

Detection of image manipulation with convolutional neural network and local feature descriptors

Ali Ahmad Aminu^{1,2}, Nwojo Nnanna Agwu¹, Steve Adeshina¹, Muhammed Kabir Ahmed²

¹Computer Science Department, Faculty of Natural and Applied Sciences, Nile University of Nigeria, Abuja, Nigeria

²Computer Science Department, Faculty of Science, Gombe State University, Gombe State, Nigeria

Article Info

Article history:

Received Aug 29, 2021

Revised Mar 31, 2022

Accepted Apr 08, 2022

Keywords:

Convolutional neural network

Image manipulation chain

Image manipulation detection

Local feature descriptors

ABSTRACT

In recent times, numerous digital image manipulation detection approaches have been proposed to detect which processing operations were applied to manipulate digital images. Most of these approaches consider the situation in which an image is manipulated by only one manipulation operation. However, practical image manipulation often involves multiple manipulation operations. It is important to detect multiple image manipulation operations and the order in which they were applied to establish the origin and genuineness of a given image as well as the processing history it has gone through. In this article, we proposed a new method to determine multiple image processing operation and operation chains based on convolutional neural network (CNN) and local optimal oriented pattern (LOOP). The proposed method is based on CNN and LOOP in which CNN extracts and learns image manipulation traces from the LOOP maps of the input images that are classified using softmax, extra-tree, and extreme gradient boosting (XGBOOST) classifiers. Detailed experiments show that the proposed model can attain overall detection accuracies of 99.81% and 99.15% in identifying different image manipulations and manipulation operation chains, respectively.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Ali Ahmad Aminu

Computer Science Department, Faculty of Natural and Applied Sciences

Nile University of Nigeria, Abuja, Nigeria

Email: aaminu3@gsu.edu.ng

1. INTRODUCTION

In recent years, digital images have become the principal means of conveying information due to their expressive capacities and ease of distribution [1]. As a mode of communication, digital images have shown to be more effective information carrier than text information. As a result, images constituting a key source of evidence in judicial proceedings and criminal investigations are becoming more widespread than ever.

However, in most of these positive aspects where they are used, the nature of the digital image has created a lot of concerns. The information in digital images can be easily manipulated to convey fabricated or deceptive information. The introduction of advanced computers and user-friendly image manipulation tools allows anyone with basic image editing skills to alter them easily without leaving behind any human detectable traces [2]. As a result, image modifications for harmful reasons have become ubiquitous in our society, resulting in the spread of fake news, erroneous verdicts, and reputational damage among others [3]. Therefore, it is important to come up with techniques for verifying the authenticity of digital images before using them to make critical decisions.

A variety of image manipulation detection approaches have been proposed [4]-[8]. However, most of the early approaches have focused on detecting specific image manipulations such as image splicing and

copy move. They cannot identify more than one type of image tampering. Thus, methods capable of detecting multiple image manipulation are required.

In line with the foregoing, recent works in image manipulation detection have concentrated on building multi-purpose methods capable of detecting different types of image manipulation. Many multi-purpose image manipulation detection techniques based on handcrafted features have been presented. The work of [9] for example, proposed a universal image manipulation detection approach using residual-based features and the spatial rich model (SRM) [10] features used in steganalysis domain. In a similar study by Farooq *et al.* [11], investigated and demonstrated the performance of SRM and local binary pattern (LBP) [12] in identifying various image manipulation using synthesized datasets from the first Institute of Electrical and Electronics Engineers (IEEE) information security and technical committee (IFS-TC) image manipulation detection competition [13]. A recent study by Peng *et al.* [14] designed a universal feature set for multi-purpose image forensics that is capable of detecting different kinds of image manipulation operations concurrently using residual-based features.

Following the success of deep learning techniques, especially convolutional neural network (CNN), in a range of visual identification studies [15]-[18], recent researches in image forensics also try to employ this approach in solving the problem of digital image manipulation. Among the deep learning-based methods, Singh and Goyal [19] designed a general-purpose image manipulation detection approach that uses deep learning technique. They proposed the use of residual dense blocks in building their model to improve classification by utilizing global residual learning and local dense connection. Wei *et al.* [20], proposed a multi-purpose image tampering approach based on reinforcement learning that could design a deep neural network automatically without manual intervention. The method consists of a learning agent which is trained to choose layers of CNN automatically according to an algorithm known as the Q-learning algorithm [21]. Singhal *et al.* [22], described a multi-purpose image manipulation detection technique based on CNN and frequency domain features of image residuals. In another work by Bibi, and Abbasi [23], suggested an image manipulation detection approach based on CNN, auto-encoder, and ensemble subspace discriminant (ESD) learner. They utilized pre-trained networks, AlexNet and visual geometric group (VGG) for feature extraction. The extracted features were then forwarded to stacked auto-encoder for image manipulation detection and finally, the ESD learner is used for classification.

Although the recent studies have enhanced the performance of image manipulation detection systems, the existing methods mainly focused on distinguishing different image manipulation operations. Thus, determining different image manipulation operations and the order in which these operations are applied remains an open challenge. Therefore, this work proposes a novel multipurpose CNN and local optimal oriented pattern (LOOP) based image manipulation and manipulation chain detection system that combines the strength of both the handcrafted and deep learning methods. To test the proposed method's performance, we evaluated it on a variety of experiments and the results obtained show that the proposed method could detect image manipulation and manipulation chains with an average accuracies of 99.81% and 99.15%, respectively.

2. METHOD

This section gives an overview of the proposed approach for detecting image manipulation and manipulation operation chains as illustrated in Figure 1. The proposed method starts by transforming a colored input image to a grayscale image, which is then used to create the LOOP map of the input image. The designed CNN architecture then extracts and learns deep features from the input images' LOOP maps, which will be forwarded to the dense layers for classification. To further examine the effectiveness of the proposed CNN in feature extraction, deep features extracted from the proposed CNN's second dense layer were used to train and evaluate two other classifiers, extreme gradient boosting (XGBOOST) and extra tree classifier, on the detection and classification of image manipulations. In the subsections that follow, the key aspects of the proposed method are detailed.

2.1. Datasets

To train and assess the proposed model's performance, we collected 9605 images from three publicly available image databases; the Microsoft common object in context (MS COCO) dataset [24], the Bossbase dataset [25] and, the IEEE image manipulation dataset [13]. Each of these datasets contributes 3555, 1050, and 5000 images, respectively to form the 9605 composite image datasets. Finally, the total images (9605) was used to synthesize the training and testing datasets for the proposed model.

To create the complete data for the proposed experiments, each of the 9605 images was pre-processed as follows. Each of these 9605 images was first centrally cropped to 256×256 and converted to grayscale images to represent unaltered images. Then each of the 9605 images was manipulated using the five

manipulation operations in Table 1, resulting in a total of 57630 (9605×6) images, 9605 original images, and their corresponding five tampered versions. The LOOP maps for each of the dataset's six classes were then generated and utilized as input to the proposed model. Table 1 summarizes the types of manipulation operations and the parameters used for creating our experimental datasets

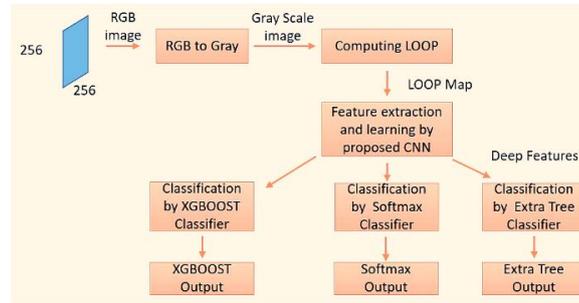


Figure 1. Illustration of the proposed method

Table 1. Image manipulation operations and their parameters

Operations	Parameters
Gaussian blurring (GB)	Kernel size (K_{size}) = 3×3, 5×5
Gamma correction (GC)	Gama (γ) = 0.5, 0.7, 0.8, 0.9
Median filtering (MF)	Kernel size (K_{size}) = 3×3, 5×5
Joint photographic experts group (JPEG) compression (JC)	Quality factor (qF) = 80, 90
Contrast enhancement (CE)	Not applicable (n/a)

2.2. LOOP computation

We used the LOOP maps of the images as the input of the proposed model, which is a measure of local image features recently proposed by Chakraborti *et al.* [26] and mostly used in the description and classification of texture. LOOP is an enhancement of LBP and local directional pattern (LDP) [27] which have gained much applications in numerous computer vision researches including image manipulation detections [11], [28], [29]. It has the capability to capture hidden texture variation that may results due to image manipulation operations. This motivates us to combine it with CNN for image manipulation detection.

Given an image I , the LOOP map can be computed using (1) and (2) [26]. Let i_c denote the image I 's intensity at pixel (x_c, y_c) , and let i_n ($n = 0, 1, \dots, 7$) denote the intensity of a pixel in the 3×3 neighbors, ignoring the center pixel. The 8 Kirsch masks are oriented in the direction of the 8 neighboring pixels ($n = 0, 1, 2, \dots, 7$), providing a measure of the strength of intensity variation in those directions. Assume that m_n denotes the responses of the eight Kirsch masks that correspond to pixels of varying intensities ($n = 0, 1, 2, \dots, 7$). Each of these pixels is given an exponential weight w_n (a number between 0 and 7) based on the rank of the magnitude of the 8 Kirsch mask activations. The LOOP map is created by aggregating the occurrence of LOOP values across the entire image.

Where:

$$LOOP(x_c, y_c) = \sum_{n=0}^7 f(i_n - i_c) \times 2^{w_n} \quad (1)$$

$$f(x) = \begin{cases} 0, & \text{if } x \geq 0 \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

2.3. Convolutional neural network design

We designed a convolutional neural network to extract and learn image manipulation traces directly from LOOP maps of the input image. The designed CNN is made up of 13 layers, with the first 10 being convolutional layers and the last three being the dense layers. The network is fed a 256×256 LOOP map, which is then used to extract low-level features using the convolutional filters.

The final classification is performed by three different classifiers – softmax, extra tree and, XGBOOST. The features extracted and learned by the set of convolution layers of the proposed CNN are passed on to the dense layers with a softmax activation, which classifies an image as either manipulated or authentic. However, because other approaches for obtaining final classification performance may surpass the performance of a dense layer with a softmax layer, as described in [2], we proposed using the extra tree and XGBOOST classifiers to produce the proposed model's final classification results as well. The types of layers, parameters and, output size of each layer of the proposed CNN architecture are summarized in Figure 2.

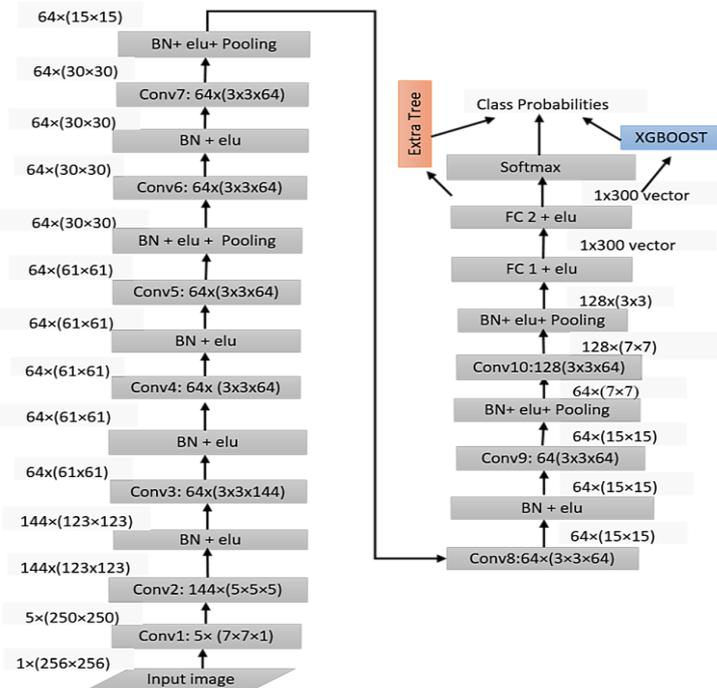


Figure 2. Proposed CNN Architecture for Image manipulation detection

2.4. Implementation details

The proposed model was trained using the Adam optimizer with the default settings for the moments ($\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-7}$), and a learning rate of 0.0001. The proposed model's weights were initialized using the Xavier initialization approach, with their biases set to 0. The biases of the three dense units were set to 0.05, and their weights initialized using random numbers drawn from a zero-mean Gaussian distribution with a standard deviation of 0.01. For each training iteration, the proposed model was trained with a batch size of 16 images to minimize the categorical cross-entropy loss without shuffling of the training data across epochs.

Suppose θ is the parameter vector denoting the weight vector corresponding to the image manipulation detection task, the categorical cross-entropy loss can be formulated as:

$$L(\theta) = \frac{-1}{M} \sum_{m=1}^M \sum_{n=1}^N 1(y^m = n) \log(y^m = n | x^m; \theta) \quad (3)$$

Where M and N represent the total number of image samples and the number of classes, $1(\cdot)$ denotes an indicator function which equals 1 if $m = n$, otherwise 0. y^m and x^m denote the image label and the feature of the sample m .

3. RESULTS AND DISCUSSION

We conducted several experiments under various experimental settings to evaluate the proposed image manipulation detection method. The model's performance in detecting individual manipulation operations is evaluated in the first experiment. In the second experiment, all of the tampering operations specified in Table 1 are used to test the model's ability to detect and classify multiple image tampering. Finally, we put the model to the test in terms of recognizing the image manipulation operation chain in compressed and uncompressed image formats.

3.1. Individual image manipulation detection

The proposed model is utilized in this experiment to determine if a given image is authentic or has been manipulated using any of the manipulation techniques given in Table 1. Five different experiments were performed with each corresponding to that of one manipulation technique. To carry out these experiment, we synthesized 5 manipulated versions of the authentic images as described in section 2.1 which resulted in 5 classes of images, each corresponding to one manipulation operation given in Table 1 and having more than 9000 images.

Using the implementation details described in section 2.4, the proposed binary classifier was trained and evaluated for 100 epochs with 14000 and 2000 subsets of the dataset respectively, and the remaining 3000 was reserved for testing the model. To further evaluate the feature extraction capabilities of the proposed method, the features extracted by the proposed model was also used to train and assess two tree based models, XGBOOST and extra tree for the detection of individual image manipulation detection. The manipulation detection results obtained by the three models are presented in Table 2.

The proposed CNN can detect the various types of image manipulations with an accuracy of at least 99.10%, as shown in the table of results. The proposed model obtained the best detection rates of 100% in JPEG compression (JC) and Gaussian blurring (GB) image manipulations respectively. This indicate that the CNN and LOOP model could detect the different individual image manipulation operations.

Additionally, we also observed that the performance of the extra tree based CNN when compared to that of the proposed CNN, led to some performance gain in the detection rates of each manipulation operation, with the exception of GB manipulation detection, where the proposed CNN got better detection accuracy. Similarly, the proposed CNN's performance in gama correction (GC) and median filtering (MF) was also improved by the XGBOOST-based CNN classifier, with the exception of contrast enhancement (CE), where the proposed CNN outperformed the XGBOOST-based CNN. In general, the extra tree-based CNN outperformed the other two classifiers in recognizing individual image manipulation operations. The obtained results show that the proposed approach is robust at extracting the necessary image manipulation fingerprints required for identifying various types of image manipulation operations and its performance can be enhanced by using other classifiers.

Table 2. Individual image manipulation detection accuracies (%) of the proposed CNN and tree based classifiers

Classifiers/manipulation	CE	GB	GC	MF	JC
Softmax	99.10	100	99.90	99.64	100
Extra tree	99.13	99.14	100	99.90	100
XGBOOST	99.03	100	99.92	99.94	100

3.2. Comparison with other methods

We compare the proposed model's individual image manipulation detection results with the results of [2], [9], [14] to further confirm the proposed model's performance on identifying individual image manipulation. The proposed model's accuracies on individual image manipulation in comparison to previously reported results are presented in Table 3. The highlighted numbers represent the best outcome for that manipulation operation. The “-” indication indicates that the given technique did not take that manipulation operation into account when evaluating their model. As shown in Table 3, our approach outperforms the methods of [2], [14] in GB, MF, and JC detections. Furthermore, the result of our method on GC revealed that it improved the works in [9] performance by 3.24%. The only exceptions were in CE and MF, in which the method of [9] outscores the proposed method by 0.08% and 0.05%, respectively. These results show that our approach has superior feature extraction and discriminating abilities, making it more resilient than existing methods in the majority of binary classification problems.

Table 3. Comparison with similar approaches from the literature

Methods/manipulations	CE	GB	GC	MF	JC
Proposed	99.13	100	100	99.94	100
[2]	-	99.95	-	99.71	99.66
[9]	99.95	100	96.76	99.99	99.94
[14]	-	96.4	-	96.7	95.8

3.3. Multiple image manipulation detection

In this experiment, the proposed approach is utilized to detect different types of image manipulations. The training, validation, and test sets were all created in the same manner as stated in the preceding experiment. Table 4 and Figure 3 show the classification results of the three classifiers in terms of average detection accuracies and confusion matrices respectively.

The experimental results indicate that our approach can detect multiple image manipulations with an average detection rate of 99.81% and each manipulation operation with an accuracy of at least 98.27% in Figure 3(a). The average detection rates of extra tree-based CNN and XGBOOST based CNN are also shown in Table 4. The extra tree-based CNN and XGBOOST based CNN could detect each of the manipulation operations with an accuracy of not less 97.80% in Figure 3(b) and 98.80% in Figure 3(c), respectively. When the results obtained are observed more closely, it can be seen that all the classifiers could detect the multiple image manipulation with a detection rate of not less than 99.43%. In contrast to the results in the individual

image manipulation detection, the proposed model obtained better results than the extra tree based and XGBOOST based CNNs in the multiple-image manipulation detection. This results further demonstrate the robustness of the proposed method in feature extraction and the discriminative powers of these features. Furthermore, the results also show that our approach can successfully subdue the impact of image contents, thus, allowing the proposed model to extract the necessary image manipulation fingerprints needed for image manipulation detection.

Table 4. Multiple image tampering detection accuracies (%) of proposed CNN and tree based classifiers

Methods	Proposed CNN	Extra-CNN	XGBOOST-CNN
Average accuracy	99.81	99.43	99.60

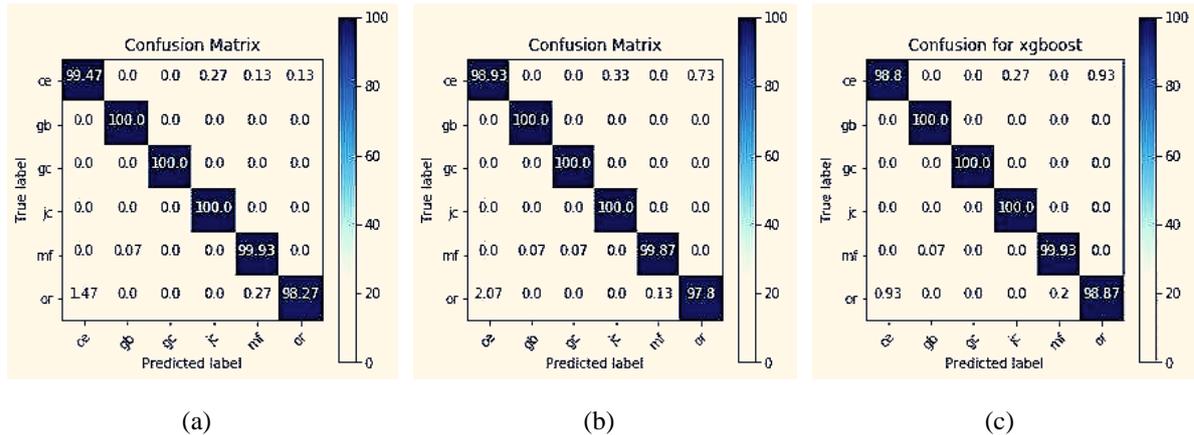


Figure 3. Comparing confusion matrices of (a) the proposed CNN, (b) extra tree-CNN, and (c) XGBOOST-CNN

3.4. Manipulation operation chain detection

We also test the effectiveness of the proposed model in identifying image manipulation operation chains in both uncompressed and compressed image formats. To conduct these experiments, each of the original images was manipulated by a chain of (A-B) manipulation operations. A-B indicates that the original image is manipulated first using “A” and then “B” manipulation operations. For instance, MF-GB will correspond to first applying median filtering operation followed by gaussian blurring operation. Applying the following image manipulation operation chains (CE-GB), (GB-CE), (GB-MF), (GC-GB), (GC-MF), and (MF-GB), we obtained 9605 images from the original image and each manipulation operation chain, resulting in a total of 67235 (9605×7) images.

Additionally, we considered two different cases where a manipulation operation chain has been followed with and without a JPEG compression with a quality factor of 80. The LOOP maps of each class of the datasets was then obtained which served as the input of the model. The datasets was then divided at random to create the training, validation, and test sets, which were respectively used for training, validating and testing the proposed model.

Table 5 depicts the average detection accuracies of the proposed models in detecting image manipulation operation chains in the uncompressed and compressed image format. The confusion matrices of the proposed method for identifying image manipulation operation chains in the uncompressed and compressed image manipulation chains are presented in Figure 4 and Figure 5 respectively. From the results, it can be observed that each of the three classifiers could detect all the manipulation operation chains with high detection rates in both cases.

Additionally, looking at the diagonal elements of Figure 4, we can notice that the proposed CNN could detect each manipulation operation chain with at least 98.0% in Figure 4(a), in the uncompressed manipulation operation chain detection. Similarly, the extra tree and XGBOOST based CNNs were able to detect all the manipulation operation chains with at least 99.36% in Figure 4(b) and 99.40% in Figure 4(c) accuracies, respectively. The extra tree and XGBOOST based CNNs improved the performance of the proposed CNN by 1.36 % and 1.40%, respectively in the uncompressed manipulation operation chain detection.

Moreover, in the compressed manipulation operation chains detection, the proposed model, extra tree-based CNN, and XGBOOST based CNN could detect each manipulation operation chain with at least

98.20% in Figure 5(a), 95.7% in Figure 5(b), and 86.5% in Figure 5(c) respectively as can be seen from the diagonal elements of Figure 5. When compared to their performances in manipulation chain detection in the uncompressed image format, the performance of the extra tree and XGBOOST based CNNs declined by 3.86% and 12.9%, respectively, indicating the effect of applying JPEG compression after the manipulation operation chains. These results indicate that, in addition to being good at identifying multiple image manipulation operations, the proposed model can also detect numerous manipulation operation chains in images that have been subjected to JPEG compression with high detection accuracies, which is a more difficult and practical problem.

Table 5. Manipulation chain operation detection accuracies (%) of proposed model and tree-based classifiers

Image Type/methods	Proposed CNN	Extra tree-CNN	XGBOOST-CNN
Uncompressed	99.15	99.57	99.51
Compressed	99.18	98.92	97.20

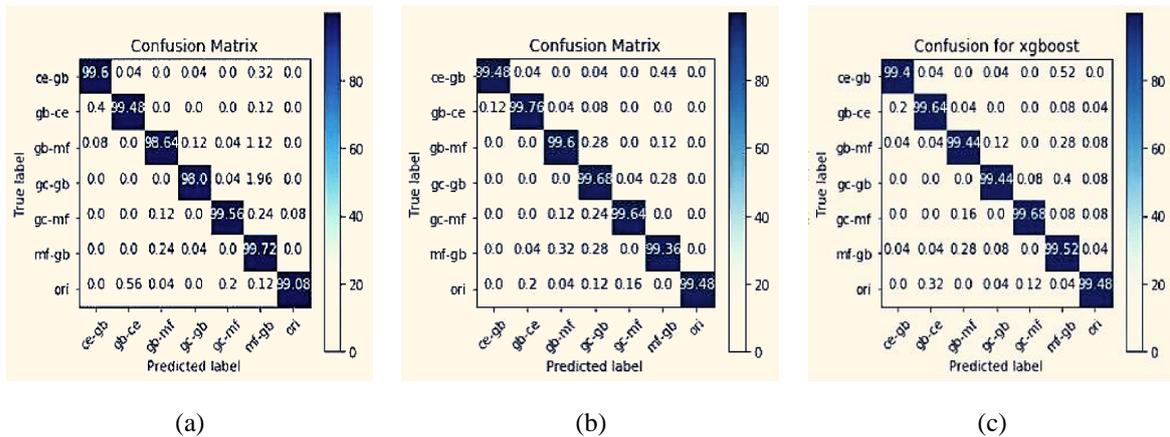


Figure 4. Comparing confusion matrices of the proposed CNN and tree based models for manipulation chain detection in uncompressed images (a) proposed CNN, (b) extra tree-CNN, and (c) XGBOOST-CNN

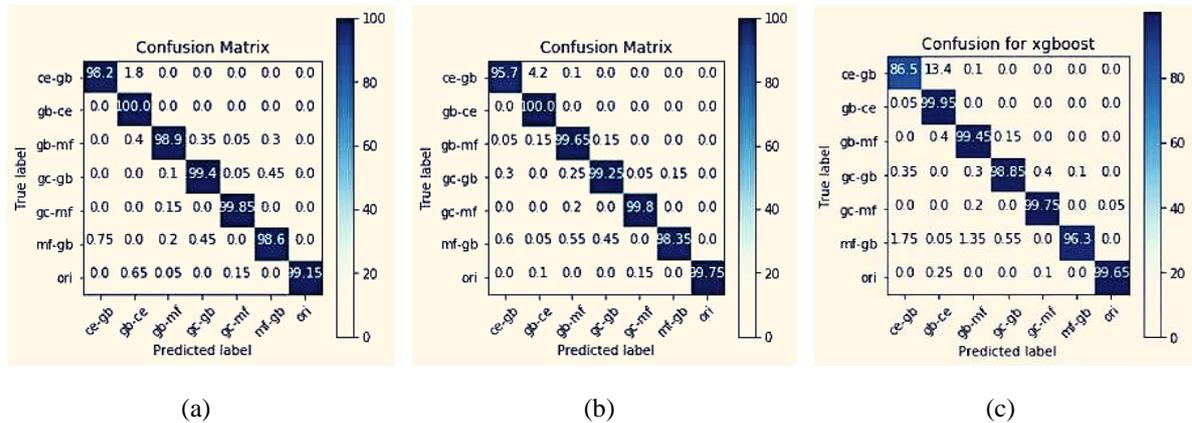


Figure 5. Comparing confusion matrices of the proposed CNN and tree based models for manipulation chain detection in compressed images (a) proposed CNN, (b) extra tree, and (c) XGBOOST

4. CONCLUSION

In this paper, we proposed a novel image manipulation detection approach that uses CNN and local feature descriptors to enhance image manipulation detection. Unlike prior methods that focused only on detecting different image tampering operations, the proposed method can detect both multiple manipulation operations as well as the order in which they were applied. We combine the strength of LOOP, a handcrafted feature, and CNN to build robust image manipulation detection methods. Our approach can extract image manipulation fingerprints directly from the LOOP maps of the input images, allowing it to extract the various fingerprints required to identify various image manipulation types. We evaluated the proposed model's

performance in several experiments, and the results show that it was effective in identifying different image manipulation operations as well as the order in which these operations were applied.

ACKNOWLEDGEMENTS

The authors wish to thank and acknowledge the National Information Technology Development Agency (NITDA) of Nigeria for their financial support.

REFERENCES

- [1] J. A. Redi, W. Taktak, and J. L. Dugelay, "Digital image forensics: a booklet for beginners," *Multimedia Tools and Applications*, vol. 51, no. 1, pp.133-162, 2011, doi: 10.1007/s11042-010-0620-1.
- [2] B. Bayar and M. C. Stamm, "Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection," in *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2691-2706, Nov. 2018, doi: 10.1109/TIFS.2018.2825953.
- [3] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, 2016, pp. 5-10, doi: 10.1145/2909827.2930786.
- [4] H. Tang, R. Ni, Y. Zhao, and X. Li, "Median filtering detection of small-size image based on CNN," *Journal of Visual Communication and Image Representation*, vol. 51, pp. 162-168, Feb. 2018, doi: 10.1016/j.jvcir.2018.01.011.
- [5] J. -Y. Sun, S. -W. Kim, S. -W. Lee, and S. -J. Ko, "A novel contrast enhancement forensics based on convolutional neural networks," *Signal Processing: Image Communication*, vol. 63, pp. 149-160, 2018, doi: 10.1016/j.image.2018.02.001.
- [6] A. N. Harish, V. Verma, and N. Khanna, "Double JPEG Compression Detection for Distinguishable Blocks in Images Compressed with Same Quantization Matrix," *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2020, pp. 1-6, doi: 10.1109/MLSP49062.2020.9231749.
- [7] Y. Abdalla, M. T. Iqbal, and M. Shehata, "Copy-move forgery detection and localization using a generative adversarial network and convolutional neural-network," *Information*, vol. 10, no. 9, p. 286, 2019, doi: 10.3390/info10090286.
- [8] Y. Rao, J. Ni, and H. Zhao, "Deep Learning Local Descriptor for Image Splicing Detection and Localization," in *IEEE Access*, vol. 8, pp. 25611-25625, 2020, doi: 10.1109/ACCESS.2020.2970735.
- [9] H. Li, W. Luo, X. Qiu, and J. Huang, "Identification of Various Image Operations Using Residual-Based Features," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 1, pp. 31-45, Jan. 2018, doi: 10.1109/TCSVT.2016.2599849.
- [10] J. Fridrich and J. Kodovsky, "Rich Models for Steganalysis of Digital Images," in *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868-882, June 2012, doi: 10.1109/TIFS.2012.2190402.
- [11] S. Farooq, M. H. Yousaf, and F. Hussain, "A generic passive image forgery detection scheme using local binary pattern with rich models," *Computers and Electrical Engineering*, vol. 62, pp. 459-472, 2017, doi: 10.1016/j.compeleceng.2017.05.008.
- [12] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," *Proceedings of 12th International Conference on Pattern Recognition*, 1994, vol.1, pp. 582-585, doi: 10.1109/ICPR.1994.576366.
- [13] IEEE IFS-TC Image Forensics Challenge Dataset. [Online]. Available: <https://signalprocessingsociety.org/newsletter/2013/06/ifs-tc-image-forensics-challenge>, (Accessed on 5 May 2021).
- [14] A. Peng, K. Deng, S. Luo, and H. Zeng, "Multi-purpose forensics of image manipulations using residual-based feature," *Computer, Materilas, and Continua*, vol. 65, no. 3, pp. 2217-2231, 2020, doi: 10.32604/cmc.2020.011006.
- [15] S. Kumar, J. S. Yadav, Y. Kurmi, and A. Baronia, "An efficient image denoising approach to remove random valued impulse noise by truncating data inside sliding window," *2nd International Conference on Data, Engineering and Applications (IDEA)*, 2020, pp. 1-7, doi: 10.1109/IDEA49133.2020.9170689.
- [16] C. -H. Lee, Z. Liu, L. Wu, and P. Luo, "MaskGAN: Towards Diverse and Interactive Facial Image Manipulation," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5548-5557, doi: 10.1109/CVPR42600.2020.00559.
- [17] H. Dharmo *et al.*, "Semantic Image Manipulation Using Scene Graphs," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5212-5221, doi: 10.1109/CVPR42600.2020.00526.
- [18] R. Popli, I. Kansal, A. Garg, N. Goyal, and K. Garg, "Classification and recognition of online hand-written alphabets using machine learning methods," in *IOP Conference Series: Materials Science and Engineering*, vol. 1022, pp. 1-9, 2021, doi: 10.1088/1757-899X/1022/1/012111.
- [19] G. Singh and P. Goyal, "GIMD-Net: An effective General-purpose Image Manipulation Detection Network, even under anti-forensic attacks," *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9533733.
- [20] Y. Wei, Y. Chen, X. Kang, Z. J. Wang, and L. Xiao, "Auto-Generating Neural Networks with Reinforcement Learning for Multi-Purpose Image Forensics," *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1-6, doi: 10.1109/ICME46284.2020.9102943.
- [21] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, LDN, UK, 1989. [Online]. Available: <https://www.researchgate.net/publication/33784417>.
- [22] D. Singhal, A. Gupta, A. Tripathi, and R. Kothari, "CNN-based multiple manipulation detector using frequency domain features of image residuals," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 4, pp. 1-26, 2020, doi: 10.1145/3388634.
- [23] S. Bibi, and A. Abbasi, I. Ul Haq, S. W. Baik, and A. Ullah, "Digital image forgery detection using deep autoencoder and CNN features," *Human Centric Computing and information sciences*, vol. 11, no. 32, Aug. 2021, doi: 10.22967/HCCIS.2021.11.032.
- [24] T. -Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 740-755, doi: 10.1007/978-3-319-10602-1_48.

- [25] P. Bas, T. Filler, and T. Pevny, "Break our steganographic system: The ins and outs of organizing BOSS," in *Conference paper International Workshop on Information Hiding*, 2011, pp. 59-70, doi: 10.1007/978-3-642-24178-9_5.
- [26] T. Chakraborti, B. McCane, S. Mills, and U. Pal, "LOOP Descriptor: Local Optimal-Oriented Pattern," in *IEEE Signal Processing Letters*, vol. 25, no. 5, pp. 635-639, May 2018, doi: 10.1109/LSP.2018.2817176.
- [27] T. Jabid, M. H. Kabir, and O. Chae, "Gender Classification Using Local Directional Pattern (LDP)," *2010 20th International Conference on Pattern Recognition*, 2010, pp. 2162-2165, doi: 10.1109/ICPR.2010.373.
- [28] R. K. Remya and W. Wilscy, "Image forgery detection using deep textural features from local binary pattern map," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 6391-6401, 2020, doi: 10.3233/JIFS-179720.
- [29] S. Sharma and U. Ghanekar, "Spliced image classification and tampered region localization using local directional pattern," *International Journal of Image, Graphics and Signal Processing*, vol. 11, no. 3, 2019, doi: 10.5815/ijigsp.2019.03.05.

BIOGRAPHIES OF AUTHORS



Ali Ahmad Aminu    is a Ph.D. student at Nile University of Nigeria, Abuja. His Research interest includes image forensics, deep learning, and image analysis. He can be contacted at email: aaminu3@gsu.edu.ng.



Nwojo Nnanna Agwu    Professor of Mathematics and Computer Science, Nile University of Nigeria, Abuja. His research interest includes artificial intelligence, optimization, numerical computation and control system theory. He can be contacted at email: nagwu@nileuniversity.edu.ng.



Steve Adeshina    Professor of Computer Engineering and Computer Vision, Nile University of Nigeria, Abuja. His research areas includes computer vision, medical image analysis and medical imaging. He can be contacted at email: steve.adeshina@nileuniversity.edu.ng.



Muhammed Kabir Ahmed    Senior Lecturer of Computer Science at Computer Science department, Gombe state University. His area of research include multi agent based systems, computational intelligence and deep learning. He can be contacted at email: mkahmed@gsu.edu.ng.