Signal, Image and Video Processing
https://doi.org/10.1007/s11760-020-01636-0

ORIGINAL PAPER

Image splicing detection using mask-RCNN

Belal Ahmed[1] · **T. Aaron Gulliver**[1] · **Saif alZahir**[2]

Received: 13 June 2019 / Revised: 28 November 2019 / Accepted: 6 January 2020

Abstract
Digital images have become a dominant source of information and means of communication in our society. However, they can easily be altered using readily available image editing tools. In this paper, we propose a new blind image forgery detection technique which employs a new backbone architecture for deep learning which is called ResNet-conv. ResNet-conv is obtained by replacing the feature pyramid network in ResNet-FPN with a set of convolutional layers. This new backbone is used to generate the initial feature map which is then to train the Mask-RCNN to generate masks for spliced regions in forged images. The proposed network is specifically designed to learn discriminative artifacts from tampered regions. Two different ResNet architectures are considered, namely ResNet-50 and ResNet-101. The ImageNet, He_normal, and Xavier_normal initialization techniques are employed and compared based on convergence. To train a robust model for this architecture, several post-processing techniques are applied to the input images. The proposed network is trained and evaluated using a computer-generated image splicing dataset and found to be more efficient than other techniques.

Keywords Image forgery · Image splicing · Mask-RCNN · Imagenet

1 Introduction

The advent of the internet along with the plethora of social media and other applications has made digital images a dominant source of information. They are used to document evidence for legal purposes as well as in medical imaging for diagnostic purposes, sports, and many other fields [1–3]. While digital imaging provides numerous possibilities for creation, it can also be used to produce forged documents. Image forgery is almost as old as photography itself and started as early as 1865 when photographer Mathew Brady added General Francis P. Blair to an original photograph to make it appear that he was present as shown in Fig. 1.

Image editing tools have become commonplace, and even sophisticated software is available free of charge. This allows anyone with a computer to easily manipulate an image. As a consequence, forged images can be found everywhere, on the covers of magazines, in courtrooms, and on the internet. Therefore, a robust and effective method for image forgery detection is of great importance in digital image forensics.

Image forgery detection techniques can be classified into two categories: active and passive. In active methods, certain information is embedded into the image during creation such as a watermark or signature. With these methods, image tampering is detected by analyzing the watermark or signature. Although watermarking can protect an image from theft, its application is limited because human intervention is required to recover the original watermark-free image. Conversely, passive techniques do not require manual processing [4]. Forgery changes the image feature statistics and introduces artifacts resulting in inconsistencies. Most passive techniques use these inconsistencies to identify forged images.

Image tampering can be done in several ways such as image splicing, retouching, and copy-move forgery [5]. Copy-move forgery refers to copying a part of an image and pasting it into the same image to conceal or change information. Several methods have been introduced to detect and localize copy-move forgeries [6]. Image retouching is used to change the appearance of a subject in an image [7]. In image splicing, part of an image is copied and pasted into another image to hide or add information. Image splicing is

✉ Belal Ahmed
bahmed@uvic.ca

[1] Department of Electrical and Computer Engineering, University of Victoria, Victoria, Canada

[2] Department of Computer Science and Engineering, University of Alaska Anchorage, Anchorage, USA

Published online: 17 January 2020

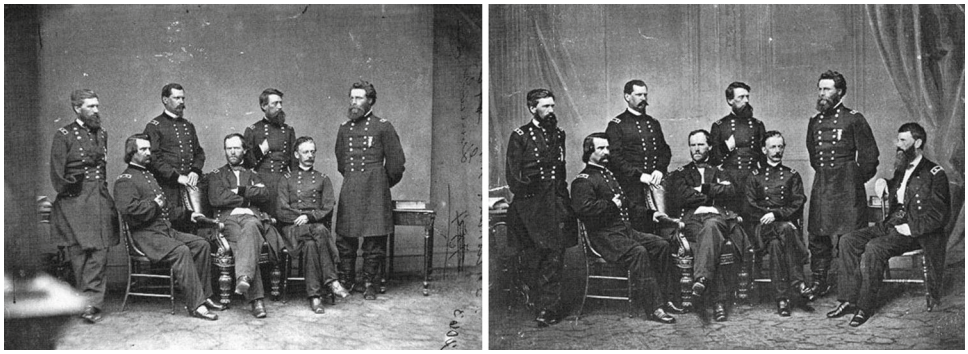

Fig. 1 An example of image splicing

widely used to create forgeries in images. Several approaches have been proposed to detect image splicing based on the abnormal transients at splicing boundaries. A method for image splicing detection based on a natural image model was introduced in [8]. This model uses statistical features extracted from the image and 2D arrays generated by applying a multi-size block discrete cosine transform (MBDCT) to the image. The statistical features include moments of characteristic functions of wavelet subbands and Markov transition probability matrices. In [9], the method in [8] was improved by capturing intra-block and inter-block correlations using DCT coefficients. The original Markov model obtained using a discrete wavelet transform (DWT) was used to extract additional features. Then, the cross-domain features were used to train a support vector machine (SVM) classifier.

In [10], grey-level run length matrix (GLRLM) texture features for forged and original images were determined. Then, statistical features extracted from the GLRLM were used to train an SVM for classification. In [11], an approach based on statistical features obtained from the run length and image edges was proposed. The method in [11] was improved in [12] by using a detection algorithm based on approximate run lengths. This improved the detection accuracy from 69% to 75% in less time.

Deep neural networks such as deep belief network [13], deep autoencoder [14], and convolutional neural network (CNN) [15] have recently been used to extract useful, high-level structure representations. This allows deep learning networks to generalize well across a wide variety of tasks such as image classification [16], speech recognition [17], camera model identification [18,19], and image and video manipulation detection [20–23]. A CNN model was introduced in [20] to detect image splicing by extracting dense features from image patches. These features are concatenated, and then a pooling operation (max or mean) is applied on each patch feature. Then, an SVM classifier is trained on these features for classification. In [24], a two-stage deep learning approach was introduced to detect tampering in images. First, a stacked autoencoder model is trained on the wavelet features of the images to learn complex features for each image patch. Then, the contextual information of each patch is integrated and used for detection.

A good detection method should consider low-resolution images resulting from compression or resizing. Thus, in [25], a shallow CNN (SCNN) was trained to distinguish the boundaries of forged regions from the original edges in low-resolution images by discriminating changes in chroma and saturation. This was done by first converting the image from RGB space to YCrCb space. Then, only the CrCb channels were used in the convolutional layers to exclude the illumination information. A different approach to copy-move forgery detection (CMFD) based on a CNN was proposed in [26]. In this method, a pre-trained CNN model was fine tuned using 3000 forged images from the ImageNet database. These images were generated by randomly moving a rectangular block from the upper left corner of the images to the center.

In our research, we train a supervised Mask-RCNN to learn the hierarchical features resulting from forgery operations such as image splicing. To generate the initial feature map, a new ResNet architecture called ResNet-conv is introduced. This is obtained by replacing the feature pyramid network FPN in ResNet-FPN [27] with a set of new convolutional layers. A comparison is made between ResNet-FPN and ResNet-conv in terms of the convergence speed. Different ResNet architectures have been tested and compared including ResNet-50 and ResNet-101 [28]. For faster convergence, a transfer learning strategy is used to initialize the proposed network. Different initialization techniques have been developed and compared such as ImageNet [29], Xavier_normal [30], and He_normal [31]. Our method has a higher efficiency than these techniques.

The remainder of this paper is organized as follows. Section 2 briefly discusses the framework of Mask-RCNN, Sect. 3 presents the algorithm used to generate the dataset, the experimental results, comparisons, and analysis. Finally, the conclusions are drawn in Sect. 4.

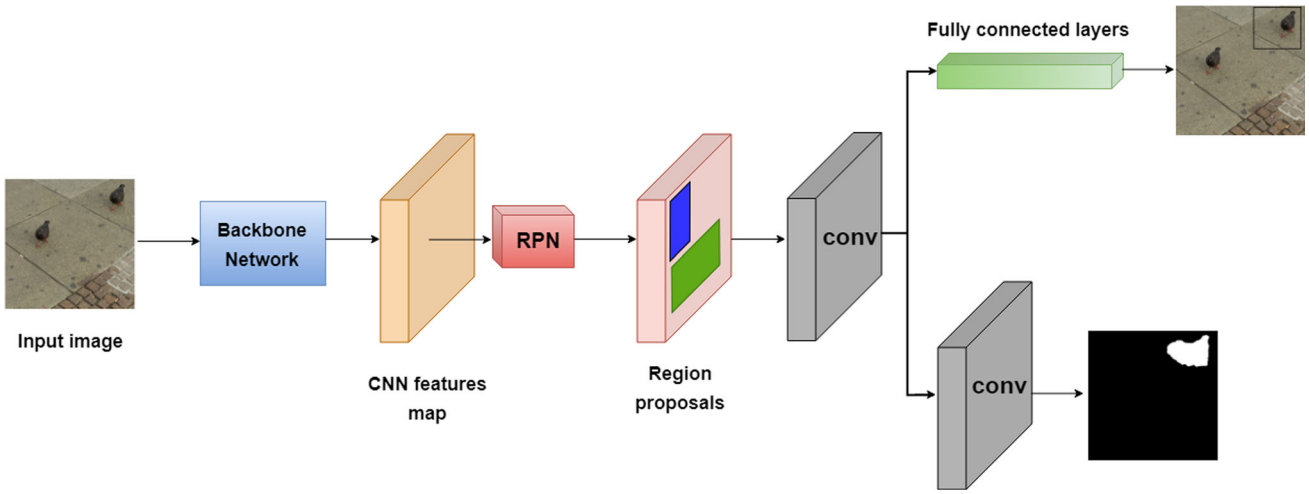

Fig. 2 The Mask-RCNN framework [32]

2 Mask-RCNN

A CNN consists of several convolutional and pooling layers and ends with one or more fully connected layers. Each convolutional layer consists of three steps, convolution, nonlinear activation, and pooling. After each convolutional layer, a feature map is generated and passed to the next layer. A convolutional layer can be represented by [20]

$$F^n(X) = \text{pooling}\big(f^n\big(F^{n-1}(X) * W^n + B^n\big)\big), \tag{1}$$

where $F^n(X)$ is the feature map for layer n of the convolution with kernel (filter) and bias given by W^n and B^n, respectively, and $*$ denotes convolution.

The mask regional convolutional neural network (Mask-RCNN) model was developed in [32] for semantic segmentation, object localization, and object instance segmentation. Mask-RCNN outperformed all existing single-model entries on every task in the 2016 COCO challenge including large-scale object detection, segmentation, and captioning dataset [33]. Mask-RCNN consists of two stages. The first stage, called region proposal network (RPN), scans the initial feature maps and generates region proposals or regions of interest (RoI) which is the same process employed by the faster-RCNN model [34]. In the second stage, an operation known as RoI-pooling [35] is applied to each RoI to downsample the feature map by using a nearest neighbor approach. This process selects important features from the feature map. RoI-pooling can result in misalignment between the RoI and the extracted features, so RoI-align is applied to each RoI to create more accurate RoIs. In RoI-align, the value of each sample point is calculated using bilinear interpolation from the nearby grid points on the feature map. In addition to predicting the class and bounding boxes for each object, Mask-RCNN also generates a binary mask for each RoI using a fully convolutional network (FCN). Figure 2 shows the framework of the Mask-RCNN network.

Both stages of the Mask-RCNN are connected to the backbone structure. The backbone is another deep neural network that is used to create the initial feature map. In principle, the backbone network could be any CNN pre-trained on an image dataset such as ResNet [28]. ResNet is an artificial neural network (ANN) that is based on residual learning. In residual learning, a network is trained by feeding the output of an initial layer to advanced layers to share the earlier residuals [28].

3 Proposed method

With the advent of new techniques, a forged image can easily be processed to make it more difficult for a human to detect the forgery [36]. In this section, the proposed method is presented and experiments are conducted to evaluate the effectiveness in detecting image forgery. To create a robust model, it is trained on a large dataset that contains different examples of forgery.

3.1 Image dataset

In order to generate a robust deep learning model, a sufficiently large dataset is required that contains different kinds of forgery. Currently available image forgery datasets are not large and do not cover a wide range of forgeries. For this reason, a computer-generated dataset is used to train the proposed model.

Computer-generated forged images have been created using the COCO dataset [33] and a set of random objects with transparent backgrounds [37]. This dataset consists of 80 classes, 80,000 training images, and 40,000 validation

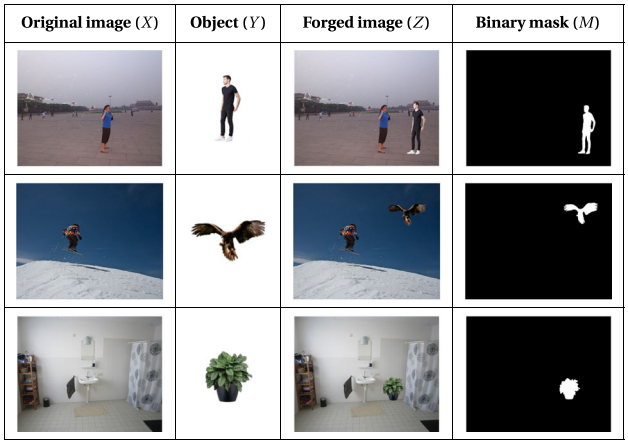

Fig. 3 Examples of forged image generation

images. The images have dimensions 480 × 640 pixels. To create image splicing, an object Y was chosen randomly and pasted into a random image X from the COCO dataset to create the spliced image Z which can be represented by the equation: $X + Y \Longrightarrow Z, M$. The COCO dataset was originally developed for detecting different types of objects such as cars, people, and vegetables. However, the proposed network is specifically designed to detect forged regions in images. Thus, the original COCO dataset labels are not used for training. Instead, a new binary mask M is generated for each spliced image with the forged region shown in white and the original region in black.

Figure 3 gives three examples of forged image generation using image splicing. A subset of 21,000 images was selected randomly from the 80,000 training images to create forged images, and an equal number of original images was selected for a total of 42,000 training images. Validation images were created similarly by selecting 905 original images and an equal number for forged images from the validation images in the dataset for a total of 1810 images. Validation images were selected randomly and are different from the images chosen for training. This ensures the model is evaluated on images not used during training for accurate evaluation. To complete the dataset, binary masks were created for each image with the forged pixels labeled by ones and the others labeled by zeros.

A forged image can be post-processed to alter any artifacts resulting from the forgery process. To create a robust model that can detect forgeries in post-processed images, image augmentation techniques were used on the computer-generated dataset. These techniques are rotation, shift, shear, and zoom. In rotation, the image is rotated with an angle chosen randomly between 0° and 360°. In shifting, a spatial shift is applied to the image with a width and height shift chosen randomly between 0 and 0.4. After shifting an image, it is cropped to its original dimensions. In shearing, a random transformation intensity between 40° and −40° is used to create a shear matrix. Then, this matrix is applied to the image using an affine transformation which results in the upper part of the image shifted to the right and the lower part to the left. In zooming, a zoom is applied to the image by randomly selecting two zoom values from the range [1, 10] for the image width and height. These values are used to create the zoom matrix which is applied to the image using an affine transformation. After zooming, the image is cropped to its original dimensions. Each of the four augmentation techniques is applied to an image during the training process with probability 0.50. Thus, an image has no augmentation or all four techniques applied with probability 0.0625. Adding augmented images expands the image dataset which improves the generalization ability and helps prevent overfitting.

3.2 Implementation

A Mask-RCNN model is used with the ResNet model to extract the initial feature map. This is based on an existing implementation by Matterport Inc. released under MIT License [38]. Stochastic gradient descent (SGD) optimization is used to optimize the proposed model with a momentum of 0.99 and a weight decay of 0.001. Using SGD with momentum helps accelerate the gradient vectors in the correct direction, thus leading to faster convergence [39]. Further, a small weight decay is multiplied by the weights after each update iteration to prevent the weights from growing too large. An initial learning rate of 0.01 is used, and this is reduced by 10% if the validation loss does not decrease for three consecutive epochs where an epoch denotes an iteration over all training examples. A reducing learning rate helps fine tuning the model to reach its local minimum. The proposed method was implemented using Keras [40] and evaluated on an NVIDIA GeForce GTX 1080 Ti GPU with a memory bandwidth of 11 Gbps.

3.2.1 Initialization

The initialization plays an important role in the convergence speed of a neural network. A good initialization strategy can reduce the feasible parameter space and help the network learn robust features related to the tampering operations rather than complex image content. Initialization can be done using random weights. Examples of random weight initializations are Xavier_normal [30], He_normal [31], Random_normal [41], and Random_uniform [42]. In Xavier_normal, the weights of the network are initialized from a distribution with zero mean and variance

$$\sigma^2 = 1/N_{\text{in}}, \tag{2}$$

where N_{in} is the number of incoming neurons. He_normal has a similar variance given by

$$\sigma^2 = 2/N_{\text{in}}. \tag{3}$$

Initialization can also be done using the weights of a network trained on a large dataset. In this case, the weights of a network are used to initialize another network to perform a different task. The ImageNet dataset has been used to pre-train networks such as ResNet [29]. This is because ImageNet contains over 14 million images which belong to more than 20,000 classes. Hence, networks pre-trained on ImageNet can learn a wide variety of features and so can be a good backbone.

Figure 4 presents the network convergence with initialization using a ResNet network pre-trained on ImageNet [29], Xavier_normal [30], and He_normal [31]. The training and validation losses are defined as $L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}}$ where L_{cls} is the classification loss, L_{box} is the bounding box loss, and L_{mask} is the mask loss [28]. Classification loss is calculated using sparse categorical cross-entropy, bounding box loss is calculated using smooth L_1 loss, and mask loss is calculated using binary cross-entropy [28]. These results show that the ResNet network pre-trained on the ImageNet dataset outperforms the other techniques with a difference of about 0.1 in both training loss and validation loss. Xavier_normal, and He_normal converge to the same minimum value which is about 0.2. The convergence speed is the same for both training and validation for all three methods which indicates that the model can achieve the same accuracy on new examples and overfitting did not occur.

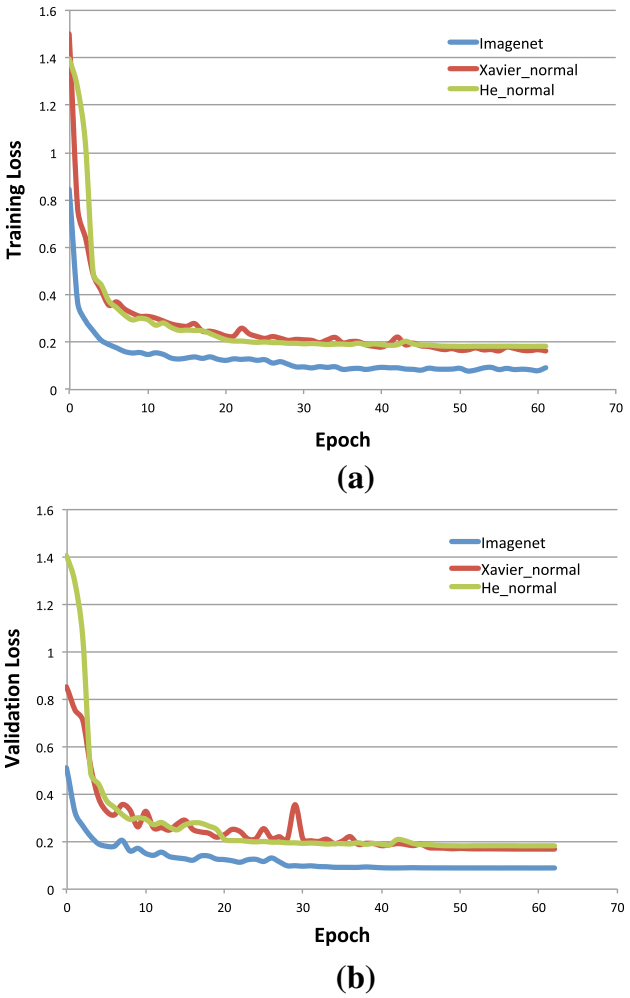

Fig. 4 Convergence performance for three different initialization methods, **a** training and **b** validation

3.2.2 Backbone

Convergence speed can also be improved based on the number of layers in the backbone. The number of layers depends on how many features exist in the dataset and that the network needs to be trained on. For this reason, two different ResNet architectures were considered which are ResNet-50 and ResNet-101 [28]. ResNet-50 consists of five stages. Stage 1 is the input stage which consists of one convolutional layer followed by batch normalization and activation functions to generate the initial feature map. Stages 2 to 5 have convolutional blocks and identity blocks. Both blocks consist of convolutional layers followed by batch normalization and activation functions. The convolutional blocks have an extra bridge to add residuals learned in the input layer to the output layer. The ResNet-101 architecture is the same as ResNet-50 except that it has 22 convolutional blocks in stage 4 compared to only 5 in ResNet-50. In total, ResNet-50 has 50 layers, while ResNet-101 has 101 layers.

To study the performance of the proposed model with ResNet-50 and ResNet-101 backbones, the convergence is compared. Both models were initialized with ImageNet weights [29] and trained for an equal number of epochs. All layer weights have been used to initialize the proposed model except the output layer. This is because the ImageNet weights are for 1000 output categories, while our network output has only two categories: original and forged. Figure 5 shows the training loss and validation loss versus the number of epochs. Although ResNet-101 contains twice the number of convolutional layers as ResNet-50, the training loss with ResNet-50 converges to a lower minimum. The difference in validation loss is even greater. This is because the number of layers in ResNet-50 is sufficient to learn the features in the dataset. On the other hand, ResNet-101 has extra layers that cause overfitting, resulting in an increase in the validation loss. Both models were tuned by reducing the learning rate by 0.005 if the validation loss did not improve for five consecutive epochs.

ResNet-FPN has been shown to improve both the accuracy and speed of some tasks [38]. A feature pyramid network

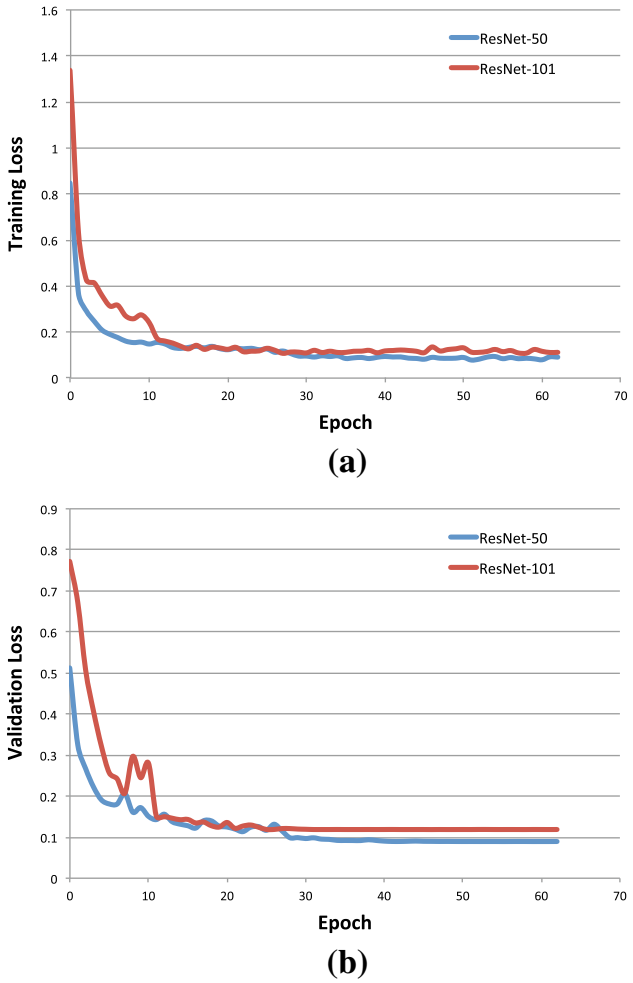

Fig. 5 Convergence performance for ResNet-50 and ResNet-101, **a** training and **b** validation

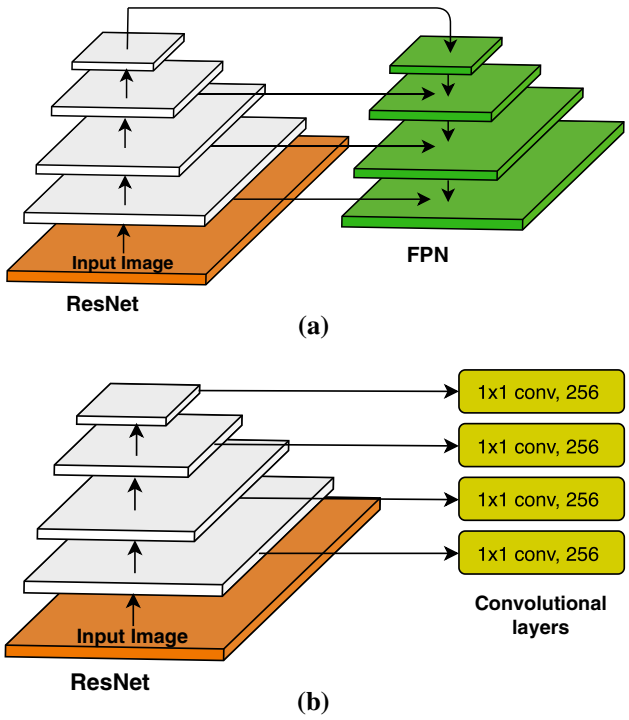

Fig. 6 The backbone architectures for **a** ResNet-FPN [27] and **b** ResNet-conv

(FPN) is a feature extractor designed in a pyramid for the purpose of generating multi-scale feature maps. FPN requires its own backbone network in order to create the feature pyramid.

To detect image forgeries, the proposed model needs to learn features based on the artifacts resulting from forgeries. Adding more layers to the backbone network may increase the convergence speed, but can also lead to overfitting. To choose a backbone suitable for our problem, ResNet-FPN is compared with a simplified version of ResNet called ResNet-conv. In ResNet-conv, the FPN network is replaced by four convolutional layers each with 256 filters, one layer for each of the four feature maps created by ResNet. Figure 6 shows the ResNet-FPN and proposed ResNet (ResNet-conv) structures.

Figure 7 presents the convergence performance with ResNet-FPN and ResNet-conv. For a fair comparison, both networks were trained for the same number of epochs. Although ResNet-FPN has lower validation and training losses than ResNet-conv up to epoch 30, the losses subsequently converge to the same minimum for both networks. Thus, adding FPN to ResNet does not improve the training which indicates that the feature maps created by ResNet have sufficient features to differentiate between the original and forged regions in an image. This is because the features in a forged region are related to sharp edges, color consistency with surrounding pixels, and differences in contrast and brightness which are basic features that can be learned in backbone training. FPN is a very complex network that may be necessary for more complex problems such as instance segmentation of objects [38]. In summary, the results presented here show that an FPN is not required.

Figure 8 presents the receiver operating characteristic (ROC) curve for the proposed network. This illustrates the ability of the network to discriminate between original and forged regions. It was created by plotting the true positive ratio (TPR) against the false positive ratio (FPR) at different threshold values. TPR and FPR are given by

$$\mathrm{TPR} = \frac{T_\mathrm{P}}{T_\mathrm{P} + F_\mathrm{N}}, \tag{4}$$

$$\mathrm{FPR} = \frac{F_\mathrm{P}}{T_\mathrm{N} + F_\mathrm{P}}, \tag{5}$$

where T_P is true positive which represents the number of pixels that are correctly detected as forged, F_N is false negative which represents the number of pixels that are falsely detected as original, and T_N is true negative which represents

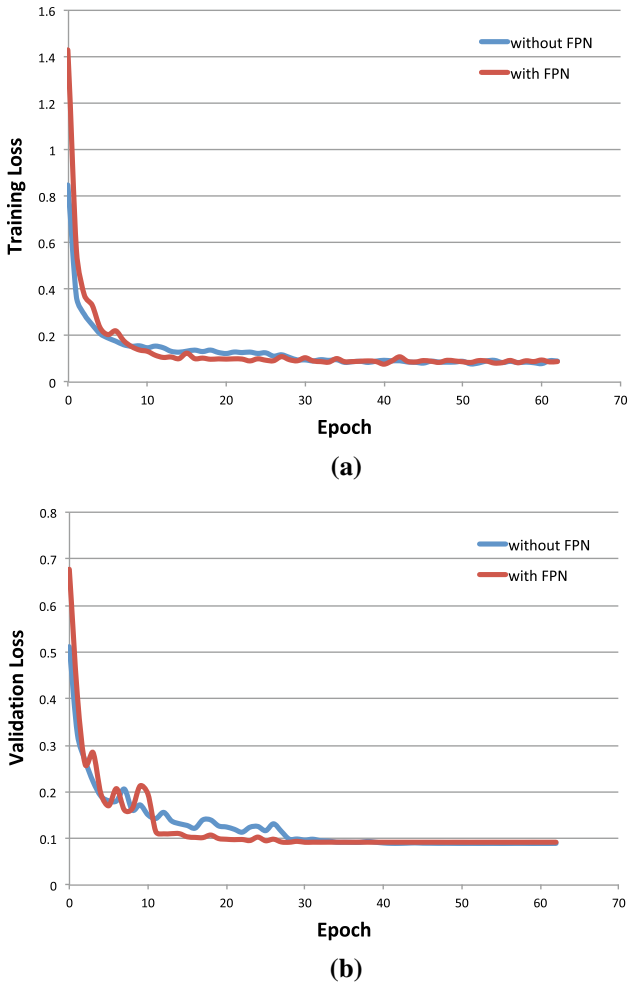

Fig. 7 Convergence performance for ResNet-50 with and without an FPN, **a** training and **b** validation

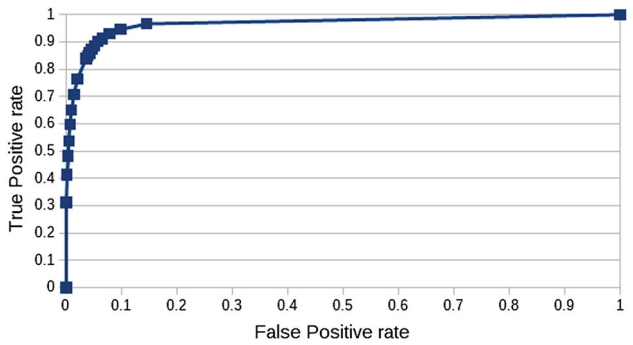

Fig. 8 The ROC curve for the proposed network

the number of pixels correctly detected as original. Thus, TPR and FPR are determined at the pixel level. They can be calculated by comparing the output binary mask with the truth binary mask. Threshold values from 0 to 1 in steps of 0.01 were used to calculate TPR and FPR for the ROC curve. The network performance can also be measured by calculating the area under the curve (AUC) which has a value between 0 and 1. A high AUC means high TPR and low FPR and thus precise model predictions. The proposed network has an AUC value of 0.967, which is excellent.

4 Conclusion

In this paper, a new image forgery detection method based on a new variation of Mask-RCNN network was introduced. A new backbone architecture called ResNet-conv was designed to create the initial feature map to train the Mask-RCNN. This new backbone is a simplified version of ResNet-FPN which is obtained by replacing the FPN with convolutional layers. ResNet-conv was shown to have the same convergence speed as ResNet-FPN. Two ResNet architectures were considered, ResNet-50 and ResNet-101. The convergence of ResNet-50 was shown to be faster. This is because features related to forgery are basic features that can be learned in the early layers of the network. Using additional layers does not improve the detection accuracy but rather slows convergence. Different augmentation techniques were used to create a model that is robust to several post-processing techniques. The ImageNet, Xavier_normal and He_normal initialization techniques were considered. Initialization with ImageNet weights provided better results than the other techniques. The proposed method was trained and evaluated using computer-generated forged images. Future work can consider generalizing the model to other kinds of image forgery such as copy-move forgery or image retouching. This can be done by tuning the model using datasets containing examples of these kinds of forgery.

Acknowledgements The authors would like to thank Radwa Hammad for her comments and advice that greatly improve the manuscript. They would also like to thank the anonymous reviewers for their insightful suggestions and comments.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Qazi, T., et al.: Survey on blind image forgery detection. IET Image Process. **7**(7), 660–670 (2013)
2. Mahdian, B., Saic, S.: Blind methods for detecting image fakery. In: Proceedings of the IEEE International Carnahan Conference on Security Technology. Prague, Czech Republic, Oct 13–16 (2008)
3. Shivakumar, B., Baboo, L.D.S.S.: Detecting copy-move forgery in digital images: a survey and analysis of current methods. Glob. J. Comput. Sci. Technol. **10**(7), 61–65 (2010)
4. Lin, C., Chen, C., Chang, Y.: An efficiency enhanced cluster expanding block algorithm for copy-move forgery detection. In: Proceedings of the IEEE International Conference on Intelligent

Networking and Collaborative Systems. Taipei, Taiwan, Sep 2–4 (2015)
5. Ardizzone, E., Bruno, A., Mazzola, G.: Copy-move forgery detection by matching triangles of keypoints. IEEE Trans. Inf. Forensics Secur. **10**(10), 2084–2094 (2015)
6. Cozzolino, D., Gragnaniello, D., Verdoliva, L.: Image forgery detection based on the fusion of machine learning and block-matching methods. arXiv preprint arXiv:1311.6934 (2013)
7. Bharati, A., et al.: Detecting facial retouching using supervised deep learning. IEEE Trans. Inf. Forensics Secur. **11**(9), 1903–1913 (2016)
8. Shi, Y.Q., Chen, C., Chen, W.: A natural image model approach to splicing detection. In: Proceedings of the Workshop on Multimedia & Security. Dallas, TX, USA, pp. 51–62, Sep 20–21 (2007)
9. He, Z., et al.: Digital image splicing detection based on Markov features in DCT and DWT domain. Pattern Recognit. **45**(12), 4292–4299 (2012)
10. Mushtaq, S., Mir, A.H.: Novel method for image splicing detection. In: Proceedings of the IEEE International Conference on Advances in Computing, Commun. and Informatics. New Delhi, India, Sep 24–27 (2014)
11. Dong, J., et al.: Run-length and edge statistics based approach for image splicing detection. In: Proceedings of the International Workshop on Digital Watermarking. Busan, Korea, Nov 10–12 (2008)
12. He, Z., Lu, W., Sun, W.: Improved run length based detection of digital image splicing. In: Proceedings of the International Workshop on Digital Watermarking. Lecture Notes in Computer Science, vol. 7128. Springer, Berlin, Atlantic City, NJ, USA, Oct. 23–26 (2011)
13. Lee, H., Ekanadham, C., Ng, A.Y.: Sparse deep belief net model for visual area V2. In: Proceedings of the Advances in Neural Information Processing Systems Conference. Vancouver, BC, Canada, Dec 3–5 (2007)
14. Hugo, L., et al.: Exploring strategies for training deep neural networks. J. Mach. Learn. Res. **10**, 1–40 (2009)
15. LeCun, Y., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the Advances in Neural Information Processing Systems Conference. Lake Tahoe, NV, USA, Dec 3–8 (2012)
17. Swietojanski, P., Ghoshal, A., Renals, S.: Convolutional neural networks for distant speech recognition. IEEE Signal Process. Lett. **21**(9), 1120–1124 (2014)
18. Tuama, A., Comby, F., Chaumont, M.: Camera model identification with the use of deep convolutional neural networks. In: Proceedings of the IEEE International Workshop on Information Forensics and Security. Abu Dhabi, UAE, Dec 4–7 (2016)
19. Baroffio, L., et al.: Camera identification with deep convolutional networks (2016)
20. Rao, Y., Ni, J.: A deep learning approach to detection of splicing and copy-move forgeries in images. In: Proceedings of the IEEE International Workshop on Information Forensics and Security. Abu Dhabi, UAE, Dec 4–7 (2016)
21. Rota, P., et al.: Bad teacher or unruly student: can deep learning say something in image forensics analysis? In: Proceedings of the IEEE International Conference on Pattern Recognition. Cancun, Mexico, Dec 4–8 (2016)
22. Bayar, B., Stamm, M.C.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: Proceedings of the ACM Workshop on Information Hiding and Multimedia Security. Galicia, Spain, Jun 20–22 (2016)
23. Wang, Q., Zhang, R.: Double JPEG compression forensics based on a convolutional neural network. EURASIP J. Inf. Secur. **2016**, 23 (2016)
24. Ying, Z., et al.: Image region forgery detection: a deep learning approach. In: Proceedings of the Singapore Cyber-Security Conference. Singapore, Jan 14–15 (2016)
25. Zhang, Z., et al.: Boundary-based image forgery detection by fast shallow CNN. In: Proceedings of the IEEE International Conference on Pattern Recognition. Beijing, China, Aug 20–24 (2018)
26. Ouyang, J., Liu, Y., Liao, M.: Copy-move forgery detection based on deep learning. In: Proceedings of the IEEE International Congress on Image and Signal Processing, BioMedical Engineering and Informatics. Shanghai, China, Oct 14–16 (2017)
27. Lin, T.Y., et al.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA, Jul 22–25 (2017)
28. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA, Jun 26–Jul 1 (2016)
29. Deng, J., et al.: Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL, USA, Jun 20–25 (2009)
30. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the International Conference on Artificial Intelligence and Statistics. Sardinia, Italy, May 13–15 (2010)
31. He, K., et al.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE International Conference on Computer Vision. Santiago, Chile, Dec 13–16 (2015)
32. He, K., et al.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. Honolulu, HI, USA, Jul 22–25 (2017)
33. Lin, T.Y., et al.: Microsoft COCO: common objects in context. In: Proceedings of the European Conference on Computer Vision. Zurich, Switzerland, Sep 6–12 (2014)
34. Ren, S., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
35. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. Santiago, Chile, Dec 13–16 (2015)
36. Meena, K., Tyagi, V.: Image forgery detection: survey and future directions. In: Shukla, R., Agrawal, J., Sharma, S., Singh Tomer, G. (eds.) Data, engineering and applications, pp. 163–194. Springer, Singapore (2019)
37. Braxmeier, J.: Stunning free images & royalty free stock. https://pixabay.com. Cited 09-Jun-2019 (2018)
38. Matterport Inc.: Mask-RCNN. https://github.com/matterport/Mask_RCNN. Cited 09-Jun-2019 (2017)
39. Qian, N.: On the momentum term in gradient descent learning algorithm. Neural Netw. **12**(1), 145–151 (1999)
40. Keras-team: Keras. https://github.com/fchollet/keras. Cited 09-Jun-2019 (2015)
41. Casella, G., Berger, R.L.: Statistical Inference, 2nd edn. Duxbury Press, Pacific Grove (2002)
42. Sakamoto, H.: On the distributions of the product and the quotient of the independent and uniformly distributed random variables. Tohoku Math. J., First Stage **49**, 243–260 (1943)