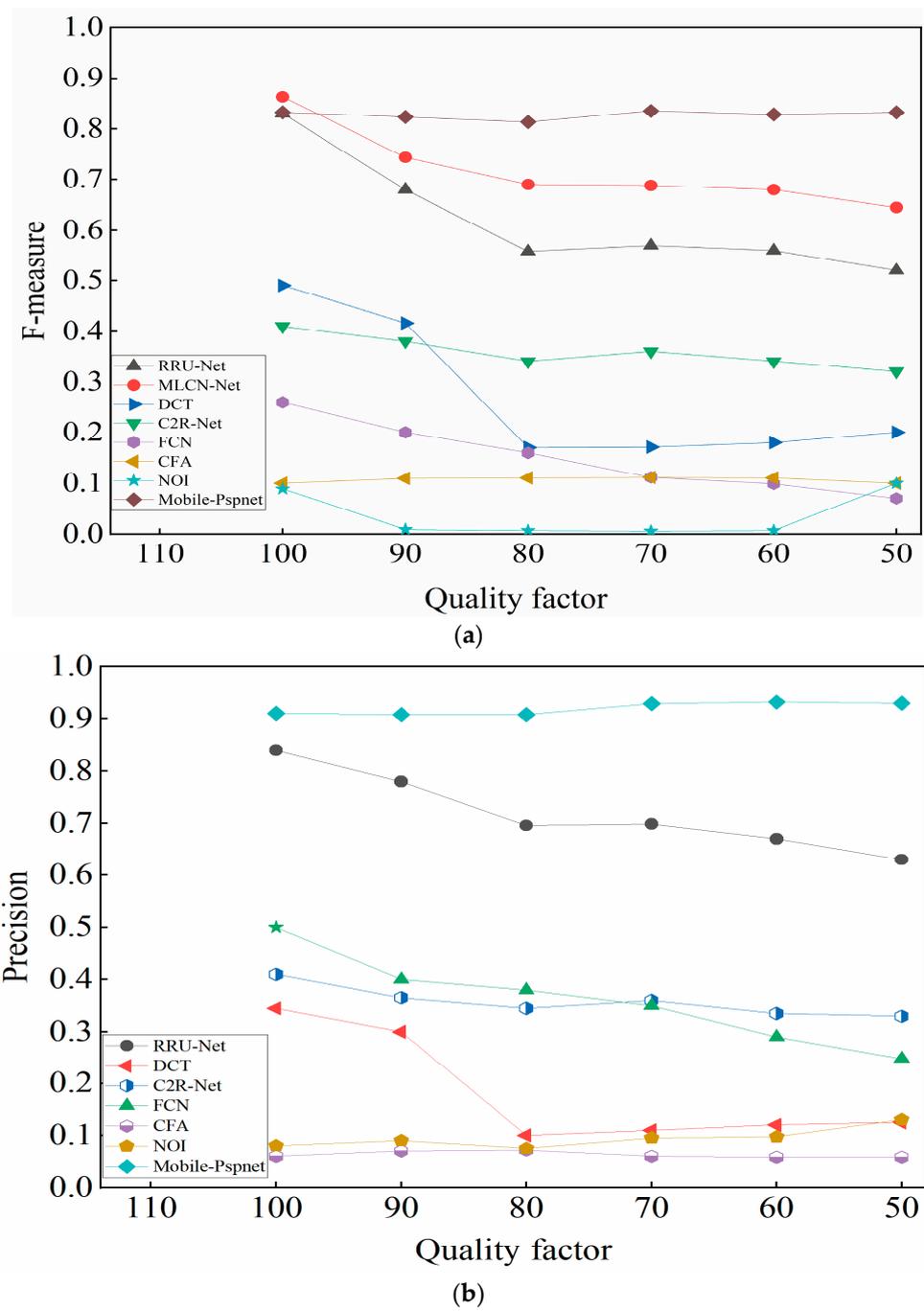


Figure 5. Cont.

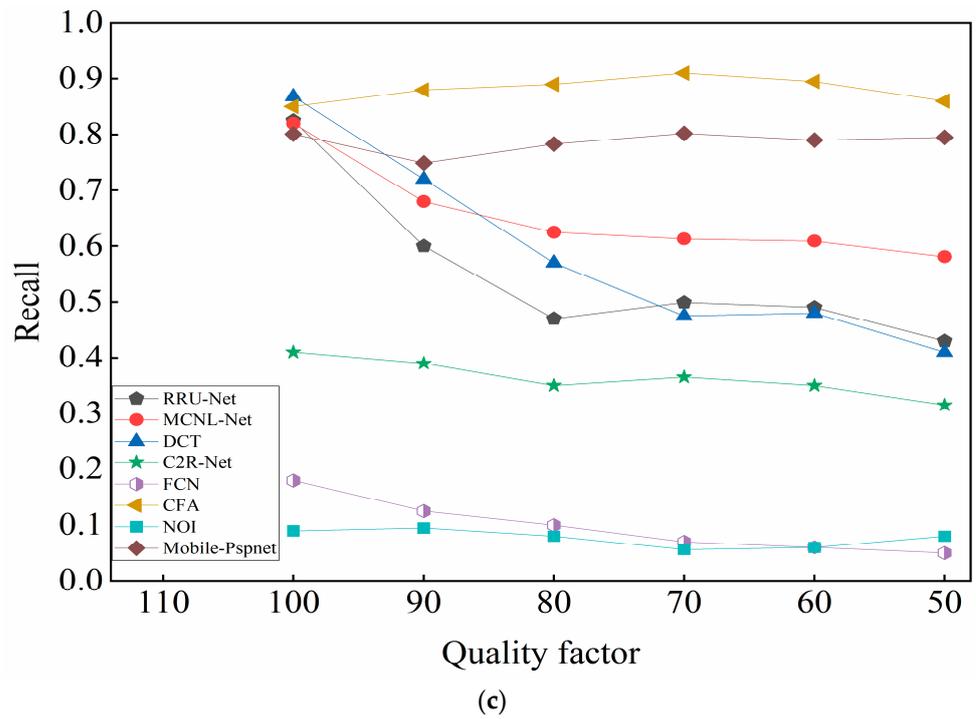


Figure 5. Detection results based on different quality factor compression of CASIA. (a) The influence of different image compression quality factors on F-measure; (b) The influence of different image compression quality factors on Precision; (c) The influence of different image compression quality factors on Recall.

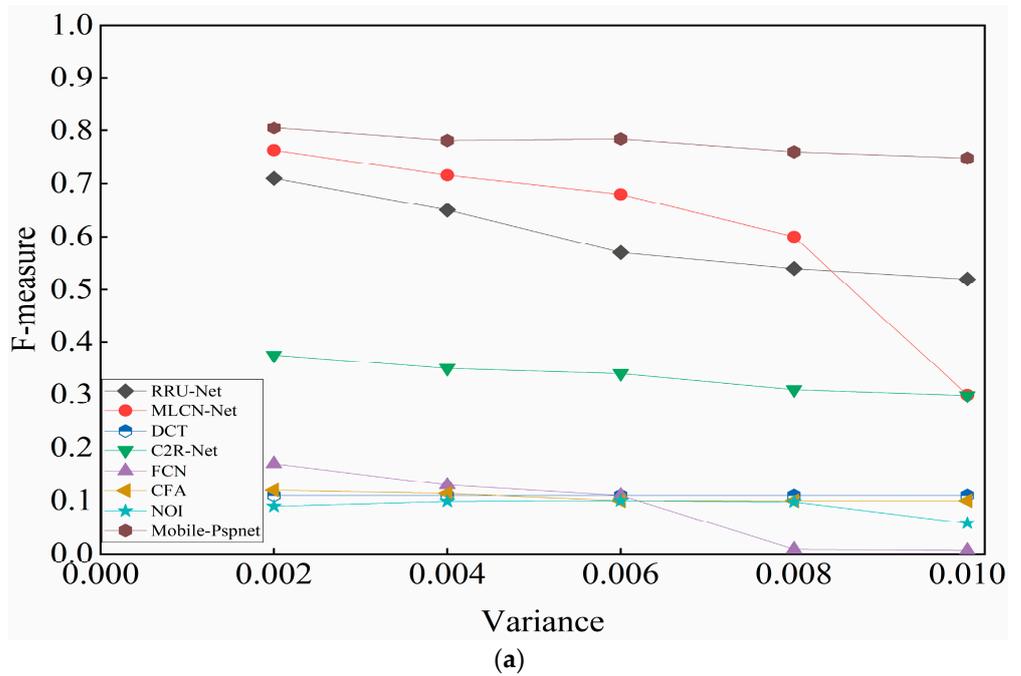


Figure 6. Cont.

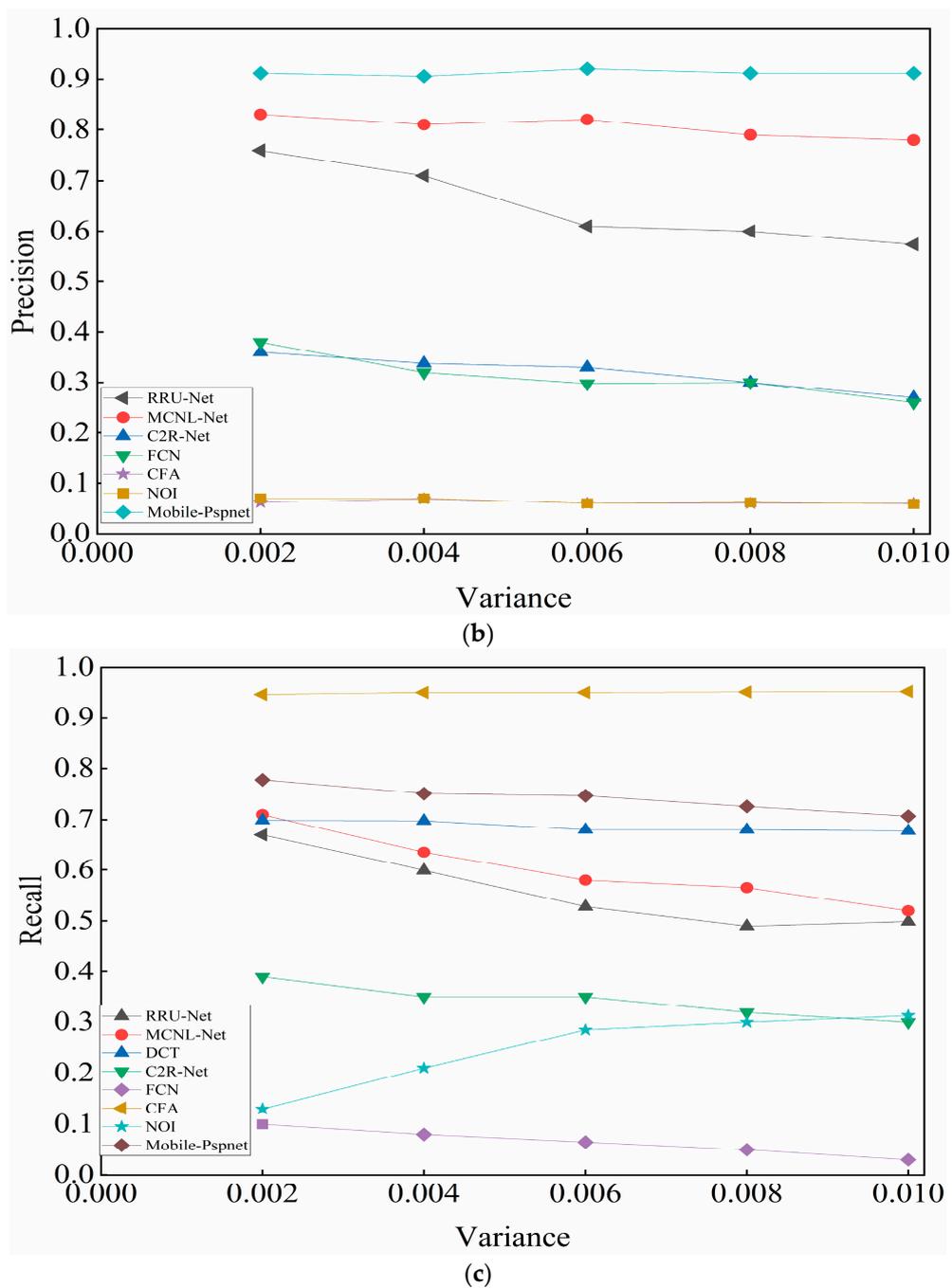


Figure 6. Detection results based on different noise levels of CASIA. (a) The influence of different image noise intensity on F-measure; (b) The influence of different image noise intensity on Precision; (c) The influence of different image noise intensity on Recall.

The partial detection results of the algorithm in this paper on the two public datasets of CASIA and COLUMB are shown in Figures 7 and 8.

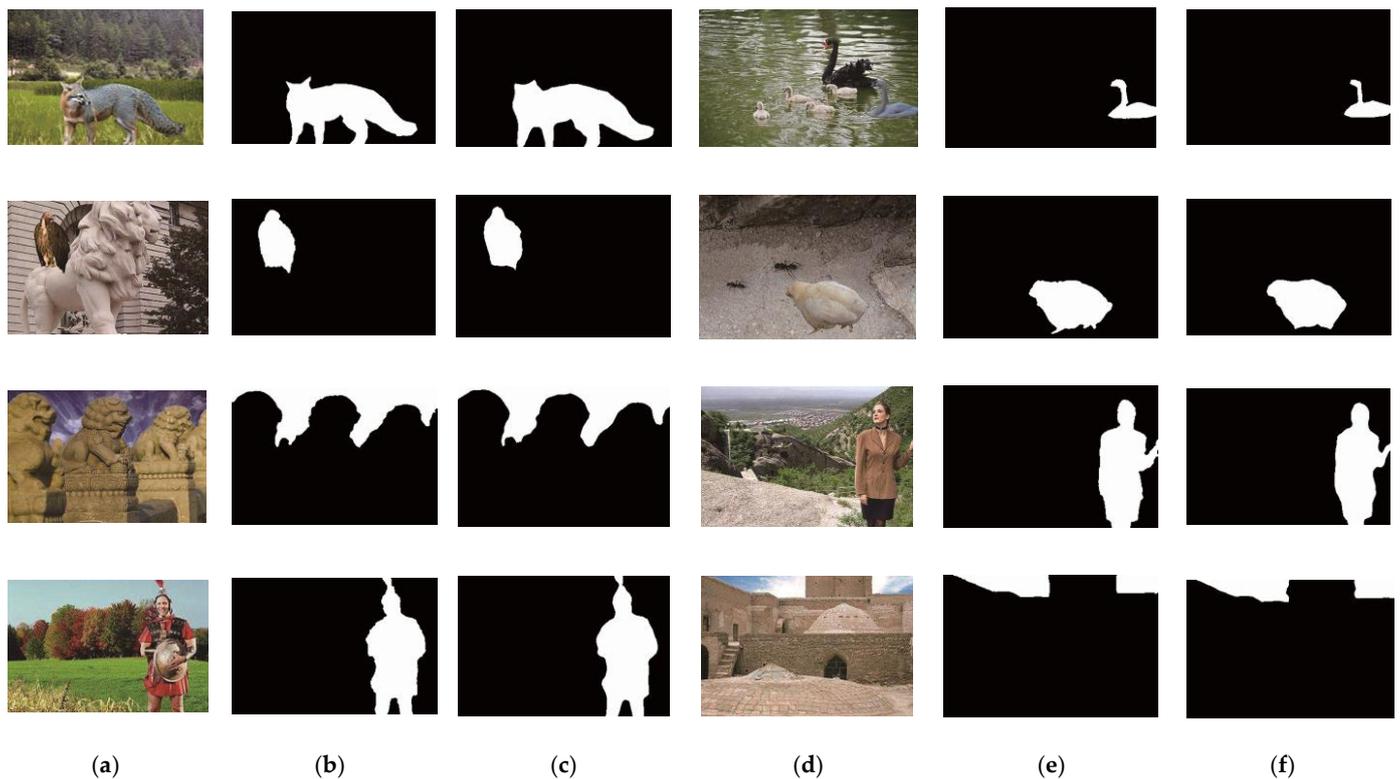


Figure 7. Detection results based on CASIA dataset. (a) Input image; (b) ground truth; (c) prediction; (d) input image; (e) ground truth; (f) prediction.

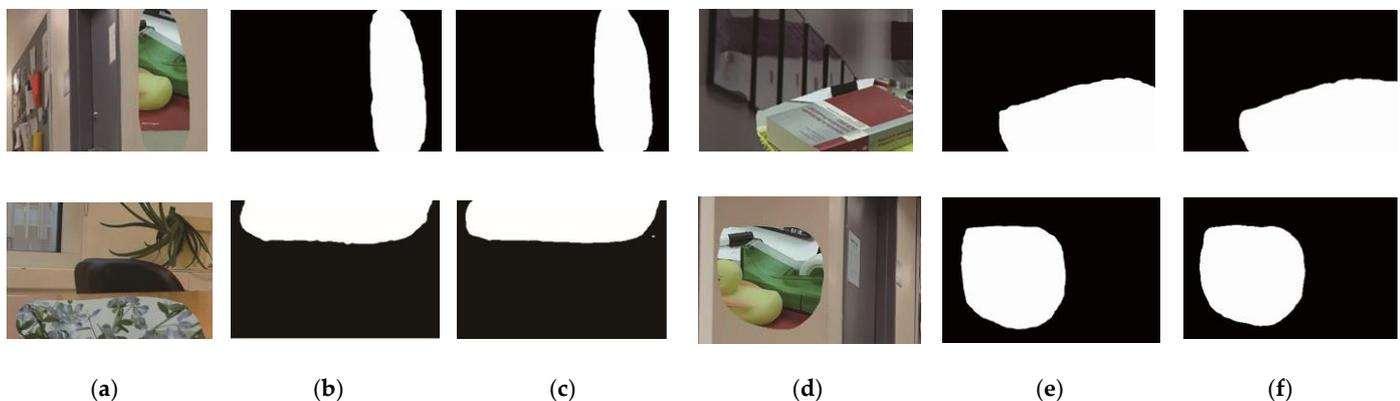


Figure 8. Detection results based on COLUMB dataset. (a) Input image; (b) ground truth; (c) prediction; (d) input image; (e) ground truth; (f) prediction.

5. Conclusions

We proposed a multiscale fusion lightweight image-splicing tamper-detection model, Mobile-Pspnet. This model uses MobilenetV2 as the backbone network after improving, and then increases the constraints on the backbone network using skip connections, strengthens the extraction of underlying features, and is combined with the pyramid pooling module. Multiple receptive fields were considered in parallel to improve the ability to obtain global information. We performed many comparative experiments with some classical image-splicing detection methods based on two public datasets. Experimental results show that Mobile-Pspnet outperformed most other detection methods. Furthermore, the proposed network is a lightweight segmentation network with a small number of parameters requiring a small amount of computation. In future work, under the condition of ensuring detection accuracy and low model complexity, the rapidity of

Mobile-Pspnet will be studied, so that it can be detected in a timely and efficient manner in video tamper detection.

Author Contributions: D.Z. substantially contributed to the conception of the work. D.Z. drafted the work. X.T. revised the work critically for important intellectual content. D.Z. substantially contributed to data analysis and interpretation. D.Z. collected the data in the experiments. All authors have read and agreed to the published version of the manuscript.

Funding: Natural Science Foundation of Hebei Province of China (grant number F2019201329).

Data Availability Statement: The data that support the findings of this study are available upon request.

Acknowledgments: The authors wish to express their gratitude to the anonymous reviewers and the associate editor for their rigorous comments during the review process. In addition, the authors would like to thank the instructor for guiding them in the experiment.

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

References

1. Tsai, Y.H.; Shen, X.; Lin, Z.; Sunkavalli, K.; Lu, X.; Yang, M.H. Deep image harmonization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3789–3797.
2. Verdoliva, L. Media forensics and deepfakes: An overview. *IEEE J. Sel. Top. Signal Process.* **2020**, *14*, 910–932. [[CrossRef](#)]
3. Mahdian, B.; Saic, S. Using noise inconsistencies for blind image forensics. *Image Vis. Comput.* **2009**, *27*, 1497–1503. [[CrossRef](#)]
4. Ferrara, P.; Bianchi, T.; De Rosa, A.; Piva, A. Image forgery localization via fine-grained analysis of CFA artifacts. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1566–1577. [[CrossRef](#)]
5. Wang, X.; Wang, Y.; Lei, J.; Li, B.; Wang, Q.; Xue, J. Coarse-to-fine-grained method for image splicing region detection. *Pattern Recognit.* **2022**, *122*, 108347. [[CrossRef](#)]
6. Singh, G.; Singh, K. Digital image forensic approach based on the second-order statistical analysis of CFA artifacts. *Forensic Sci. Int. Digit. Investig.* **2022**, *05*, 7. [[CrossRef](#)]
7. 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.
8. Bianchi, T.; Piva, A. Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1003–1017. [[CrossRef](#)]
9. Abecidan, R.; Itier, V.; Boulanger, J.; Bas, P. Unsupervised JPEG Domain Adaptation for Practical Digital Image Forensics. In Proceedings of the 2021 IEEE International Workshop on Information Forensics and Security (WIFS), Montpellier, France, 7–10 December 2021; pp. 1–6.
10. Chierchia, G.; Poggi, G.; Sansone, C.; Verdoliva, L. A Bayesian-MRF approach for PRNU-based image forgery detection. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 554–567. [[CrossRef](#)]
11. Flor, E.; Aygun, R.; Mercan, S.; Akkaya, K. PRNU-based Source Camera Identification for Multimedia Forensics. In Proceedings of the 2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI), Las Vegas, NV, USA, 10–12 August 2021; pp. 168–175.
12. De Carvalho, T.J.; Riess, C.; Angelopoulou, E.; Pedrini, H.; de Rezende Rocha, A. Exposing digital image forgeries by illumination color classification. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1182–1194. [[CrossRef](#)]
13. Wu, X.; Fang, Z. Image splicing detection using illuminant color inconsistency. In Proceedings of the 2011 Third International Conference on Multimedia Information Networking and Security, Shanghai, China, 4–6 November 2011; pp. 600–603.
14. Zhang, Y.; Goh, J.; Win, L.L.; Thing, V.L. Image Region Forgery Detection: A Deep Learning Approach. *SG-CRC* **2016**, *2016*, 1–11.
15. Liu, B.; Pun, C.M. Deep fusion network for splicing forgery localization. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
16. Wei, Y.; Bi, X.; Xiao, B. C2r net: The coarse to refined network for image forgery detection. In Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 1656–1659.
17. Huh, M.; Liu, A.; Owens, A.; Efros, A.A. Fighting fake news: Image splice detection via learned self-consistency. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 101–117.
18. Zhou, P.; Han, X.; Morariu, V.I.; Davis, L.S. Learning rich features for image manipulation detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1053–1061.
19. Wu, Y.; AbdAlmageed, W.; Natarajan, P. 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, Long Beach, CA, USA, 16–20 June 2019; pp. 9543–9552.
20. Remya Revi, K.; Wilscy, M. Image forgery detection using deep textural features from local binary pattern map. *J. Intell. Fuzzy Syst.* **2020**, *38*, 6391–6401. [[CrossRef](#)]

21. El-Latif, E.; Taha, A.; Zayed, H.H. A passive approach for detecting image splicing based on deep learning and wavelet transform. *Arab. J. Sci. Eng.* **2020**, *45*, 3379–3386. [[CrossRef](#)]
22. Nath, S.; Naskar, R. Automated image splicing detection using deep CNN-learned features and ANN-based classifier. *Signal Image Video Process.* **2021**, *15*, 1601–1608. [[CrossRef](#)]
23. Horváth, J.; Baireddy, S.; Hao, H.; Montserrat, D.M.; Delp, E.J. Manipulation Detection in Satellite Images Using Vision Transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 1032–1041.
24. Jaiswal, A.K.; Srivastava, R. Forensic image analysis using inconsistent noise pattern. *Pattern Anal. Appl.* **2021**, *24*, 655–667. [[CrossRef](#)]
25. Kwon, M.J.; Yu, I.J.; Nam, S.H.; Lee, H.K. CAT-net: Compression artifact tracing network for detection and localization of image splicing. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Online, 19–25 June 2021; pp. 375–384.
26. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 4510–4520.
27. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2022**, arXiv:1704.04861v1. [[CrossRef](#)]
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]
29. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [[CrossRef](#)]
30. Shore, J.; Johnson, R. Axiomatic derivation of the principle of maximum entropy and the principle of minimum crossentropy. *IEEE Trans. Inf. Theory* **1980**, *26*, 26–37. [[CrossRef](#)]
31. 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.
32. Hsu, Y.F.; 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, 9–12 July 2006; pp. 549–552.
33. Pham, N.T.; Lee, J.W.; Kwon, G.R.; Park, C.S. Hybrid image-retrieval method for image splicing validation. Symmetry Hybrid image-retrieval method for image splicing validation. *Symmetry* **2019**, *11*, 83. [[CrossRef](#)]
34. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
35. Bi, X.; Wei, Y.; Xiao, B.; Li, W.S. RRU-Net: The ringed residual U-Net for image splicing forgery detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–20 June 2019.
36. Wei, Y.; Wang, Z.; Xiao, B.; Liu, X.; Yan, Z.; Ma, J. Controlling neural learning network with multiple scales for image splicing forgery detection. *ACM Trans. Multimed. Comput. Commun. Appl.* **2020**, *16*, 1–22. [[CrossRef](#)]