SN Computer Science (2022) 3: 392
https://doi.org/10.1007/s42979-022-01311-w

ORIGINAL RESEARCH

Discovering Tampered Image in Social Media Using ELA and Deep Learning

Sunen Chakraborty[1] · **Kingshuk Chatterjee**[2] · **Paramita Dey**[1]

Received: 8 June 2021 / Accepted: 6 July 2022 / Published online: 23 July 2022

Abstract
In the era of social media, we have access to millions of images. Nowadays with the rise of many advanced photo editing software finding a tampered image online is a very common situation. Most of the time an image is tampered for fun, but there are scenarios where an image is tampered with malicious intent and can cause harm to society. Digital image forensics is having a tough time dealing with tampered images due to the advancement of technology. Here, in our approach, we combined error level analysis (ELA) with a convolutional neural network (CNN) to classify whether an image is authentic or not. Our experiment has yielded a validation accuracy of 96.18% after 24 epochs.

Keywords Image tampering · Error level analysis · Deep learning · Convolutional neural networks

Introduction

Majority of the people on this planet use social media platforms. This leads to an emergence of trend where an image or a video is tampered to spread amusement which is now known as "Meme Culture". Nowadays seeing tampered images on the internet is a common situation, but there are multiple cases where an image or a video is tampered for nefarious purposes. Creation of these tampered images leads to a rise in the numbers of fake news. Digital image forensics is facing problems in finding new methods of detection.

These days the amount of data available and development in technology leads to colossal growth in the field of deep learning. Deep learning models proved their value when it comes to image processing and computer vision. CNNs are the most favoured of all deep learning models. CNNs are getting more recognition than other deep learning models because they can extract and learn features automatically. Also adding more data can increase the performance of the CNN. Chen et al. [1] first applied a CNN to detect tampering in images.

Even though CNNs are faster, can automatically understand and establish relationships between the features in images and can improve their performance with the amount of data. But it is not suitable for tampering detection in its usual configuration. When Chen et al. [1] used tampered images directly as an input to their CNN it learned the features of the images rather than the aspects related to tampering. It happened because the evidences of tampering are present in the underlying statistics of the images. Due to this issue, some researchers chose to pre-process the images using suitable methods like Huang et al. [2], some decided to add an additional layer in their CNN architecture like Chen et al. [1], Bayar et al. [3], while some utilized a variation of CNN or other deep learning models like Region-based CNN (R-CNN) like Zhou et al. [4], Fully Convolutional Network (FCN) like Salloum et al. [5], Deep Neural Network (DNN) like Wu et al. [6], Autoencoder like Zhang et al. [7].

To deal with the issue mentioned above we chose a procedure that can assist a CNN to detect image forgery. Error level analysis [8] is a method in which tampering on images of lossy compression can be detected. It means when an image undergoes compression some of its information is lost. In this process, first a suspected tampered image is

"This article is part of the topical collection "Social Data Science: Research Challenges and Future Directions" guest edited by Sarbani Roy, Chandreyee Chowdhury and Samiran Chattopadhyay."

✉ Paramita Dey
paramitadey@gcect.ac.in

[1] Department of Information Technology, Government College of Engineering & Ceramic Technology, Kolkata 700 010, India

[2] Department of Computer Science & Engineering, Government College of Engineering & Ceramic Technology, Kolkata 700 010, India

compressed and then the difference between pixel intensity of that image before and after compression is calculated. Altered regions can be spotted easily because of having different error levels than the unaltered regions.

The main contribution of this paper is as follows-

- The created CNN model has the smallest size but it achieves comparable accuracy with respect to the different models present in literature.
- It needs less resource for its execution.
- It has lesser number of parameters which lead to a decrease in the training time of the model.

The rest of the paper is arranged into four segments. In "Related Works", we briefly describe about other previously proposed methods. "Methodology" describes the procedure we used, then "Experimental Results and Discussions" states the results we get after conducting our experiment. Eventually, "Conclusion" depicts the conclusion and the direction of future research.

Related Works

Different machine learning and deep learning methodologies are applied for identification of fake image. Gunawan et al. [12][1] used a CNN to classify between authentic and tampered image. They used 80% data for training and 20% data for validation which leads to a poor approximation and low accuracy of their model. Sudiatmika et al. [13] used VGG-16 for tampering detection but bigger network have huge size, which means it needs a considerable amount of time and computational resources. Also deep networks suffer from vanishing gradient problem which makes it more difficult to train the network. That is the reason of poor performance compare to other models.

Kanwal et al. [14] first extract the chroma components of an image. Then in the first part, feature vectors are generated using DCT over different local feature descriptor. In the second part, Fourier transform is applied and final feature vectors are generated using an enhanced version of local feature descriptor. The feature vectors are fed into SVM classifiers. Still we can see that the accuracy is low because the features are generated using traditional and handcrafted methods.

Doegar et al. [15] feed the real and tampered images directly into an AlexNet model that was previously trained. Features extracted by the network are then used as an input for an SVM to classify. They didn't utilize any particular strategy that can spot the hidden indications of tampering. Thakur et al. [16] proposed a method to detect copy-move and splicing. First, the images are resized and transformed into greyscale. Then, traces of median filtering and image blurring are detected using suitable methods which are common post-processing employed to hide the tampering. Lastly, a CNN is used to classify the images.

Zhang et al. [17] applied a CNN in combination with ELA for classifying whether an image is DeepFake or not. The size of their model is 225 MB and their total model parameters are 2.95×10^7. Doegar et al. [18] employed three deep residual networks whose combined features are then used for training a classifier. Using residual networks may help with the problem of vanishing gradient. However deeper networks tend to learn more and unnecessary information from an image which may lead to over fitting as noticed by Zhang et al. [19].

In this paper, we propose an algorithm, which combined error level analysis (ELA) with a convolutional neural network (CNN) to classify authentic or fake image. This methodology yields the validation accuracy of 96.18% after 24 epochs.

Methodology

We pre-processed the authentic and tampered images before feeding them to the CNN. In our approach the first step is to generate the error level analysis (ELA) [8] of original and fake images, then we resize the images and normalize the pixels, after which we add the labels accordingly. Then we split the data into training set and validation set. In the second step, we feed the images and their corresponding labels into the CNN for training our model. Complete process is illustrated in Fig. 1.

Error Level Analysis (ELA)

The idea of ELA was proposed by Neal Krawetz et al. [8]. This technique is performed on an image that uses lossy compression, mostly JPEG images. If an image of JPEG format is tampered and resaved as a JPEG image again, then some of its information is lost after compression. The task of ELA is to resave an image at a notable error rate of 95% compression, and evaluating the difference between the original image and resaved image. Since JPEG images consist of 8×8 blocks, after compression all of the blocks should have almost similar error levels. In the case of tampering, modified areas can be easily identified because 8×8 blocks of these areas will have a different error level than the areas that have not been modified. The functioning of ELA is shown in Figs. 2, 3, and 4.

From the above three rows, we can observe that the more times an image gets resaved, the more its information gets lost. From Figs. 2 and 3 we can see that the changes in the

[1] https://github.com/agusgun/FakeImageDetector.

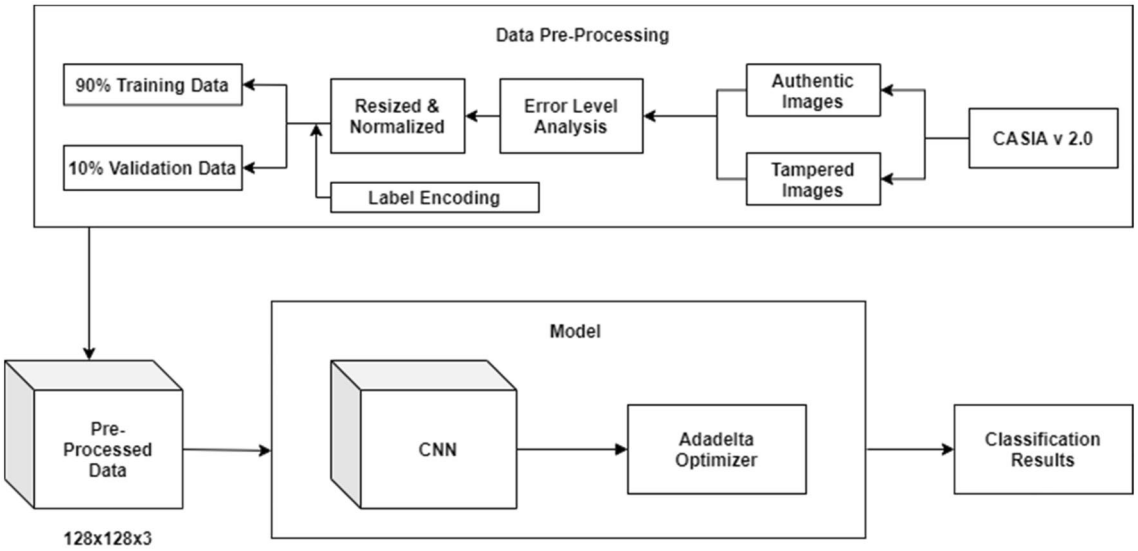

Fig. 1 Outline of the overall process

Fig. 2 An authentic image and its ELA

Fig. 3 Resaved image at 75% compression and its ELA

Fig. 4 Tampering of 75% resaved image and it's ELA

authentic and resaved images are imperceptible to human eyes but the differences are clear in their corresponding ELA. In Fig. 4 some aspects of the images are changed like the building is copied and the paraglider and helicopter are added. From the ELA of Fig. 4, it can be clearly seen that regions which have undergone tampering have a different error level than other regions.

Convolutional Neural Network (CNN)

Convolutional neural networks was designed by LeCun et al. [9], where it was used to recognize hand-written digits from the images. The task of the CNN is to reduce the data into a structure that is simpler to process, without losing the attributes which are essential for getting satisfactory results. CNNs are mostly utilized for working with 2-dimensional data like images or videos. Just like other neural networks CNN also have three kinds of layers, input layer, hidden layers, and output layer.

CNN Architecture of Proposed Method

Convolutional Layer

As the name suggests CNN uses convolution operation to convert the data into a map consisting of features.

It is the central component of a CNN. At first, there is an input layer of shape: (input image height) × (input image width) × (input image channels), followed by a convolutional layer which is a set of filters/masks/kernels used to change the input image into a map by separating the features. A filter is a matrix that is smaller than the input image. Individual filter slide across the width and height of the input image and performs convolution, in other words generate dot product of the filter with a patch of the image whose size is equivalent to the filter producing information of all the spatial locations. Each filter is capable of capturing an important feature. After all the filters completely pass over the image it generates a feature map which is passed to the next layer. *Proposed model uses two convolutional layers.*

Pooling Layer

The convolutional layer sums up the number of features by generating a feature map of an image. A major problem is that it focuses on the positions of the features more than the relationships between those features. This increases the number of parameters and computation time needed by the network. Also slight changes in features' locations may create problems. To handle these issues pooling layers are used. Pooling layers used a downsampling method with makes it easier to process the information and compress the size of feature maps. This makes the model more resilient to changes and also reduces the number of parameters and computation time. There are three kinds of pooling operation which are, max pooling, average pooling, and global pooling. we have used max-pooling.

In max-pooling, a filter is used over the feature map generated by the convolutional layer in a non-overlapping manner. Now only the maximum element will be extracted from the area covered by the filter. In this manner, only important elements from each feature of the feature map are considered. *Proposed model uses two max-pooling layers.*

Fully Connected Layer

These are the last layers in a CNN. It performs classification operations like an artificial neural network dependent on the information extracted by the preceding layers of a CNN. Output from the last convolutional or pooling layer in a 3-dimensional format must be converted into a vector before passing it into the fully-connected layer. The output

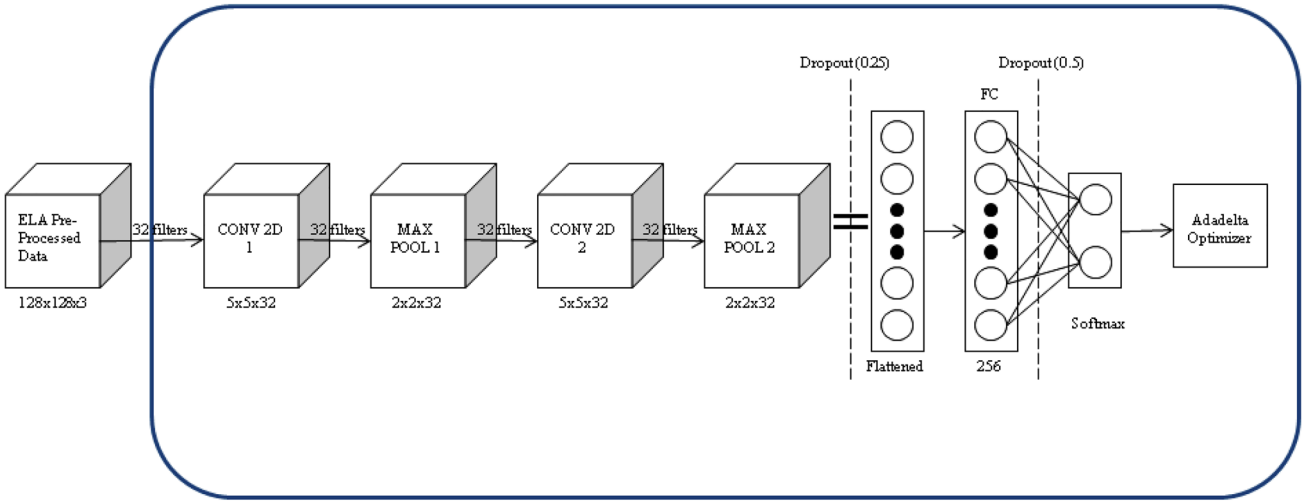

Fig. 5 Architecture of proposed CNN

layer is generally a layer with softmax activation function which converts the input vector of the fully connected layer into a probability vector that generates the probability of each class label in the CNN. *Our model uses one fully connected layer and a two-way softmax layer.*

Tuning Parameters

During training period we need to change the parameters of the model to bring down the loss as much as possible which will help our model to make more accurate predictions thus optimizing it further. The algorithms or methods which help us to modify these parameters are called optimizers. To tune our model during training, we used "*Adadelta*" [10] as the optimizer. After passing the result using the "softmax" function, the function utilized to minimize the variation between actual result and predicted result is called loss function, we used "*categorical_crossentropy*" as the loss function to tune our model. Architecture of our CNN is shown in Fig. 5.

Experimental Results and Discussions

Experimental Setup

All of our experiments are conducted using Jupyter Notebook available on Google Colab. Training of the model is performed using a GPU runtime on Google Colab which assigned a RAM of 12.72 GB and a Disk Space of 68.40 GB.

Dataset

For training our model we chose the CASIA dataset [11] available on Kaggle. More specifically we chose the CASIA v2.0 dataset because CASIA v1.0 dataset contains fewer samples. CASIA v2.0 dataset consists of 7492 authentic images and 5124 tampered images of various lossy and lossless formats. We chose CASIA v 2.0 dataset because the images in CASIA v 2.0 are tampered in two ways. First one is copy-move tampering in which part of an image is copied and pasted back to another part of the same image, it is usually done to hide some features or add some extra features in an image. Second one is splicing, here objects from two or more images are combined to form a tampered image. Both copy-move and splicing are basic kinds of tampering, which is why this dataset is more suitable for tampering detection. Figure 6 shows samples of authentic images while Fig. 7 shows samples of tampered images from the dataset.

Training and Performance

In our experiment, we chose a combination of images with one lossy format which is JPEG and one lossless format which is PNG and discarded images of other formats. After that our dataset contained 9418 authentic and tampered images. After generating ELA of the images and adding their labels, the data is split into 90% for training and 10% for validation. Then the images are fed into the neural network for training up to 40 epochs. The curve for accuracy and loss is shown in Fig. 8 where x-axis represents number of epochs and y-axis shows the value of accuracy and loss in Fig. 8a and b respectively.

The training process stops at epoch 24. As you can see from the figures above that our model achieved training accuracy of 98.34% and validation accuracy of 96.18%. Then we evaluate our model over the validation data whose confusion matrix is presented below in Fig. 9.

Fig. 6 Authentic samples from CASIA v 2.0

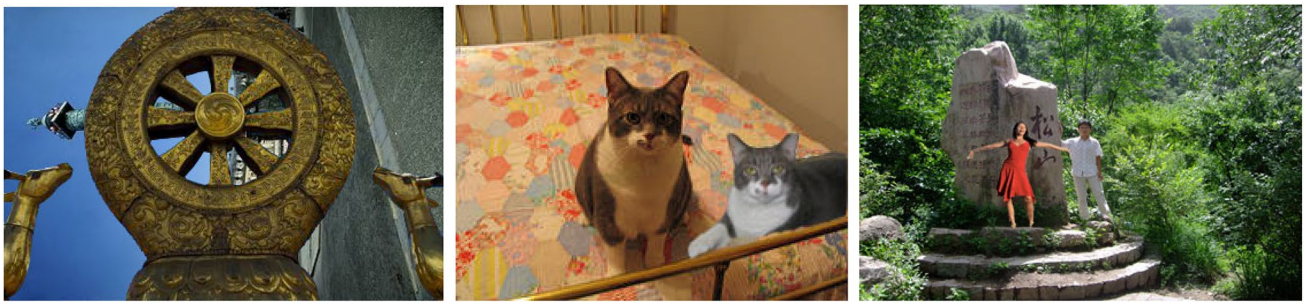

Fig. 7 Tampered samples from CASIA v 2.0

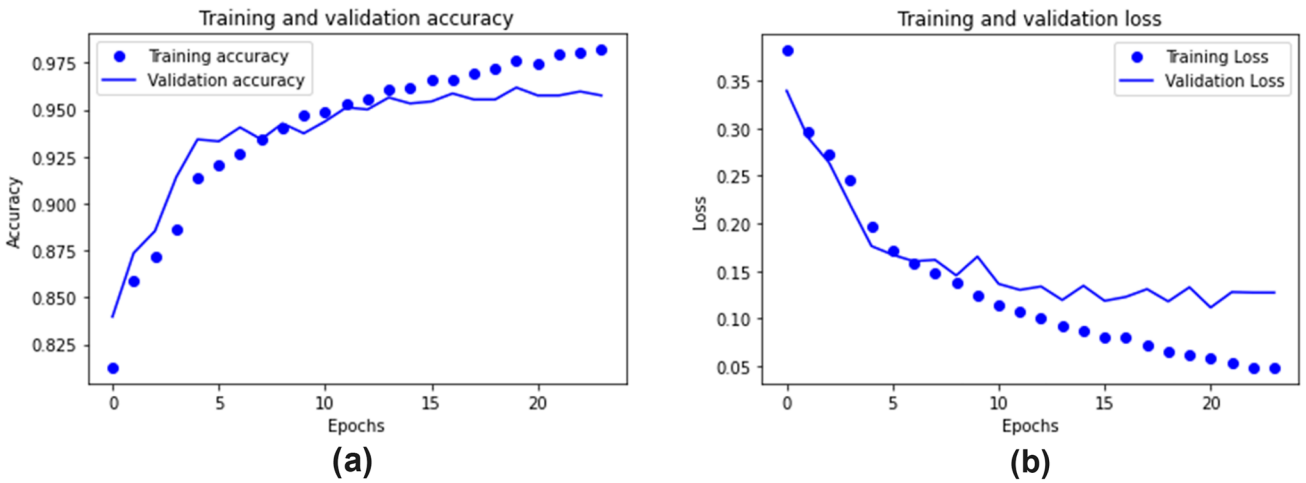

Fig. 8 Accuracy curve and loss curve

From the confusion matrix shown above we can calculate precision, recall and f1-score of our model, whose formulae are shown below. Also the calculated results of our model are illustrated in Table 1.

$$\text{Precision} = \frac{\text{Truepositive}}{\text{Truepositive} + \text{Falsepositive}}$$

$$\text{Recall} = \frac{\text{True positive}}{\text{Truepositive} + \text{Falsenegative}}$$

$$\text{F1score} = 2 \times \frac{\text{Precision} \times \text{recall}}{\text{Precision} + \text{recall}}$$

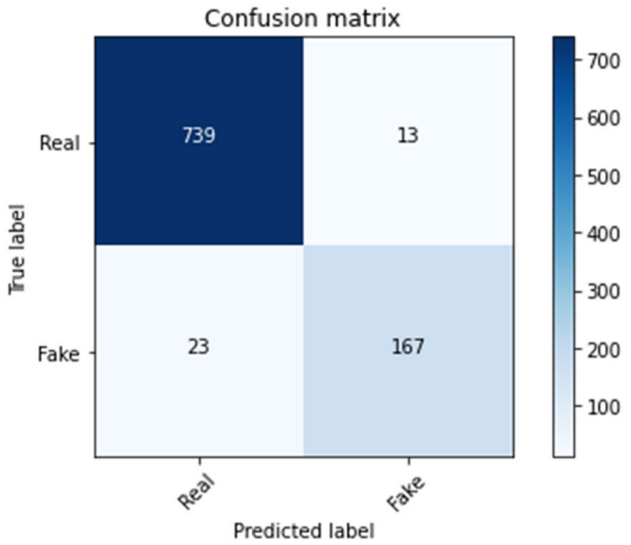

Fig. 9 Confusion matrix

Table 1 Precision, recall and F1 score of our model

Precision	Recall	F1 score
0.9698	0.9827	0.9762

Comparison

We compare the performance of our model with other models that also classify whether an image is tampered or not. Comparisons of results are discussed in Table 2 along with other details of the experiments.

Our model has the smallest size with respect to all other models discussed above and also has better accuracy than all the models except Zhang et al. [17]. Zhang et al. [17] achieved an accuracy of 97.6% and our proposed model achieved an accuracy of 96.18%. Although their accuracy is slightly better than the proposed model, the size of their model is 225 MB whereas our model size is only 96 MB. Their total model parameters are 2.95×10^7, but our model parameters are 8.41×10^6. Therefore the proposed model takes less time and computation resources.

Conclusion

The amount of tampered images we find these days makes us question the information we come across. Digital image forensics is having a tough time dealing with these kinds of fake information. Convolutional neural networks have a

Table 2 Comparison with other models

References	CNN architecture	Dataset used	Network parameters	Performance
Gunawan et al.[a] [12]	2 Convolutional layers, 1 Pooling layer, 1 Fully-connected layer, 1 Two-way Softmax classifier	CASIA v 2.0	2.95×10^7	Accuracy—91.83%
Sudiatmika et al. [13]	VGG-16 (13 Convolutional layers, 5 Pooling layers, 3 Fully-connected layers, 1 Softmax classifier)	CASIA v 2.0	1.44×10^8	Training accuracy—92.2% Validation accuracy—88.46%
Kanwal et al. [14]	NA	CASIA v1.0	NA	Accuracy—88.62%
Doegar et al. [15]	AlexNet (5 Convolutional layers, 3 Pooling layers, 2 Fully connected layers, and 1 Softmax layer)	MICC-F220	–	Accuracy—93.94%
Thakur et al. [16]	6 Convolutional layers, 4 Max pooling layers, 2 Fully connected layers, 1 Softmax layer	CoMoFoD, BOSSBase	–	Validation accuracy-95.97% (CoMoFoD), 94.26% (BOSSBase)
Zhang et al. [17]	2 Convolutional layers, 1 Pooling layer, 1 Fully-connected layer, 1 Two-way Softmax classifier	Milborrow University of Cape Town (MUCT) database	2.95×10^7	Testing AUC—97.6%
Doegar et al. [18]	–	MICC-F220	8.19×10^7	Accuracy— Pre-trained model—90.91%, Fine tuned model—93.18%
Proposed model	2 Convolutional layers, 2 Pooling layers, 1 Fully-connected layer, 1 Two-way Softmax classifier	CASIA v 2.0	8.41×10^6	Validation accuracy—96.18%

[a] https://github.com/agusgun/FakeImageDetector

remarkable performance when it comes to extracting features from images. But CNNs are inclined to learn features from the images rather than finding the signs of tampering.

Hence, to improve the effectiveness we pre-processed the images using error level analysis and then fed them to a CNN. The model can fairly classify between authentic and tampered images as it obtained a validation accuracy of 96.18%.

The main focus of the paper is to identify Fake images from real images in social media. Region of objects which are modified/ tampered is visible from the ELA enhanced images and in our future work we will identify the tampered objects.

Declarations

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Chen J, Kang X, Liu Y, Wang ZJ. Median filtering forensics based on convolutional neural networks. IEEE Signal Process Lett. 2015;22(11):1849–53. https://doi.org/10.1109/LSP.2015.2438008.
2. Huang T, Yuan X. Detection and classification of various image operations using deep learning technology. In: International Conference on Machine Learning and Cybernetics (ICMLC). 2018. 1: 50–55. https://doi.org/10.1109/ICMLC.2018.8526999.
3. Bayar B, Stamm MC. A deep learning approach to universal image manipulation detection using a new convolutional layer. In: 4th ACM Workshop on Information Hiding and Multimedia Security 2016. p. 5–10. https://doi.org/10.1145/2909827.2930786.
4. Zhou P, Han X, Morariu VI, Davis LS. Learning rich features for image manipulation detection. In: IEEE Conference on Computer Vision and Pattern Recognition 2018. p. 1053–1061. https://doi.org/10.1109/CVPR.2018.00116.
5. Salloum R, Ren Y, Kuo CC. Image splicing localization using a multi-task fully convolutional network (MFCN). J Vis Commun Image Represent. 2018;51:201–9. https://doi.org/10.1016/j.jvcir.2018.01.010.
6. Wu Y, Abd-Almageed W, Natarajan P. Busternet: detecting copy-move image forgery with source/target localization. In: European Conference on Computer Vision (ECCV) 2018. p. 168–184.
7. Zhang Y, Goh J, Win LL, Thing VL. Image region forgery detection: a deep learning approach. In: Singapore Cyber-Security Conference (SG-CRC). 2016. p. 1–11.
8. Krawetz N, Solutions HF. A picture's worth. In: Black Hat Briefings USA. 2007. https://www.blackhat.com/presentations/bh-usa-07/Krawetz/Whitepaper/bh-usa-07-krawetz-WP.pdf. Accessed 30 Nov 2020.
9. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD. Back propagation applied to handwritten zip code recognition. Neural Comput. 1989;1(4):541–51. https://doi.org/10.1162/neco.1989.1.4.541.
10. Zeiler MD. Adadelta: an adaptive learning rate method. 2012. arXiv preprint arXiv:1212.5701. Accessed 22 Dec 2012.
11. Sovathana P, Kaggle. casia-dataset. 2018. https://www.kaggle.com/sophatvathana/casia-dataset. Accessed 09 Dec 2020.
12. Gunawan A, Lovenia H, Pramudita A. Deteksi Pemalsuan Gambar dengan ELA dan Deep Learning; 2018. https://doi.org/10.13140/RG.2.2.28571.52006.
13. Sudiatmika IB, Rahman F. Image forgery detection using error level analysis and deep learning. Telkomnika. 2019;17(2):653–9. https://doi.org/10.12928/telkomnika.v17i2.8976.
14. Kanwal N, Girdhar A, Kaur L, Bhullar JS. Detection of digital image forgery using fast fourier transform and local features. In2019 International Conference on Automation, Computational and Technology Management (ICACTM) 2019. p. 262–267. https://doi.org/10.1109/ICACTM.2019.8776709.
15. Doegar A, Dutta M, Gaurav K. CNN based image forgery detection using pre-trained AlexNet model. International Journal of Computational Intelligence & IoT. 2019. 2:1. https://ssrn.com/abstract=3355402. Accessed 01 Feb 2022.
16. Thakur R, Rohilla R. Copy-move forgery detection using residuals and convolutional neural network framework: a novel approach. In 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC) 2019. p. 561–564. https://doi.org/10.1109/PEEIC47157.2019.8976868.
17. Zhang W, Zhao C, Li Y. A novel counterfeit feature extraction technique for exposing face-swap images based on deep learning and error level analysis. Entropy. 2020;22(2):249. https://doi.org/10.3390/e22020249.
18. Doegar A, Hiriyannaiah S, Siddesh GM, Srinivasa KG, Dutta M. Cloud-based fusion of residual exploitation-based convolutional neural network models for image tampering detection in bioinformatics. BioMed Res Int. 2021. https://doi.org/10.1155/2021/5546572.
19. Zhang Z, Zhang Y, Zhou Z, Luo J. Boundary-based image forgery detection by fast shallow cnn. In 2018 24th International Conference on Pattern Recognition (ICPR). 2018. p. 2658–2663. https://doi.org/10.1109/ICPR.2018.8545074.