Neural Processing Letters (2024) 56:112
https://doi.org/10.1007/s11063-024-11448-9

Detection of Image Tampering Using Deep Learning, Error Levels and Noise Residuals

Sunen Chakraborty[1] · Kingshuk Chatterjee[2] · Paramita Dey[3]

Accepted: 17 October 2023 / Published online: 19 March 2024

Abstract
Images once were considered a reliable source of information. However, when photo-editing software started to get noticed it gave rise to illegal activities which is called image tampering. These days we can come across innumerable tampered images across the internet. Software such as Photoshop, GNU Image Manipulation Program, etc. are applied to form tampered images from real ones in just a few minutes. To discover hidden signs of tampering in an image deep learning models are an effective tool than any other methods. Models used in deep learning are capable of extracting intricate features from an image automatically. Here we proposed a combination of traditional handcrafted features along with a deep learning model to differentiate between authentic and tampered images. We have presented a dual-branch Convolutional Neural Network in conjunction with Error Level Analysis and noise residuals from Spatial Rich Model. For our experiment, we utilized the freely accessible CASIA dataset. After training the dual-branch network for 16 epochs, it generated an accuracy of 98.55%. We have also provided a comparative analysis with other previously proposed work in the field of image forgery detection. This hybrid approach proves that deep learning models along with some well-known traditional approaches can provide better results for detecting tampered images.

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

1 Introduction

At present age, countless persons are associated with some form of social networking website or application. In every one of these sites images or videos are shared. This is done either to

✉ Paramita Dey
dey.paramita77@gmail.com

[1] Department of Computer Science and Engineering, Haldia Institute of Technology, Haldia, India

[2] Department of Computer Science and Engineering, Government College of Engineering and Ceramic Technology, Kolkata, India

[3] Department of Information Technology, Government College of Engineering and Ceramic Technology, Kolkata, India

spread information, to make fun of certain situations, or to attract and influence the opinions of the masses to a cause. From the early days of conflict, humans know that information is a powerful tool. We also knew that misinformation can also be a tool if wielded properly. With the surge in the availability of low-cost or free image-editing programs, now misinformation through image tampering can be spread easily to sway the people. Tampered images are not only found in social media but can also be found in other various fields like medical diagnosis, evidence in court, political propaganda, research journals, etc. This leads to a boost in the numbers of fake news which makes it harder to decide whether any information is real or fake.

To detect tampering numerous methods have been suggested, like embedding digital watermarks or signatures [1], splitting an image into blocks to detect copy-move tampering [2], variation of the borders around an object due to image splicing [3], etc. Although these methods are suitable for spotting tampering they have some serious flaws. Most of these techniques work well only if the images are created in a specific manner in case of watermarks or signatures, or tampered in a specific manner in case of copy-move and splicing.

Progress in hardware technology specifically Graphics Processing Units (GPU) and easy access to large volumes of data promotes the research in the domain of deep learning. Deep learning set off a reaction in the fields of computer vision, pattern recognition, image processing, etc. producing results never seen before. Out of various deep learning models when it comes to working with data like images or videos, CNN is more desirable. Researchers usually prefer CNNs because of their ability to draw out complex features from an image automatically at a faster rate.

Although CNNs have such an amazing capability they cannot spot the features necessary to detect the tampering directly as discovered by Chen et al. [4]. Hence, to counteract this issue we used two pre-processing steps. The first one is Error Level Analysis [5], which can be employed to detect tampering when an image is encoded in a lossy format. As the name suggests images of this format keep losing part of its data when resaved again and again. Since tampered objects are added after an image is generated we can use this principle to seek out the tampered regions easily.

Another pre-processing approach is to obtain noise residuals of an image. Noise residuals can be achieved using a definite amount of filters which was designed for a steganalysis technique called the Spatial Rich Model [6]. It is a traditional approach and a highly robust tool that can be employed to detect numerous crucial features in an image.

There are many real-time applications for this method. For example, we can create a website or an application programming interface (API) through which anyone can detect fake images on the web to stop spreading misinformation. Further, this can be modified for video tampering detection as a video is nothing but a collection of consecutive images.

The main contributions of the proposed approach are as follows-

- A dual-branch network model is proposed which can classify between real and tampered images.
- Features from two traditional detection methods are used to train the deep learning model which helps the model to learn more tampering features.
- The proposed model is small and easy to understand and implement. Also, the model works well in environments with limited CPU and memory space.
- Our experiment shows that the proposed model performs better than many other image tampering classification methods.

The remaining part of this paper is arranged in the following manner. Some of the previous methods that employed deep learning for fake image detection were described in “Related

Works". We explain the techniques used in this experiment in "Methodology". "Resources" will briefly describe the dataset and setup needed to conduct our experiment. "Experimental Results" expresses the outcome obtained based on various metrics like precision, recall, and F1-score. Then comparative evaluation and findings from the experiment are displayed in "Comparison and Analysis". Ultimately we conclude our observation in "Conclusion".

2 Related Work

There are already numerous proposed methods where deep learning was applied for detecting tampered images. In this section, we tried to briefly explain a handful of those methods.

The approach proposed by Rao et al. [7] can effectively detect copy-move and image splicing. The first layer of the CNN uses designed filters that can capture the details generated due to tampering. Initially, the images were divided into patches and fed into the CNN to learn the features of tampering. Then, the features were combined in a way that produced the final features for classification. Bayar et al. [8] developed and added a unique layer to their model to eliminate the necessity of pre-processing. It focuses mainly on the association between the pixels and how tampering changes that association in a specific region. The new layer was capable of discovering those changes while ignoring the additional contents of the images.

Rota et al. [9] divided images into patches for the classification that can be used to train large neural networks that can also identify those patches which contain the edges around the altered regions. This method can be implemented further to localize the manipulated regions. Thakur et al. [10] 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.

Xiao et al. [11] employed a pair of CNN to detect image tampering. The first network is used to generate an approximation of tampered regions in an image. The second CNN is used to narrow down and remove the falsely tampered regions generated by the previous CNN. Then a proposed method for adaptive clustering is applied to find the final tampered regions. Zhang et al. [12] applied a sliding window over the images to create patches. Then those patches are utilized by a shallow CNN to detect edges around the tampered regions. They suggested that this process is time-consuming. Hence, they devised a fast shallow CNN that accepts the entire image as input instead of patches and detects edges around the forged areas. This method also works well with low-resolution images.

Wang et al. [13] applied a Mask Region-based CNN (R-CNN) to the images. The Mask R-CNN produces a segmentation mask of tampered areas from the feature maps of the images. Those masks were unrefined in nature. Hence, a Sobel filter is used to further refine them. Bondi et al. [14] developed a method to detect objects that are spliced into an image from a different camera model. Here a CNN is used to extract features from individual patches of an image. The confidence score of each patch is also computed as the result of the CNN is not always reliable. Finally, a clustering algorithm is used on the features and the confidence score to localize those regions where an object is from another camera model.

Zhou et al. [15] proposed a two-stream faster region-based CNN (R-CNN). Here the first stream accepts colour images while the second stream accepts noise images generated by certain SRM filters. The second stream works well even if a forger uses heavy post-processing

on the image. At last, features from both streams are merged to detect the tampering style as well as the tampered area.

Walia et al. [16] also combined features from a CNN and a traditional technique to detect tampered images. In the first branch, images are divided into segments. Then Markov features are generated using discrete cosine transform. In the second branch, local binary patterns of the images were fed into a pre-trained ResNet-18 network to generate feature maps. Finally, for classification features from both the branches are combined and fed into a shallow neural network.

Ding et al. [17] created a dual-branch U-net for image splicing detection. At first images with edge information are created by applying high-pass filters on the sample images. Then sample images are fed into one branch and images with edge information are fed into another branch of the network. Features are extracted from both branches using encoder networks. Those features are then combined and sent into a decoder network to create a rough localization map of the tampered area. Then further post-processing is done to generate the final tampering map.

Niloy et al. [18] also created a two-stream encoder network. Here, the first stream accepts the RGB images and the second stream accepts images generated by using SRM filters developed by Fridrich et al. [6]. Features from both streams are combined and sent to an Atrous Spatial Pyramid Pooling (ASPP) module developed by Chen et al. [19]. This module is used to capture multi-scale features. Then those features are sent to the segmentation head for the final tampering map and to the contrastive learning module to learn the difference between authentic and tampered pixels.

All these works utilized traditional features along with deep learning models for image tampering detection. Most of them chose filters developed for spatial rich model because the noise residuals for a particular pixel is computed from its neighbouring pixels, which makes it easier to detect the signs of tampering. While most of them use larger and heavier models, we try to create a model which is comparatively lighter and easier to implement.

3 Methodology

As we have already mentioned that we employed two pre-processing steps for further enrichment of the learning of our CNN. In the beginning, we produce Error Level Analysis (ELA) images and noise residual images of all the authentic and tampered images. Noise residual images were acquired using 30 high-pass filters that were developed for Spatial Rich Model (SRM). Then, we resize the pictures and include the labels in a corresponding manner. Then, at that point we divide the information into two sets, one is for training and another is for validation purposes. All the steps are summarized in Fig. 1.

3.1 Error Level Analysis (ELA)

Krawetz et al. [5] came up with the concept of ELA. This procedure works well on an image if it was formed using a lossy format, like JPEG. In case an image in JPEG format is altered and resaved in the same format over and over, then it will lose a portion of its data every time. Exploiting this issue, ELA resaves a picture at a pre-determined compression rate e.g. 95%, and then checks the dissimilarity between the original and the resaved image. After generating the ELA of an image, unaltered regions will have nearly equivalent error levels

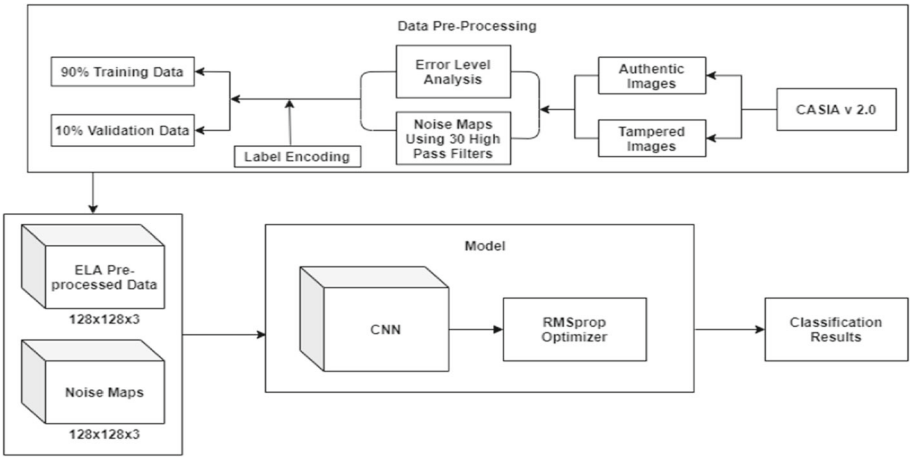

Fig. 1 Summary of the entire procedure

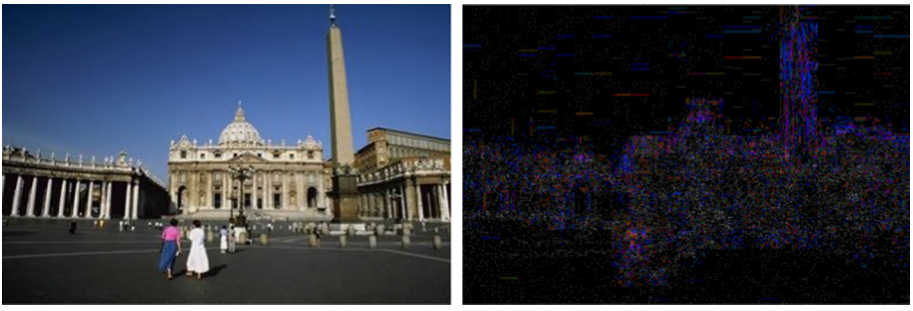

Fig. 2 Authentic image and its ELA

whereas altered areas will have a variation in their error levels. How the ELA can detect tampered regions in an image is shown in Figs. 2 and 3.

From Figs. 2 and 3, we can somewhat understand the working principle behind ELA. In Fig. 2 you can see that authentic images have different error levels, while in Fig. 3 when the authentic image was tampered with and resaved again as a JPEG image the authentic regions have different error levels compared to tampered regions. In Fig. 3 the picture was altered by adding objects from the same as well as some other images, these are the two distinct types of tampering that are generally used by the forgers, which is why we chose ELA in our experiment.

3.2 Noise Residuals from Spatial Rich Model (SRM)

An additional technique that we employed is to extract features from the noise residuals of an image using SRM. The idea of SRM was first proposed by Fridrich et al. [6]. It was developed for steganalysis purposes where a certain number of high-pass filters were applied to extract intense features from the noise residuals of an image. Then those features

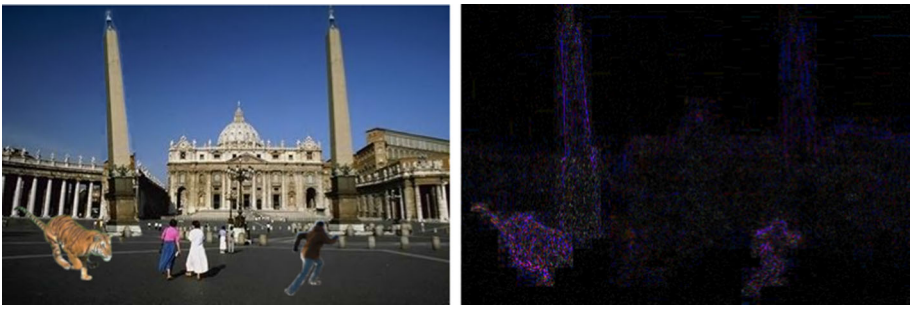

Fig. 3 Tampered image and its ELA

are joined together and fed into ensemble classifiers. It is a custom-made technique that computes the statistics needed to discover certain features from the noise residuals present around the pixels' neighbourhood in an image. Since the residuals are linked with the details of an image, alterations in those details will affect the corresponding residuals. Before deep learning approaches became famous SRM was an ideal choice to detect changes in images.

To transform the images into noise residual maps we applied the actual 30 high-pass filters designed by Fridrich and Kodovský [6]. Each filter is capable of generating noise residuals based around a central pixel. The detailed description of individual filters and their importance is already described by Fridrich et al. [6]. Based on that importance we chose to utilize all of the 30 filters to generate noise residual maps of individual images. All 30 high-pass filters are shown below in Fig. 4.

An example of extracted noise residuals from an image after applying the 30 high-pass filters depicted above is illustrated in Fig. 5 below.

3.3 Convolutional Neural Network (CNN)

The idea of Convolutional Neural Networks was first conceived by LeCun et al. [20]. CNN is a deep learning model which is generally used in research areas concerning visual data like images or videos. The fundamental architecture of a CNN closely resembles that of an artificial neural network. The only basic difference is that it has some additional layers that make it more suitable to detect necessary features from 2-D data formats to create competent outcomes.

3.4 Proposed CNN Architecture

For our experiment, we suggested a dual-branch Convolutional Neural Network that can distinguish between genuine and forged images. ELA images are fed into the first branch and the noise residual images are fed into the second branch. The complete architecture of the proposed dual-branch CNN is shown in Fig. 6. Rest of the attributes of our method is described below.

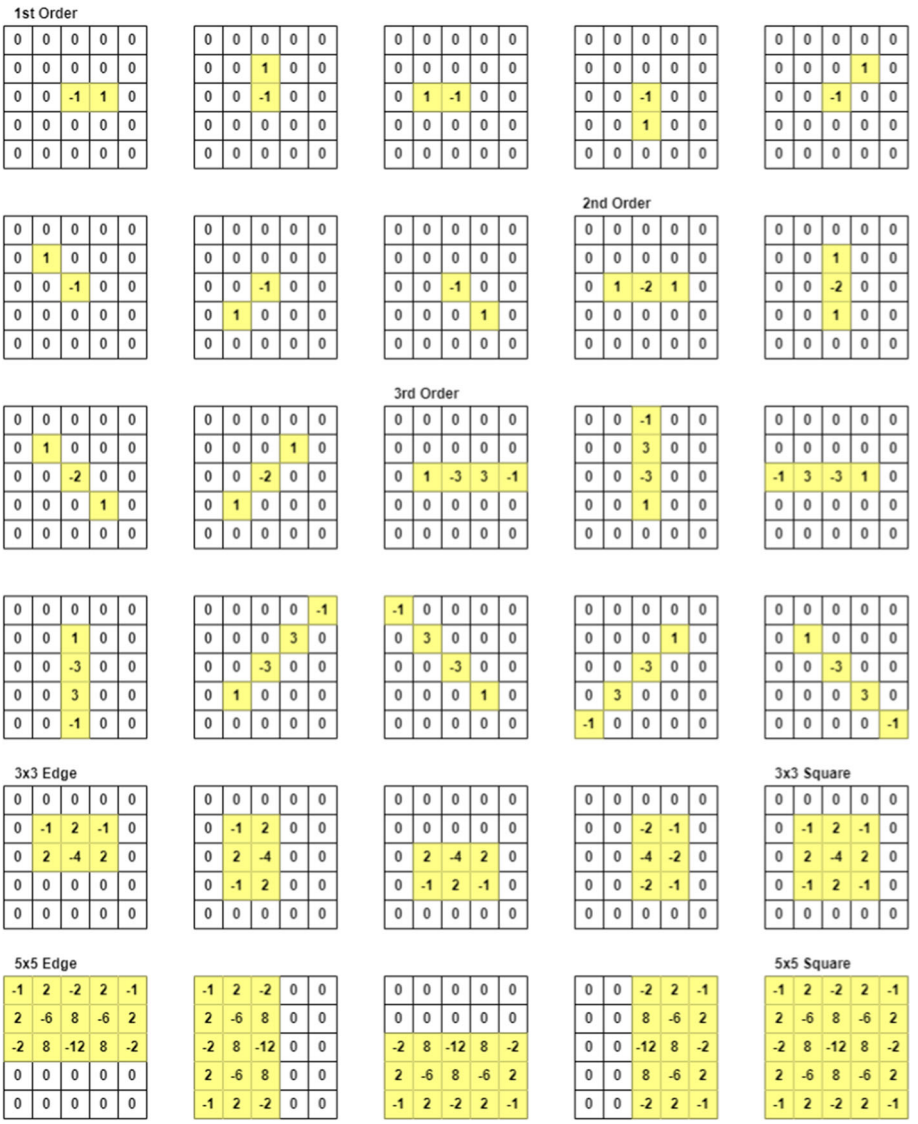

Fig. 4 Thirty high pass SRM Filters

3.4.1 Convolutional Layer

This is the most fundamental and crucial layer of a CNN as the name implied. These layers produce or obtain features from an image by applying convolution operation on it. It is an array of filters/masks/kernels where every filter glides across an image and computes a dot product between the filter and a portion of the image identical to the size of the filter. Every filter is designed to capture only a specific feature. At first, we used two 5×5 filters in both branches but it increased complexity and training time without any significant increase in the

Fig. 5 Noise residuals after using 30 SRM filters

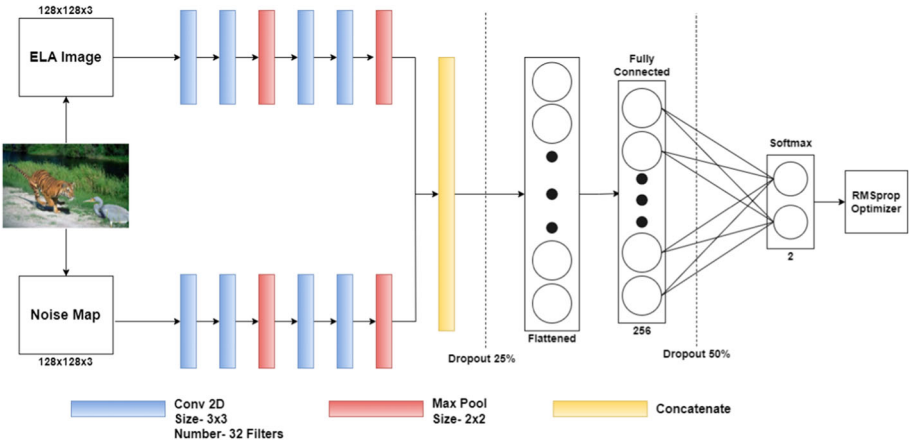

Fig. 6 Architecture of the dual-branch CNN

performance. Then according to the concept proposed by Szegedy et al. [21] for Inception-v2 and v3, we factorize all the 5×5 convolutions into two 3×3 convolutions. This way we decreased the complexity and training time while increasing the performance. Hence, our CNN has four convolutional layers at each branch of size 3×3 and each layer has 32 filters inside it.

3.4.2 Max Pooling Layer

Pooling layers applies downsampling which is used to minimize the scale of the feature maps forwarded by the convolutional or any other previous layers. This further reduces the training time and overall parameters in the model. When max-pooling is applied, only the element that has the maximum value is selected from the section covered by the filter. Thus, only the crucial components of each feature are taken into account. Our model uses two max-pooling layers per branch.

3.4.3 Concatenation Layer

A concatenation layer accepts the output of two or more different layers as inputs and concatenates them along a particular axis. The inputs must have identical dimensions except for the concatenation dimension. As you can see we applied one concatenation layer that combines the output from both the branches.

3.4.4 Fully Connected Layer

These are the final layers of a CNN. It looks exactly like a standard artificial neural network and is used for various classification tasks. 3-Dimensional feature maps obtained from the preceding layers must be flattened into a 1-Dimension format before sending into the fully connected layer. At the end of these layers, there is usually an output layer with a softmax activation function, where the number of the output is equal to the number of classes or labels required for the task of classification. This model uses one fully connected layer and a two-way softmax layer.

3.4.5 Tuning Parameters

In the training phase, we chose "RMSprop" [22] as an optimizer for constantly updating the parameters required to bring down the loss while increasing the performance of the model. Then to reduce the deviation between the actual outcome and the outcome anticipated by the network we used "categorical_crossentropy" as the loss function for our model. This loss function is used for classification tasks when the output is a categorical variable of two or more classes. The formula of the loss function is shown below.

$$\mathrm{CCE} = -\sum_{i=1}^{n} Y_i \log(\widehat{Y}_i)$$

Here, Y_i is the one-hot encoded vector for the true/actual class while the $\widehat{Y}_i$ is the probability distribution of the predicted class. Whenever an image is fed into the network/model along with its true class label the model tries to predict its class. Then using Categorical Cross Entropy (CCE) function the model computes the loss value. This way the model tries to learn and adjust its prediction and gets better over time.

4 Resources

4.1 Setup

The entirety of our experiments is carried out using the Jupyter Notebook accessible through Google Colab. Training of the model is executed over a GPU runtime on Google Colab that allocated a RAM of 12.72 GB and a Disk Space of 68.40 GB.

4.2 Dataset

To train our model with authentic and tampered images we chose the CASIA dataset [23] that can be found on Kaggle. This dataset was first introduced by Dong et al. [24]. It was

Fig. 7 Authentic samples from CASIA v2.0

Fig. 8 Tampered samples from CASIA v2.0

created for the researchers to test their algorithms developed for image forensics in real-world scenarios. Since there were fewer samples in CASIA v1.0 dataset to work with, we chose CASIA v2.0 dataset for our experiment. CASIA v2.0 dataset is made up of 7492 original images and 5124 altered images of different formats. In CASIA v2.0 dataset some of the fake images were altered by applying copy-move forgery while others were tampered using image splicing. These two techniques are essential kinds of tampering employed by the forgers generally, which makes this dataset more fitting for discovering image tampering. Samples of both authentic and tampered images are shown in Figs. 7 and 8 respectively.

5 Experimental Results

We have stated already that at first we convert the images into the ELA images and then we create the noise residual images by applying 30 high pass filters from SRM before feeding them to the CNN. Since ELA works well on images that are encoded using lossy compression, we chose images with a lossy format which is JPEG, and discarded images of other formats. Then our dataset contained a total of 7564 images, out of which 5500 were authentic images and 2064 were tampered images. After generating ELA and noise residuals of those images and adding their labels there are 7564 ELA image samples and 7564 noise residual samples. All the images are changed to a size of 128×128. Then the data is split into 90% for training and 10% for validation. After splitting 6807 ELA images and 6807 noise residual samples were in the training set. While the validation set contained 757 ELA images and 757 noise residual samples. Then the images are fed into the dual-branch neural network for training up to 40 epochs. Performance of the model after training is shown below, where Fig. 9a illustrates the accuracy curve and Fig. 9b illustrates the loss curve.

The training process stopped early at the 16th epoch. It can be clearly seen from the figures above that the model reached a training accuracy of 99.07% and a validation accuracy of 98.55%. Following this we assess our model over the validation set whose confusion matrix is shown in Fig. 10.

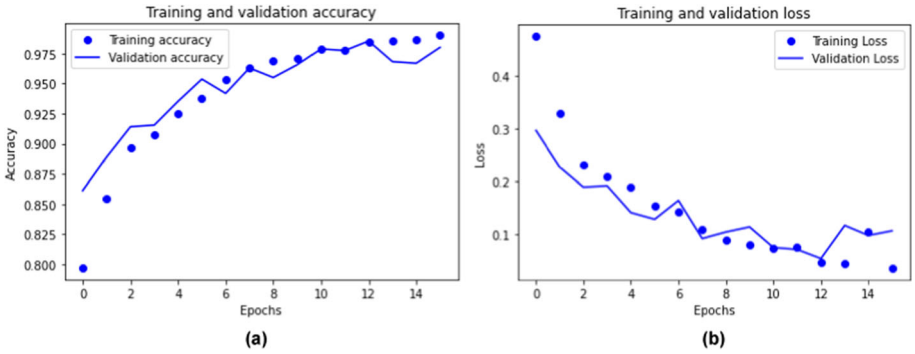

Fig. 9 Graph for accuracy and loss curve

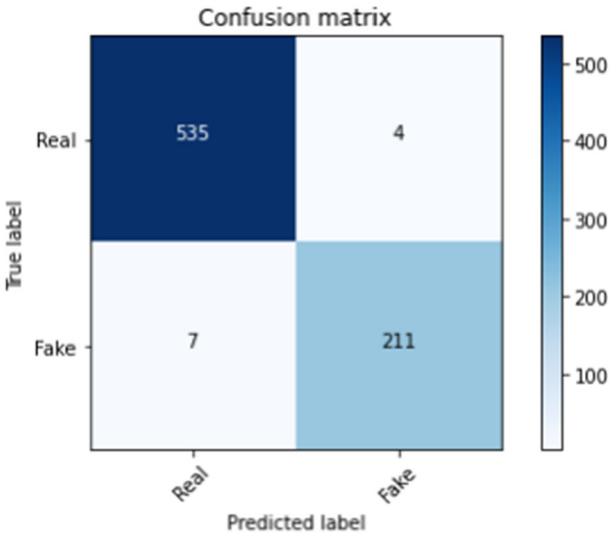

Fig. 10 Confusion matrix

Now we assess the performance of our model using various metrics such as accuracy, precision, recall, and F1 score, the results of the evaluation are shown in Table 1.

We have also generated the Receiver Operating Characteristics (ROC) curve of our model which is shown in Fig. 11. After the generation of the ROC curve we have computed the Area Under the ROC Curve (AUC) score. The AUC score of our model is 0.99.

Table 1 Results under various evaluation metrics

Accuracy	Precision	Recall	F1 score
0.9855	0.9871	0.9926	0.9898

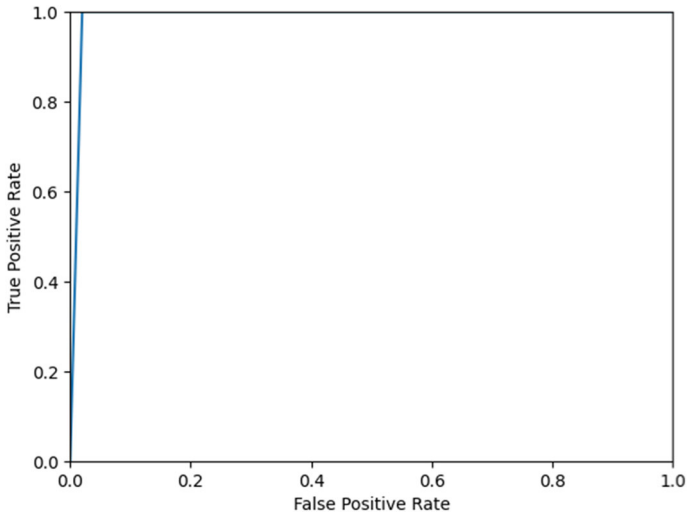

Fig. 11 ROC curve

Table 2 Accuracy of the proposed model with different kernel settings

Method	Accuracy
Two 5 × 5 convolutional layers in both branches	93.89%
Two 5 × 5 convolutional layers in one branch Four 3 × 3 convolutional layers in another branch	96.25%
Four 3 × 3 convolutional layers in both branches	98.55%

Now we have conducted the ablation experiment to demonstrate how the model performs with slight variations. Different variations and their corresponding accuracy are shown in Table 2.

From the above table, it is clear that using four 3 × 3 convolutional layers instead of two 5 × 5 convolutional layers can greatly increase the accuracy of the proposed model. This also proves the concept proposed in the work of Szegedy et al. [21] which states that factorizing a large convolutional layer into smaller layers can greatly increase the performance of a model while reducing the training time.

6 Comparison and Analysis

In this section, we try to measure the functioning of our neural network in contrast with other previously proposed techniques that likewise identify image tampering in a comparable manner. A comparative analysis is displayed in Table 3 alongside additional attributes of the experiments.

Rao et al. [7] modify the weights of the first layer of their CNN with the same 30 high-pass filters used in the SRM. They split up the images into patches and used the CNN to obtain

Table 3 Comparison with pre-existing methods

References	Architecture of the CNN	Utilized dataset	Performance metrics
Rao et al. [7], 2016	1 30 high-pass SRM filter layer, 7 conv. layers, 2 max pool layers, 1 fully-connected layer with a two-way softmax classifier	CASIA v1.0, CASIA v2.0, Columbia gray DVMM	Accuracy-98.04% (CASIA v1.0), 97.83% (CASIA v2.0), 96.38% (DVMM)
Sudiatmika et al. [25], 2019	VGG-16 (13 Conv. layers, 5 Max Pool layers, 3 fully-connected layers, 1 Softmax Classifier)	CASIA v 2.0	Training accuracy—92.2% Validation accuracy —88.46%
Zhang et al. [26], 2020	2 Conv. layers, 1 max pool layer, 1 fully-connected layer, 1 two-way softmax classifier	Milborrow University of Cape Town (MUCT) database	Testing AUC—97.6%
Goel et al. [27], 2021	Dual branch CNN [(1st Branch—3 conv. layers, 3 max pool layers, 1 Zero-padding layer, 2nd branch —3 conv. layers, 3 max pool layers) 1 concat layer, 1 global-max pool layer, 2 fully-connected layers, 1 sigmoid classifier]	MICC-F2000	Accuracy—96%
Singh et al. [28], 2021	1 16 high-pass filter layer, 1 max pool layer, 1 average pool layer, 3 fully-connected layers, 1 softmax classifier	CASIA 2.0, Self made dataset from twitter images	Accuracy—92.3% (CASIA 2.0), 81.3% (self made dataset)
Chakraborty et al. [30], 2022	2 Conv. layers, 2 max pool layers, 1 fully-connected layer, 1 Two-way softmax classifier	CASIA 2.0	Accuracy—96.18%
Proposed	Dual branch CNN [(both branches—4 conv. layers, 2 max pool layers) 1 concat layer, 1 fully-connected layer, 1 two-way softmax classifier	CASIA v 2.0	Accuracy—98.55%

the necessary features from the patches, and then said features are combined together using a proposed fusion technique so that a classifier can utilize them for detecting image tampering.

Sudiatmika et al. [25] first converted the images into error-level analysis images and then fed them into a VGG-16 network. Because of using a very deep neural network, it took more time to train and obtain a fair result. Also, bigger networks are prone to a problem known as vanishing gradient which is why its performance is lower than other models.

Zhang et al. [26] used their model to detect images generated using DeepFake. They also employed error-level analysis before loading the images to the CNN. Although they achieved quite well with such a small network, their network size is 225 MB while ours is only 105.7 MB. Also, their overall model parameters are 2.95×10^7 whereas ours is 1.38×10^7, which means our model takes less time and resources. This situation occurs because they used 5×5 filters which needed 2.78 times further computations than 3×3 filters as proposed by Szegedy et al. [21].

Goel et al. [27] also used a dual-branch CNN to capture features from tampered images. Their technique was only to detect copy-move tampering and not image splicing. They didn't use any specific pre-processing that can identify the hidden signs of tampering apart from resizing and standardizing the images before putting them into the CNN.

To detect image forgery, Singh et al. [28] also used high-pass filters. The first layer of their CNN contains filters to extract the noise features from the images. Then they used two pooling layers. After which they send the features in two different directions. First, the features are sent into three dense and a softmax layer to classify whether the image is fake or real. Also, gradients are extracted from the feature map of the last pooling layer. These gradients are used to generate a map that roughly localizes the area of tampering as proposed by Selvaraju et al. [29].

Chakraborty et al. [30] also utilized error-level analysis to detect tampering in images. After producing the error-level image of the individual image, all the images are fed into a CNN. Although their model size and model parameters are less than ours their accuracy is also less. It is because the ELA of an image is generated by comparing individual pixels between original and resaved images. But noise residuals of an image are generated from the neighbouring pixels of a particular pixel. That is why our model has combined features from both approaches.

From all the above research work it is evident that features extracted from either ELA or Noise residuals can help a CNN to detect image tampering to a great extent. This is why we have chosen to incorporate features from both the traditional techniques to train our model. As we can see because of this our model can achieve even higher accuracy. While ELA can extract features from tampered images, it has a drawback. If a forger uses heavy post-processing on the image or uses a lossless format like PNG, then ELA can give uncertain results. Hence, we have also utilized noise residuals from SRM. It basically focuses on the noise distributions in an image which can help our model if features from ELA were insufficient. The advantages of noise residuals are also mentioned in the work of Zhou et al. [15]. To further reduce the error we can utilize resource intensive deeper models.

7 Conclusion

To differentiate between authentic or tampered images, here we have proposed a dual-branch convolutional neural network. We have employed image pre-processing in combination with neural networks that can further enhance performance. In the beginning, we evaluate the

error level analysis of the images and feed it to one branch of our network. Simultaneously, we compute the noise residuals using 30 high-pass filters designed for steganalysis and feed it to another branch of the network.

We also displayed that this approach can perform better than many other pre-existing techniques. Our model achieved this accuracy with less number of parameters and has a small size hence it requires less time and computational resources. Though our model shows great results for classification between real and tampered images, it cannot detect or localize the area of tampering. This will be included in future work, in which we will try to create models that are small and simple, yet effective enough to classify and localize tampering in images.

Author contributions All authors wrote the main manuscript text and reviewed the manuscript.

Declarations

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

References

1. Boland FM, O'Ruanaidh JJ, Dautzenberg C (1995) Watermarking digital images for copyright protection. In: Fifth international conference on image processing and its applications. p 326–330. https://doi.org/10.1049/cp:19950674
2. Popescu AC, Farid H (2004) Exposing digital forgeries by detecting duplicated image regions. https://digitalcommons.dartmouth.edu/cs_tr/254/. Accessed on 07 Sept 2021
3. Fang Z, Wang S, Zhang X (2010) Image splicing detection using color edge inconsistency. In: 2010 international conference on multimedia information networking and security. p 923–926. https://doi.org/10.1109/MINES.2010.196
4. Chen J, Kang X, Liu Y, Wang ZJ (2015) Median filtering forensics based on convolutional neural networks. IEEE Signal Process Lett 22(11):1849–1853. https://doi.org/10.1109/LSP.2015.2438008
5. Krawetz N, Solutions HF (2007) A picture's worth. In: Black Hat Briefings USA. https://www.blackhat.com/presentations/bh-usa-07/Krawetz/Whitepaper/bh-usa-07-krawetz-WP.pdf. Accessed on 07 Sep t2021
6. Fridrich J, Kodovsky J (2012) Rich models for steganalysis of digital images. IEEE Trans Inf Forensics Secur 7(3):868–882. https://doi.org/10.1109/TIFS.2012.2190402
7. Rao Y, Ni J (2016) A deep learning approach to detection of splicing and copy-move forgeries in images. In: 2016 IEEE international workshop on information forensics and security (WIFS). p 1–6. https://doi.org/10.1109/WIFS.2016.7823911
8. Bayar B, Stamm MC (2016) 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. p 5–10. https://doi.org/10.1145/2909827.2930786
9. Rota P, Sangineto E, Conotter V, Pramerdorfer C (2016) Bad teacher or unruly student: Can deep learning say something in image forensics analysis?. In: 2016 23rd international conference on pattern recognition (ICPR). p 2503–2508.https://doi.org/10.1109/ICPR.2016.7900012
10. Thakur R, Rohilla R (2019) 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). p 561–564. https://doi.org/10.1109/PEEIC47157.2019.8976868

11. Xiao B, Wei Y, Bi X, Li W, Ma J (2020) Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering. Inf Sci 511:172–191. https://doi.org/10.1016/j.ins.2019.09.038
12. Zhang Z, Zhang Y, Zhou Z, Luo J (2018) Boundary-based image forgery detection by fast shallow cnn. In: 2018 24th international conference on pattern recognition (ICPR). p 2658–2663. IEEE. https://doi.org/10.1109/ICPR.2018.8545074
13. Wang X, Wang H, Niu S, Zhang J (2019) Detection and localization of image forgeries using improved mask regional convolutional neural network. Math Biosci Eng 16(5):4581–4593. https://doi.org/10.3934/mbe.2019229
14. Bondi L, Lameri S, Güera D, Bestagini P, Delp EJ, Tubaro S (2017) Tampering Detection and localization through clustering of camera-based CNN features. In: 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW). p 1855–1864. IEEE. https://doi.org/10.1109/CVPRW.2017.232
15. Zhou P, Han X, Morariu VI, Davis LS (2018) Learning rich features for image manipulation detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p 1053–1061. https://doi.org/10.1109/CVPR.2018.00116
16. Walia S, Kumar K, Kumar M, Gao XZ (2021) Fusion of handcrafted and deep features for forgery detection in digital images. IEEE Access 12(9):99742–99755
17. Ding H, Chen L, Tao Q, Fu Z, Dong L, Cui X (2023) DCU-Net: a dual-channel U-shaped network for image splicing forgery detection. Neural Comput Appl 35(7):5015–5031
18. Niloy FF, Bhaumik KK, Woo SS (2023) CFL-Net: image forgery localization using contrastive learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision 2023. p 4642–4651. https://openaccess.thecvf.com/content/WACV2023/html/Niloy_CFL-Net_Image_Forgery_Localization_Using_Contrastive_Learning_WACV_2023_paper.html. Accessed on 10 June 2023
19. Chen LC, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587. https://arxiv.org/pdf/1706.05587.pdf. Accessed on 10 June 2023
20. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551. https://doi.org/10.1162/neco.1989.1.4.541
21. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognitio. p 2818–2826. https://doi.org/10.1109/CVPR.2016.308
22. Tieleman T, Hinton G (2012) Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: neural networks for machine learning. vol 4(2), p. 26–31
23. Sovathana P (2018) Kaggle. Casia-dataset https://www.kaggle.com/sophatvathana/casia-dataset. Accessed on 03 July 2021
24. Dong J, Wang W, Tan T (2013) Casia image tampering detection evaluation database. In: 2013 IEEE China Summit and International Conference on Signal and Information Processing. p 422–426. https://doi.org/10.1109/ChinaSIP.2013.6625374
25. Sudiatmika IB, Rahman F (2019) Image forgery detection using error level analysis and deep learning. Telkomnika. 17(2):653–659. https://doi.org/10.12928/telkomnika.v17i2.8976
26. Zhang W, Zhao C, Li Y (2020) A novel counterfeit feature extraction technique for exposing face-swap images based on deep learning and error level analysis. Entropy. 22(2):249. https://doi.org/10.3390/e22020249
27. Goel N, Kaur S, Bala R (2021) Dual branch convolutional neural network for copy move forgery detection. IET Image Proc 15(3):656–665. https://doi.org/10.1049/ipr2.12051
28. Singh B, Sharma DK (2021) Image forgery over social media platforms—a deep learning approach for its detection and localization. In: 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom). p 705–709. IEEE
29. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. p 618–626. https://doi.org/10.1109/ICCV.2017.74
30. Chakraborty S, Chatterjee K, Dey P (2022) Discovering tampered image in social media using ELA and deep learning. SN Comput Sci 3(5):1–8. https://doi.org/10.1007/s42979-022-01311-w