Neural Processing Letters (2022) 54:75–100
https://doi.org/10.1007/s11063-021-10620-9

Detection of Copy-Move Forgery in Digital Image Using Multi-scale, Multi-stage Deep Learning Model

Ankit Kumar Jaiswal[1] · **Rajeev Srivastava**[1]

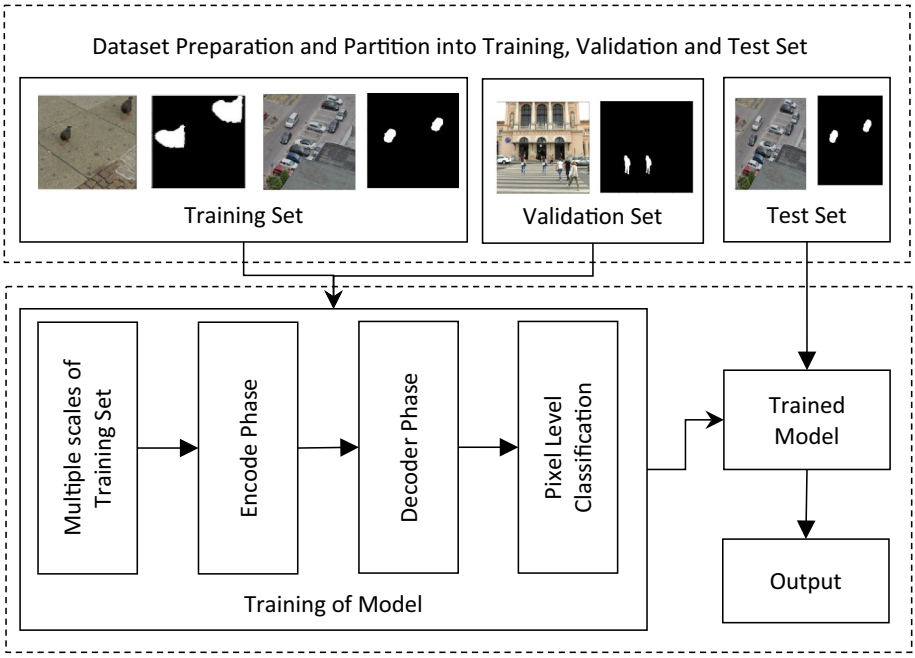
Accepted: 3 August 2021 / Published online: 12 August 2021

Abstract
Images are an important source of information and copy-move forgery (CMF) is one of the vicious forgery attacks. Its objective is to conceal sensitive information from the image. Hence, authentication of an image from human eyes become arduous. Reported techniques in literature for detection of CMF are suffering from the limitations of geometric transformations of forged region and computation cost. In this paper, a deep learning CNN model is developed using multi-scale input with multiple stages of convolutional layers. These layers are divided into two blocks i.e. encode and decoder. In encoder block, extracted feature maps from convolutional layers of multiple stages are combined and down sampled. Similarly, in decoder block extracted feature maps are combined and up sampled. A sigmoid activation function is used to classify pixels into forged or non-forged using the final feature map. To validate the model two different publicly available datasets are used. The performance of the proposed model is compared with state-of-the-art methods which show that the presented data-driven approach is better.

✉ Ankit Kumar Jaiswal
ankitkrjaiswal.rs.cse17@iitbhu.ac.in

[1] Computing and Vision Lab, Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, Uttar Pradesh 221005, India

Graphic Abstract



Keywords Image forensics · Digital image forgery · Copy-move forgery · Image segmentation · Deep learning

1 Introduction

Image forgery is a procedure of creating manipulated images for concealing the information. The first forged image was developed as early as the 1840s. And digital image forgery is not much different, the powerful software like Adobe Photoshop, Coral Paint shop contributes very well in the making of tampered images. Image forgery is performed in two manners. One is image splicing in which the region of an image is replaced by the region of a different image. Another is a copy-move forgery (CMF), this is a procedure of tampering with the images by replacing the portion of an image with a patch of same the image. This type of forgery is used frequently by forgoers because of the similar texture and patterns tampered and non-tampered area of the image. Hiding the information of an image is the main goal to be acquired through CMF. There are two major bifurcations in the forgery detection scheme, namely active and passive. The active techniques involve the extraction of information from the images having add-ons like digital signature, watermarking [1–3]. While passive techniques involve information extraction for image alone with no extraneous information (i.e. they don't need prior information about the image).

Copy-move forgery is one of the major tampering techniques. In this, the duplicate patch is placed over the targeted area with or without any image processing operation. If duplication is done without any operation, patches will be identical but if duplication is

done with any operation patches become un-identical. This is performed either to enhance the visual content of the image or to hide the essential information. The replaced region is from the same image, therefore, the texture and pattern of the replaced patch may same as that of the original image. Hence it is nearly impossible to recognize with the naked human eye (see Fig. 1). However, images are often used in courtrooms for evidence and social media for sharing. Hence, they are also known as information currency. So, to establish the veracity of an image and to identify the authenticity of an image forensic tool is needed, with this digital image forensic comes into the picture. Image Forensic Analysis is an application of image science where the image or contents of an image are to be interpreted by domain expertise. Image Forensic Analysis includes photographic comparison, content analysis and image authentication.

Copy-Move Forgery Detection (CMFD) is one of the passive forgery detection techniques. The first CMFD technique was given by Fridrich et al. [4] in 2003. This is a statistical-based method in which the image is divided into patches and patches are matched together by two different methods. One is an exact match, and another is a robust match. However, this method works well when no operation is performed over the duplicated region. But this method fails with any type of geometrical transformation of divided block e.g. rotation (when rotation angle is more than 5°) and scaling. After this, a lot of methods came based on block-matching during the last 2 decades [5–10]. Except for these block-based method key-point based methods are also published in the literature [11–13]. But the problem with these methods is these methods have poor performance when duplicated regions are very small. Deep learning CNN models are gaining huge momentum in almost every application [14–17] of image processing and computer vision. From 2018 to 2020, a lot of CMFD methods have been given using deep learning methods [18–20]. But they

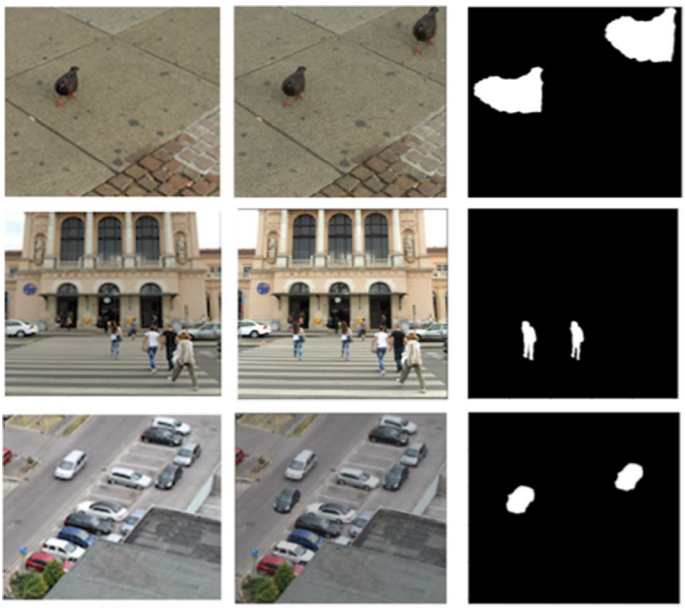


Fig. 1 Examples of Copy Move Forgery, **a** Original Image, **b** Forged Image, **c** Ground Truth mask of Forged Image

are either only classification methods (i.e. image analysis not content) or have poor performance with geometrical transformation. The deep learning networks are highly inspired by human networks that are biological neurons, which constitute multiple nonlinear layers for processing of simple object parallelly. A scale robust deep learning classification technique is given by Ryo Takahashi et al. [21].

The concept of the multi-stage and multi-scale image has been used here to develop a deep learning-based CMFD technique. In this paper, we have developed a deep learning CNN network which has two different part one is an encoder and another is the decoder. The encoder has multiple layers of convolution, rectified linear unit (ReLU), pooling and batch-normalization while the decoder has multiple layers of convolution, ReLU, batch normalization and up-sampling block. The need for a decoder block is to localize the duplicate region in the forged image. The major contributions that have been made in this paper are-

- Proposed deep learning-based CMFD technique using multi-scale input image with a multi-stage convolutional network to overcome the challenge of scale invariant.
- Comparison of the literature of three different types of CMFD schemes i.e. block-based, keypoint based and data-driven approaches.
- Quantitative and visual result analysis of the proposed model on two different publicly available datasets using different performance measures.
- The result produced by the proposed model is compared with other state-of-the-art techniques.

The remains of the paper are organized in the following sections: Section 2 gives a brief literature review of algorithms used for detection of copy-move forgery. Section 3 explains the proposed model for CMFD with its subsections of architecture and training. In Sect. 4, performance evaluation with the details of datasets has been explained and the paper is concluded in the fifth section.

2 Related Works

From 2003 to now a lot of methods for CMFD has been published in conferences and journals. These methods can be divided into three types one is block-based methods, the second is key-point feature extraction-based methods and the third is data-driven approaches (i.e. machine learning and deep learning techniques). Literatures reported in journals and reputed conferences are given in the below paragraphs based on types of methods.

The first one is block-based techniques. The major steps that performed in the block-based matching technique are- block division, feature extraction and then feature matching as shown in Fig. 2. Fridrich et al. [4] was the first who proposed a block-based technique for CMFD by two different matchings. One is exact matching, and another is robust. In this method, the image is extracted into $b \times b$ sized overlapping patches and each patch are sorted into lexicographical order. Two or more identical patches are saved as duplicated regions in exact matching while the shift vector is used in robust matching. No publicly available dataset has been used for validation purpose. Also, this method fails when any type of geometric transform has been performed over the duplicated region. Weiqi et al. [6] proposed another method based on block matching. In this method, three colour features and four statistical features are extracted and then sorted in lexicographical order. Identical

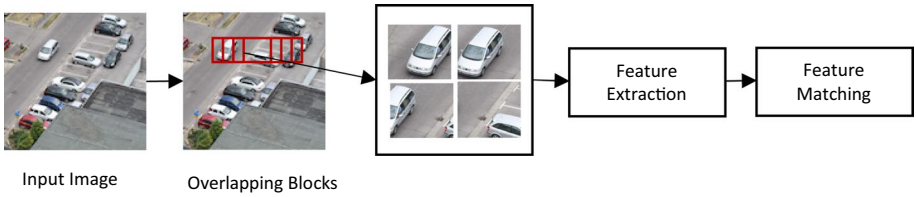


Fig. 2 Steps involved in black-based copy-move forgery detection schemes

feature blocks are saved as duplicated regions. Here also, no publicly available dataset has been used for validation purpose. This method also fails with the geometric transformation of duplicated regions. Mahadian [8] proposed another method using blur moment features. In this method, the image is first divided into overlapping blocks and then 24 blur moment features have been extracted from each overlapping block. Then principal component analysis (PCA) is applied to features and then features are sorted using a K-dimensional tree. This method works well in the case of additive noise, but the computation cost of the method is very high. This method is not robust for geometric transformation. Mahmood et al. [22] proposed a method using stationary wavelet transform (SWT) and local binary patterns (LBP). In this method image is first transformed into the SWT then the image is divided into circular overlapped blocks. Then LBP feature descriptors are extracted from overlapped blocks. Then these features are matched using Euclidian distance. A similar method is proposed in Srivastava et al. [23] using mean features of SWT. This method first transforms the image into SWT and then extract mean features block-wise. These feature vectors are matched using lexicographical sorting of the feature vector. Shift vector is used here to ignore neighbourhood blocks. The problem with these methods is that they don't work with affine transformation. Except these, [7, 8, 22–27, 27] methods are also based on block matching based. In these methods, the image is divided into blocks and the features are extracted from blocks and extracted features are matched together using different matching techniques. Problems with these techniques are- computation cost is very high and not robust with geometrical transformation.

The second type of CMFD is keypoint feature-based techniques. In this technique major steps performed are- preprocessing, keypoint feature extraction from the image and feature matching as shown in Fig. 3. Amerini et al. [11] proposed a keypoint based CMFD technique in 2011. In this method, scale-invariant feature transform (SIFT) keypoints are extracted from the image of 180 features then instead of using direct Euclidian distance a ration of distance pair has been calculated and made a cluster of nearest neighbours using generalized 2NN for matching the keypoints in the image and detected the duplicated

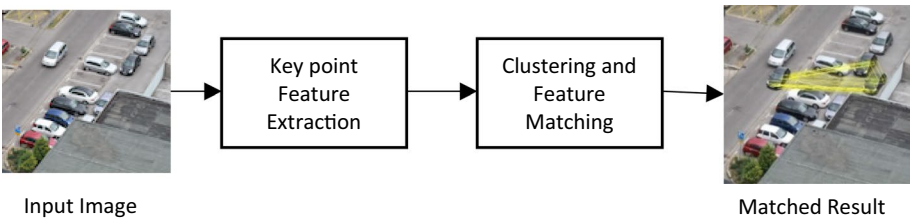


Fig. 3 Steps involved in keypoint matching based copy-move forgery detection techniques

regions in the image. The problem with this technique is that it gives lower accuracy result in the case of small-duplicated regions. Yang et al. [13] Copy-Move Forgery Detection Based on Hybrid Features proposed a method using SIFT and KAZE keypoint features. In this method, hybrid features are extracted from the image and then are matched with the gNN matching technique. Although it works with some geometrical transform of the duplicated region but fails with smooth images. A lot of hybrid approaches are reported in the literature by combining block-based and keypoint features based [28, 29, 30]. Mehta et al. [29] proposed a hybrid approach based on DCT and ORB feature set. DCT feature set is extracted from blocks of the image while ORB feature is keypoint based feature set. DCT feature set is matched using Euclidian distance and ORB feature set are matched using k-NN based hamming distance. These all have either problem of computation cost or problem of performance.

The third type of CMFD approaches is data-driven based. In this method some training data are used to train a classification model and using this model forged image can be predicted as forged or original. Also, some of the data-driven approaches use ground truth mask for training purpose and this type of model localize the duplicate region in the forged image. Liu [19] proposed a data-driven based deep learning approach for feature extraction from keypoints. Then k-NN is performed on extracted features for feature mapping which is used for matching. Another approach of deep learning by Wu et al. [31]. In this technique convolution layers are used to extract features then the correlation is performed on point-wise extracted features then these features are deconvoluted by upsampling to localize the result. Elaskily et al. [18] proposed another deep learning-based CMFD. This method is a classification of an image into forged and original not the localization of the forged region (i.e. image-wise classification is performed instead of pixel-level classification). Another deep learning approach is given by Chen et al. [32]. The given method uses VGG16 architecture to extract features from the image. This method localizes the forged region from the manipulated image. The major limitation of the method is, it doesn't provide the accurate localization of forged region. Problems with other approaches are performance and robustness with geometrical transformation.

Based on the above literature, we have identified some points of the reported literature. These points are summarized in below Table 1. In Table 1 A, S and R represent the Availability of the dataset, the number of images in the dataset and the resolution per image respectively.

From the above-mentioned table, it can be summarized that the method should be robust to transformation, computation cost should be low for image-level detection as well as pixel-level analysis.

3 Proposed Model

This section gives the architecture of the proposed model for CMFD based on the deep learning CNN model. As mentioned above most important challenges in this problem are the geometric transformation of manipulated objects. In case copied objects are scaled or rotated and pasted in the image, most of the existing techniques are unable to detect the manipulated region. Therefore, a method is required that should be scale and rotation invariant. Deep learning CNN models are gaining huge momentum in almost every application of image processing and computer vision. This motivated us to develop the CMFD technique using the deep convolutional network (Conv-Net). To the best of our knowledge,

Table 1 Related works and their comparison on different parameters

Method type	Method	Features	Matching technique	Dataset			Evaluation metrics	Robustness against transformations	Remark
				A	S	R			
Block-Based	[4]	DCT	Lexicographical Sort	No	–	–	–	Doesn't work with transformation	Computation Cost is very high
	[6]	Colour and Statistical	Lexicographical Sort	No	100	300×400	Accuracy, FNR	Doesn't work with transformation	Lower computation than previous
	[22]	SWT and LBP	Euclidian Distance	Yes	10,400	768×512	TDR, FDR	Scale-Invariant	Lower Computation than previous
Keypoint Based	[11]	SIFT	Nearest Neighbour Clustering	Yes	2000	800×600	TPR and FPR	Scale-Invariant	Poor Performance in case of the small duplicated region
	[13]	SIFT and KAZE	Multiple copied feature matching	Yes	–	3000×2300	P, R, F1	Rotation and Scale Invariant	Poor performance with smooth images
Data-Driven	[31]	Deep Features	Classification Pixel wise	Yes	> 10,000	256×256	P, R, F1	Robust against affine transformation, blurring and JPEG compression	Performance is not satisfactory
	[18]	Deep Features	Classification Image wise	Yes	> 3000	< 3888×2592	A, TPR, FPR	–	Only Image-Level Classification

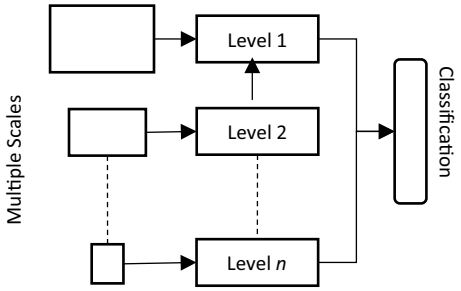


Fig. 4 Example of Multi-Scale Network

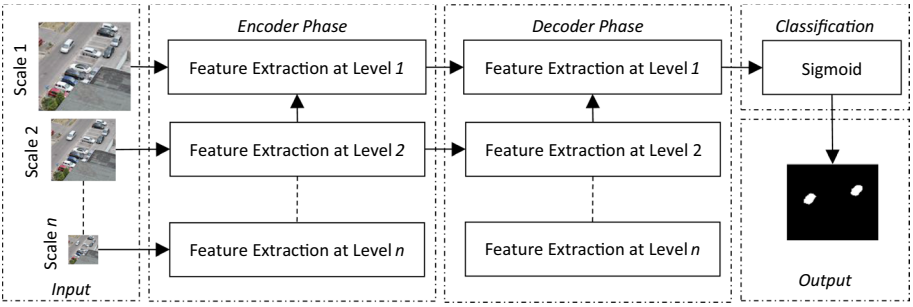


Fig. 5 Block-diagram of the proposed method

none of the deep learning approaches exists for CMFD that is scaling invariant. The proposed architecture is purely designed for the segmentation of copied and pasted object to the same image. Considering the challenges of the existing state-of-the-art technique, the proposed model is developed using multi-scale input of image as shown in Fig. 5. To tackle the challenge of scaling, the concept of multiple scales input of image is taken from [21] and features are extracted at multiple levels which gives an advantage of scaling robustness. These multiple scales are used to extract features at multiple levels. Then from different levels of different layers, these features are concatenated with the first level of different pooled layers (Fig. 4).

3.1 Architecture

This architecture has an encoder network and corresponding to this encoder network, a decoder network exists. Followed by decoder network a sigmoid activation function is there for pixel-wise classification. In this way, the whole architecture is divided into three phases first is the encoder phase second is the decoder phase and the third is the classification phase as shown in Fig. 5. These three phases are explained below subsections:

3.1.1 Encoder Phase

First input images are scaled (half-sampled) multiple times for multiple levels of the architecture. At every level of the model scaled image is taken as input. The proposed model takes an input image of dimension 256×256 and this input image is half-sampled to the dimension of 16×16. In this way, five levels are used in the given model.

The scaled image is then passed through the convolution layer, and batch normalization layer. Then elementwise rectified linear non-linearity (ReLU) activation function ($\max[0, x]$) is applied over normalized feature space. At the first level of the model the activated feature space is again down-sampled i.e. max-pooling is applied over 2×2 patches. However, max pooling is not beneficial for object segmentation when input images are much scaled, in the proposed method multiple scaled are used and higher dimension feature space are used automatically. Max-pooling is also important as max-pooling analyses feature space and extract principal components only from higher features space and convert higher feature space into lower feature space. The lower feature space takes less memory for computation than the higher feature space. Then this max-pooled layer is concatenated with activated feature space of similar dimension of next level as shown in Fig. 5. These layers are continued till their lowest scaled input dimension. Also, at each convolution layer from beginning to end of the encoder phase number of filters are increasing. For every convolution layer, stride value is taken as one and padding is taken as 'same' i.e. feature space should be covered fully and output dimension should be the same as the input dimension.

3.1.2 Decoder Phase

After the encoder phase of the model, the dimension of the output feature gets smaller which is not appropriate for the pixel-wise localization of the manipulated region. To localize the forged region in tampered image training of the model is done with a ground truth segmented image of the input image. The dimension of the output feature space should be sufficient for visualization as well as to segment the manipulated object. Therefore, the output feature of the encoder phase needs to be upsampled. So, corresponding to each max-pooling layer of the encoder phase there is one upsampling layer is added in the decoder phase. This upsampling is also done using a 2×2 window size. The output feature of the upsampling layer is then convoluted with the number of filters. The convoluted features are batch normalized and then activated through the ReLU activation function. From every level of activated feature space is concatenated with the first level of corresponding output of upsampled layer as shown in Fig. 6. This process is continued until the dimension of the output feature is not matches with the dimension of the input image. Similar to the encoder phase, for every convolution layer of the decoder phase stride value is taken as one and padding is taken as 'same' (Fig. 7).

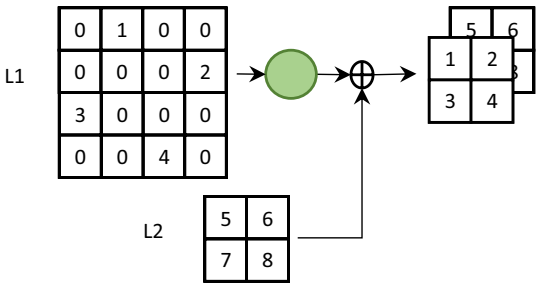


Fig. 6 An Illustration of max-pooling of activated feature space and then the concatenation of another level feature space with first level feature space

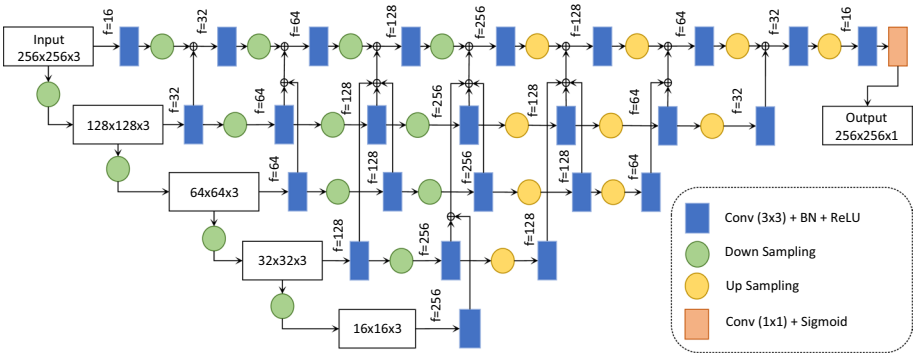


Fig. 7 Architecture of proposed model for copy-move forgery detection using deep learning CNN model

3.1.3 Classification Phase

The decoder phase of the model gives higher depth output feature space but the same height and width as the input image. To train the classifier we need a single depth feature space whose height and width should be the same as the input ground truth image. Thus, higher depth feature space is convoluted with a 1×1 kernel size filter with 'same' padding and stride value 'one'. This convoluted output feature is then passed through the sigmoid activation function which is generally used for binary class classification problem. To detect and localize manipulated region in given input image pixels of the image should be classified into white and black pixels i.e. forged and authenticated pixels. This is a binary class classification problem that's why the sigmoid activation function is better for such problems. Suppose y is input to the sigmoid function and $f(x)$ is the feature space of the output layer of the model and 'w' is weight then-

$$y = w \times f(x) + b \tag{1}$$

here 'b' is a bias value-added to the function. If we assume that predicted class of pixel $I = 0$ denotes that pixel is forged and $I = 1$ denotes that pixel is authenticated, then probabilities using function will be-

$$p(I = 0) = \frac{1}{1 + e^{-y}} \tag{2}$$

$$p(I = 1) = \frac{e^{-y}}{1 + e^{-y}} \tag{3}$$

Now, to evaluate the classifier's prediction capability there will be a need for loss function. The Loss function returns values for prediction. Higher values define bad prediction and lower values define good prediction by the classifier. Generally, for binary class classification binary cross-entropy loss function is used which is also known as a log loss function. Mathematically, loss function can be defined by:

$$L = -I \log(p(I)) + (1 - I)\log(1 - p(I)) \tag{4}$$

here I is the pixel level value of ground truth mask at location i.e. 1 for forged pixel and 0 for authenticated pixel and $p(I)$ is the probability of a pixel being forged for all pixels.

3.2 Training of the Model

For the training of the proposed model, two different datasets have been used in this paper. One is CoMoFoD and another is CMFD. The input size of the image is taken as 256×256. The kernel size for the convolution has been taken as 3×3. Padding during the convolution is taken as 'same' which means before and after the convolution height and width of the feature shape will be the same. Stride value is taken as one which means feature extraction is done on the sliding window of the image. Optimization function Adam is used for the training of the model with 200 maximum epochs. This optimizer updates the weight of the network after each epoch based on the given condition of validation accuracy i.e. weight will update with maximum patience of ten validation accuracy. After this condition weights will be saved in the middle of the iteration. This optimization is a stochastic gradient descent method that is based on the adaptive estimation of the first order and second-order moments whose default learning rate is 0.001 is taken in this training.

By modifying the layers of the architecture variants of this model can be designed. To analyze the performance, we have modified the layers of the architecture and made its variants. First of all, the kernel size of the convolution layers has been changed from 3×3 to 5×5 and 7×7. It is observed that if we extract features from large size kernels training as well as validation accuracy get reduces. The developed model is trained and validated on two different publicly available datasets i.e. CoMoFoD [33] and CMFD [34] datasets. For the first variant in which kernel size is taken as 3×3, the training and validation accuracy on the CMFD dataset were got 99.31% and 99.15% while on the CoMoFoD dataset were get 99.63% and 99.56% respectively. Training and validation loss on the CMFD dataset were 0.0201 and 0.0256 while on the CoMoFoD dataset were 0.0044 and 0.0072 respectively. The following table summarizes the training and validation accuracy with the size of trained weights (Table 2).

As mentioned above the maximum number of epochs has been taken as 200 for training and validation with a condition of validation accuracy should not degrade till 10 epochs. With this condition, the training of the CMFD dataset was run till 91 epochs after which validation accuracy was decreasing and after ten epochs of patience, training weight was saved. Similarly, on CoMoFoD dataset model run till 97 epochs and after 97th iteration validation accuracy was decreasing and hence training weight of 107th epochs was stored.

Table 2 Training result on the various kernel size of convolutional layers

	3×3		5×5		7×7	
	CoMoFoD	CMFD	CoMoFoD	CMFD	CoMoFoD	CMFD
Training accuracy	0.9963	0.9931	0.9958	0.9886	0.9949	0.9671
Training loss	0.0044	0.0201	0.0053	0.313	0.0080	0.1318
Validation accuracy	0.9956	0.9915	0.9955	0.9793	0.9948	0.9732
Validation loss	0.0072	0.0256	0.0065	0.0904	0.0083	0.4307
Size of weight	41 MB	42 MB	114 MB	114 MB	223 MB	222 MB
No. of params	10.71 M	10.71 M	29.73 M	29.73 M	58.26 M	58.26 M

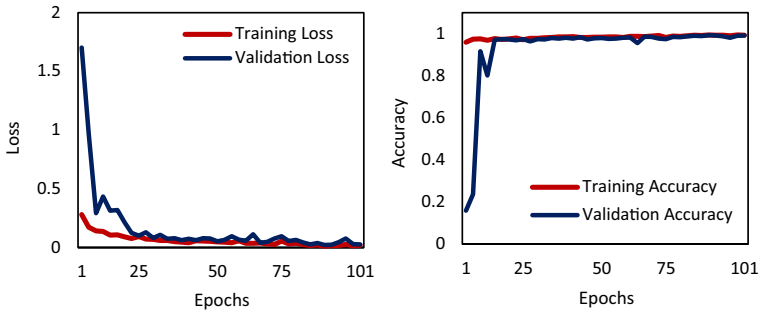


Fig. 8 Accuracy and loss of model (3×3) during training on CMFD dataset

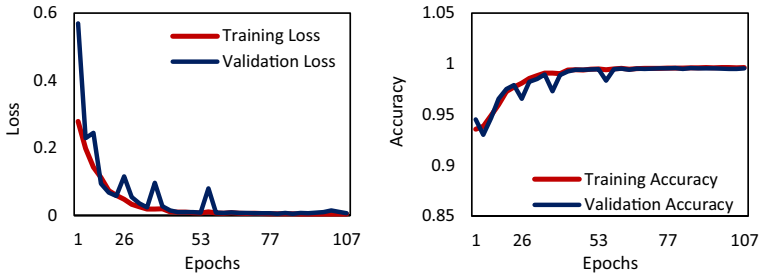


Fig. 9 Accuracy and loss of model (3×3) during training on CoMoFoD dataset

Training and validation accuracy with training and validation loss of each epoch on CMFD dataset is shown using a line graph in Fig. 8 and the same on CoMoFoD dataset is shown in Fig. 9.

The pseudo cod for the training of the model is given below:

Pseudo Code: Training of the proposed model

input: *D*: Input Training Dataset
output: *W*: Resultant Weight corresponding to the trained model
functions: resize: Resize original image into (h, w, d) form
split: Split Training Dataset into Training and Validation Data
scale: down sample input data
encoder_phase: consists of multiple convolution, down sample and concatenation blocks
decoder_phase: consists of multiple convolution, up sample and concatenation blocks
classification: classification of feature map
compile_train: compile and train the model using optimizer and loss function

```
initialize D;
images = resize (D, [256, 256, 3]);
masks = resize (D, [256, 256, 1]);
[T, V] = split(D,0.2)      //80% for training and 20% for validation
Function Model(T, V):
   S[n] = scale(T,V, [2,2]);   //half sampling of input data
   En = encoder_phase(S[n])          //Feature Map from encoder phase
   De = decoder_phase(En);           //Feature Map from decoder phase
   R = classification(De, sigmoid);  //Classification of feature map using sigmoid activation
Return model;
Model = model(T, V);
W = compile_train(Model, optimizer = adam, loss = binary_cross_entropy)
return W;
```

4 Performance Analysis and Discussion

This section of the paper explains the experiment performed for the validation of the proposed model on two different publicly available datasets. The following points are discussed in this section:

- System configuration and application details where code is written for the proposed model.
- Details of publicly available datasets on which training and validation have been performed.
- Evaluation metrics used for the performance analysis over the images of the dataset.
- Performance comparison of the proposed model with the existing state-of-the-art methods.

Simulation of the proposed model is performed on Ubuntu Server with the allocation of 16 GB memory and a single graphics processing unit. Simulation code of architecture is written in python language with the help of different python libraries. TensorFlow GPU is used for backend and Keras libraries for front end coding of the architecture. Except these additional helping libraries are used like os, matplotlib, NumPy and OpenCV. The written simulation code is submitted to the server using a job scheduler and training weights are stored on the hard disk. These weights are transferred to the local system where testing is done on test cases.

Two publicly available datasets have been used for training and validation purpose. One is CMFD [34] and another is CoMoFoD [33]. These datasets are diverse. They have a collection of geometrically transformed copy-move forgery attacks i.e. scaling and rotation. Except these CoMoFoD has a collection of mild processing copy-move forgery attacks i.e. brightness enhancement, noise addition, contrast adjustment, image blurring, JPEG compression and colour reduction. Details about these datasets are shown in the Table 3:

The rotation angles in both datasets were different from −25° to 360° the scaling factor is taken from 50 to 130. Mild processing operation on these images like JPEG compression quality factor is taken from 20 to 100 with a gap of 10, similarly, other operations are also taken for different factor to show the diverse nature of the dataset. To train and validate the model we have taken 70% and 20% of all image. To test the model rest of 10% are used.

To evaluate the performance of the proposed model and comparison of the proposed model with other state-of-the-art methods some evaluation metrics is required. In this paper, the model is validated on a pixel level as well as on an image level. Image level classification defines whether the model can classify forged and authenticated image or not and pixel-level classification defines that model can classify forged and authenticated pixels in a tampered image. The performance of the machine learning model can be evaluated using a confusion matrix. This confusion matrix has four different parameters—true positive, true negative, false positive and false negative. If the model predicts non-forged pixels as non-forged, then these pixels are counted as true positive. Similarly, if forged pixels are predicted as forged by the model then these pixels are counted as true negative. If non-forged pixels are predicted as forged, then these pixels are counted as these collected pixels are known as false positive. Similarly, if forged pixels are predicted as non-forged by the model, then these pixels are known as a false negative. Based on these four parameters, some performance metrics can be formulated such as precision, recall, accuracy, specificity, f1-score, miss rate and MCC.

Precision defines the ratio of correctly forged pixels and total predicted forged pixels where recall defines the ratio of correctly forged pixel and total actual forged pixels. Accuracy defines the ratio of correctly classified pixels and the total population of pixels and accuracy gives biased result in the case of imbalanced pixels class. Even F1-score is also unreliable in such case which is the harmonic mean of precision and recall. In the case of forged pixels localization, an imbalanced class of pixels problem will always arise. That's why the Matthews correlation coefficient (MCC) is more important. This performance measure is the most suitable in the case of a binary class classification problem. Mathematically, MCC can be defined as:

$$\mathrm{MCC} = \frac{(\mathrm{tp} \times \mathrm{tn} - \mathrm{fp} \times \mathrm{fn})}{\sqrt{((\mathrm{tp} + \mathrm{fp}) \times (\mathrm{tp} + \mathrm{fn}) \times (\mathrm{tn} + \mathrm{fp}) \times (\mathrm{tn} + \mathrm{fn}))}} \tag{5}$$

Using the above-mentioned performance measures result analysis has been done on two different datasets. From datasets, 10% of the images have been taken out to test the proposed model and the rest of the images are used to train and validate the model. Except for these images from both datasets 100 non-forged images (50 from each) have been also taken to test the model. The average precision, recall, accuracy, specificity (aka TNR), FNR (aka miss-rate), F1-score and MCC values are calculated and shown in Table 4. This result is the combined and average result of the geometrical transformation as well as post-processing operations. Individual results on different post-processing and different geometrical transformation are shown in different tables. The visual result is also shown for the individual dataset. Performance analysis of the proposed

Table 3 Details of the publicly available dataset used

S. no.	Transformation	CoMoFoD			CMFD		
		Resolution	Forged images	Respected mask	Resolution	Forged images	Respected mask
1	Natural (Without Transformation)	512×512	40	40	700×1000	100	100
2	Scaling	512×512	40	40	700×1000	320	320
3	Rotation	512×512	40	40	700×1000	600	600
4	Distortion (Skew)	512×512	40	40	–	–	–
5	Combination	512×512	40	40	–	–	–
6	JPEG Compression	512×512	1800	1800	–	–	–
7	Image Blurring	512×512	600	600	–	–	–
8	Noise Addition	512×512	600	600	–	–	–
9	Brightness Change	512×512	600	600	–	–	–
10	Colour Reduction	512×512	600	600	–	–	–
11	Contrast Adjustment	512×512	600	600	–	–	–
		Total	5000	5000		1020	1020

Table 4 Average test result using performance measures precision, recall, accuracy, TNR, FNR, F1-score and MCC value on different datasets

Dataset	P	R	A	TNR	FNR	F1	MCC
CMFD	0.9892	0.9982	0.9878	0.7764	0.0018	0.9936	0.8329
CoMOoFoD	0.9863	0.9962	0.9839	0.8247	0.0038	0.9909	0.8578

model is also shown using a line graph on different post-processing operation for both datasets.

Table 5 shows the average result on the CoMoFoD dataset for different post-processing operations using performance measures precision, recall, accuracy, true negative rate, false-negative rate, f1-score and MCC values. In the table post-processing operations are shown using abbreviations. These abbreviations stand for—F: Only translation without post-processing, BC1-BC3: Brightness change, CA1-CA3: Contrast

Table 5 Average test result using performance measures precision, recall, accuracy, TNR, FNR, F1-score and MCC value on CoMoFoD dataset on different post-processing operations

Post-processing	P	R	A	TNR	FNR	F1	MCC
F	0.9863	0.9961	0.9838	0.8215	0.0039	0.9909	0.8558
BC1	0.9862	0.9961	0.9837	0.8186	0.0039	0.9908	0.8537
BC2	0.9860	0.9961	0.9835	0.8152	0.0039	0.9907	0.8510
BC3	0.9852	0.9961	0.9827	0.8029	0.0039	0.9903	0.8419
CA1	0.9864	0.9961	0.9840	0.8261	0.0039	0.9910	0.8589
CA2	0.9865	0.9962	0.9841	0.8293	0.0038	0.9910	0.8613
CA3	0.9858	0.9963	0.9835	0.8279	0.0037	0.9907	0.8607
CR1	0.9863	0.9961	0.9838	0.8213	0.0039	0.9909	0.8556
CR2	0.9863	0.9961	0.9838	0.8212	0.0039	0.9909	0.8556
CR3	0.9863	0.9961	0.9838	0.8214	0.0039	0.9909	0.8556
IB1	0.9861	0.9964	0.9838	0.8221	0.0036	0.9909	0.8575
IB2	0.9862	0.9964	0.9840	0.8217	0.0036	0.9910	0.8579
IB3	0.9862	0.9964	0.9841	0.8195	0.0036	0.9910	0.8560
JC1	0.9862	0.9962	0.9838	0.8277	0.0038	0.9909	0.8595
JC2	0.9860	0.9962	0.9835	0.8268	0.0038	0.9907	0.8573
JC3	0.9864	0.9963	0.9841	0.8287	0.0037	0.9910	0.8610
JC4	0.9867	0.9962	0.9843	0.8300	0.0038	0.9911	0.8611
JC5	0.9865	0.9961	0.9840	0.8298	0.0039	0.9910	0.8604
JC6	0.9865	0.9962	0.9841	0.8281	0.0038	0.9910	0.8602
JC7	0.9863	0.9961	0.9837	0.8261	0.0039	0.9909	0.8573
JC8	0.9863	0.9961	0.9837	0.8245	0.0039	0.9909	0.8570
JC9	0.9871	0.9961	0.9846	0.8456	0.0039	0.9913	0.8710
NA1	0.9869	0.9962	0.9846	0.8269	0.0038	0.9913	0.8600
NA2	0.9868	0.9961	0.9843	0.8278	0.0039	0.9911	0.8598
NA3	0.9868	0.9961	0.9843	0.8273	0.0039	0.9912	0.8593

Adjustment, CR1-CR3: Color Reduction, IB1-IB3: Image blurring, JC1-JC9: JPEG Compression and NA1-NA3: Noise addition.

Visual results obtained by the proposed model on the CoMoFoD dataset can be seen in Fig. 10. From all post-processing, a random image is taken and validated through the

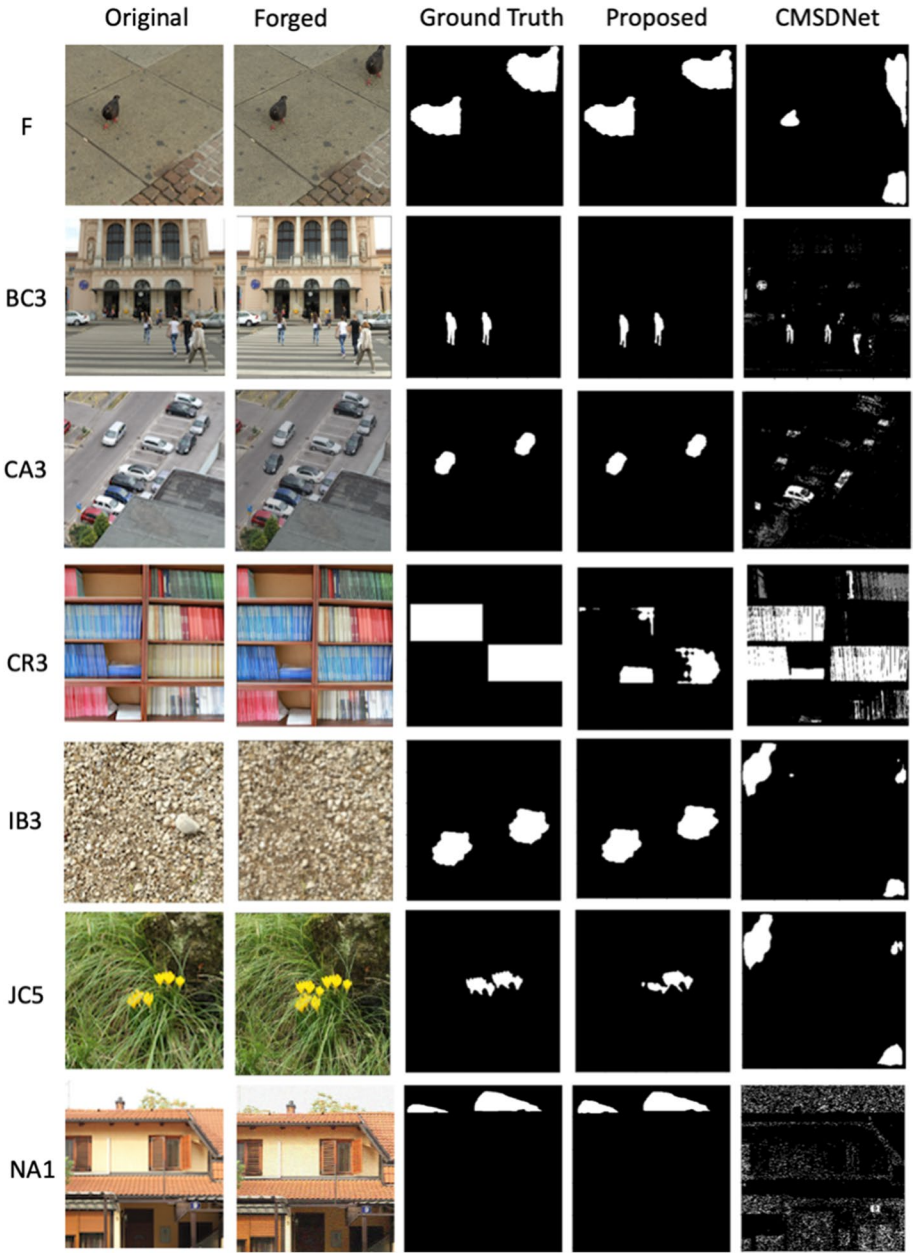


Fig. 10 The visual result of the proposed model on test images of CoMoFoD dataset- the result of the proposed model is shown as predicted and compared state-of-the-art is shown as CMSDNet

proposed model. Its predicted visual result can be seen in the fourth column of the shown figure. The first column shows the original image, the second column represents copied and pasted region from the original image and made it tampered whose ground truth mask is shown in the third column. The visual results shown in the fifth column is the predicted output of the state-of-the-art CMSDNet [32]. Seven rows of the visual result represent different post-processing operations performed on the tampered region. These operations are brightness enhancement, contrast adjustment, colour reduction, image blurring, JPEG compression and noise addition respectively. From these visual results, it can be concluded that the proposed model gives better result in all the cases of post-processing operations of the forged region. However, the only case of colour reduction shown in the visual result tells that the proposed method doesn't work well. Though the visual result of this case doesn't look very impressive as the method doesn't localize the whole forged region in the output result. The reason behind the false positives is that the image has multiple similar texture region, not because of the colour reduction of the forged region. Hence, the proposed model gets confused between regions and performance becomes lower. Except this in all cases result by the proposed model gives better performance.

The result analysis of the proposed model on different post-processing operations of the CoMoFoD dataset using a line graph is shown in Fig. 11. In the first line graph precision, recall, accuracy and f1-score are shown which is not suitable in case of an imbalanced number of pixels in an image. So, another line graph is shown using MCC and true negative rate values. From the first line graph, it can be seen that in almost all post-processing operations accuracy is more than 98% and the F1-score is more than 99%. MCC values in all cases are lesser than 86%, however, MCC is a better performance measure and 86% performance is much acceptable.

The comparisons of the proposed result with stat-of-the-art techniques using precision, recall and f1-score are given in tables from Tables 6, 7 and 8. In these compared methods two methods are based on deep learning network. Table 6 shows the comparison of the performance of the proposed model with state-of-the-art techniques on the various transformation of the CoMoFoD dataset without any post-processing operations and with the post-processing operation JPEG compression of quality factor 90. However, images are also compressed from a quality factor of 20 to a quality factor of 90 with a gap of 10. In this table quality factor, 90 is compared because these images have maximum quality factor compression.

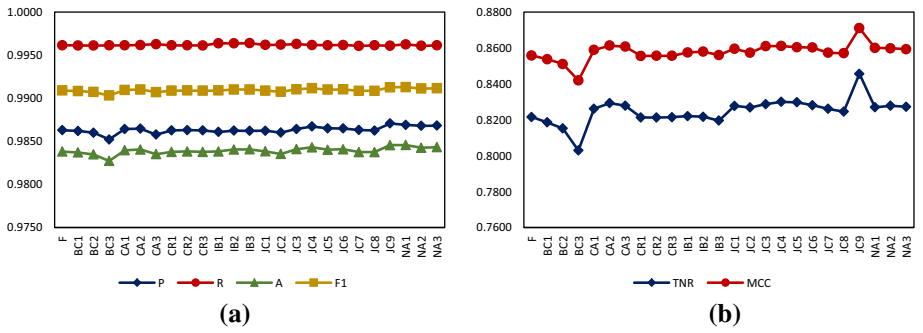


Fig. 11 Performance analysis of the proposed model using line graph on CoMoFoD dataset **a** Performance analysis using **p**recision, **r**ecall, **a**ccuracy and **F1**-score, **b** Performance analysis using TNR and MCC values

Table 6 The compared result of the proposed model with state-of-the-art methods on images of CoMoFoD dataset without any post-processing and with JPEG compression qf = 90

Algorithm	Transformation	Without any post-processing			JPEG compression qf = 90		
		P	R	F1	P	R	F1
[35]	Rotation	0.5594	0.8281	0.5979	0.5809	0.8817	0.6285
	Scaling	0.5542	0.8492	0.6059	0.5630	0.8448	0.6037
	Distortion	0.6425	0.9045	0.6961	0.6490	0.8860	0.6902
[36]	Rotation	0.5532	0.7451	0.5573	0.3990	0.3779	0.2708
	Scaling	0.4966	0.6971	0.5008	0.4858	0.3567	0.3000
	Distortion	0.5814	0.7878	0.6550	0.5625	0.3825	0.3571
[19]	Rotation	0.6833	0.9006	0.7174	0.5977	0.8014	0.6369
	Scaling	0.5696	0.7516	0.5864	0.5169	0.7248	0.5449
	Distortion	0.6631	0.8516	0.6986	0.5963	0.7787	0.7175
[32]	Rotation	0.9845	0.9834	0.9839	0.9910	0.9965	0.9937
	Scaling	0.9907	**0.9973**	0.9940	0.9956	0.9367	0.9652
	Distortion	0.9045	0.9559	0.9295	0.9742	0.9409	0.9573
Proposed	Rotation	**0.9943**	**0.9973**	**0.9958**	**0.9942**	**0.9974**	**0.9958**
	Scaling	**0.9964**	0.9970	**0.9967**	**0.9964**	**0.9971**	**0.9967**
	Distortion	**0.9808**	**0.9956**	**0.9877**	**0.9807**	**0.9956**	**0.9877**

Values in bold texts show the better performance values

Table 7 The compared result of the proposed model with state-of-the-art methods on images of CoMoFoD dataset with Noise addition (variance = 0.0005) and Image Blurring

Algorithm	Transformation	Noise addition (variance = 0.0005)			Image blurring		
		P	R	F1	P	R	F1
[35]	Rotation	0.6202	0.8399	0.6528	0.4481	0.8753	0.5280
	Scaling	0.5673	0.7438	0.5849	0.4514	0.9096	0.5304
	Distortion	0.6806	0.7821	0.7013	0.5022	0.9449	0.5953
[36]	Rotation	0.5924	0.6481	0.5265	0.5183	0.7043	0.5335
	Scaling	0.6159	0.5115	0.4987	0.5281	0.6994	0.5212
	Distortion	0.6828	0.5627	0.5270	0.6243	0.8292	0.6048
[19]	Rotation	0.6385	0.8076	0.6578	0.5114	0.8591	0.5945
	Scaling	0.5838	0.6840	0.5677	0.4890	0.7836	0.5540
	Distortion	0.7380	0.8411	0.7627	0.5715	0.8949	0.6611
[32]	Rotation	0.9907	0.9933	0.9920	0.9908	0.9968	0.9938
	Scaling	0.9948	0.9005	0.9453	0.9907	0.9966	0.9937
	Distortion	0.9661	0.9084	0.9363	0.9177	0.9426	0.9300
Proposed	Rotation	**0.9944**	**0.9972**	**0.9958**	**0.9940**	**0.9977**	**0.9958**
	Scaling	**0.9965**	**0.9969**	**0.9967**	**0.9961**	**0.9974**	**0.9967**
	Distortion	**0.9815**	**0.9956**	**0.9881**	**0.9806**	**0.9957**	**0.9877**

Values in bold texts show the better performance values

The CoMoFoD dataset constitutes images with three types of additive noise in which noises are added in the image with the variance of 0.005, 0.009 and 0.0005. From which

Table 8 The compared result of the proposed model with state-of-the-art methods on images of the CoMoFoD dataset with brightness change, colour reduction and contrast adjustment

Algorithm	Transformation	Brightness change			Colour reduction			Contrast adjustment		
		P	R	F1	P	R	F1	P	R	F1
[35]	Rotation	0.5601	0.8445	0.5933	0.5692	0.8340	0.6174	0.5447	0.8428	0.5972
	Scaling	0.5537	0.7860	0.5926	0.5628	0.8969	0.6251	0.5714	0.8400	0.5973
	Distortion	0.6464	0.8964	0.6892	0.6352	0.9226	0.6955	0.6445	0.8994	0.6941
[36]	Rotation	0.5272	0.7314	0.5333	0.5432	0.6567	0.5066	0.5722	0.6987	0.5321
	Scaling	0.3977	0.6305	0.4378	0.5004	0.7092	0.5129	0.5334	0.7227	0.5059
	Distortion	0.4834	0.7168	0.5137	0.6147	0.7584	0.5813	0.6450	0.7854	0.6080
[19]	Rotation	0.6075	0.9063	0.6531	0.6583	0.8415	0.6891	0.6157	0.8675	0.6590
	Scaling	0.5350	0.7290	0.5526	0.5984	0.7832	0.6267	0.5517	0.7987	0.5948
	Distortion	0.6342	0.7814	0.6609	0.6652	0.8193	0.6827	0.6610	0.8476	0.6924
[32]	Rotation	0.9908	**0.9972**	0.9940	0.9907	0.9972	0.9939	0.9906	**0.9974**	0.9940
	Scaling	0.9909	**0.9971**	0.9940	0.9908	**0.9972**	0.9940	0.9905	**0.9975**	0.9940
	Distortion	**0.9895**	0.9877	**0.9886**	0.9562	0.9303	0.9431	0.9781	0.9780	0.9781
Proposed	Rotation	**0.9938**	**0.9972**	**0.9955**	**0.9943**	**0.9973**	**0.9957**	**0.9943**	0.9973	**0.9957**
	Scaling	**0.9957**	0.9968	**0.9963**	**0.9964**	0.9969	**0.9966**	**0.9936**	0.9974	**0.9955**
	Distortion	0.9795	**0.9957**	0.9870	**0.9807**	**0.9956**	**0.9877**	**0.9815**	**0.9956**	**0.9880**

Values in bold texts show the better performance values

result on images with additive noise of variance of 0.0005 is compared. Image blurring is a post-processing operation which is performed on images using different kernel size. In this dataset, image blurring is performed on three different kernel size i.e. 3×3, 5×5 and 7×7. Table 7 shows the compared result of the proposed model with state-of-the-art methods on different test images of the CoMoFoD dataset. These images are either having post-processing of noise addition or having image blurring. After these post-processing operations, some transformation like rotation, scaling and distortion are also performed. Distortion comprises of different skews.

Another post-processing operation performed on images of the CoMoFoD dataset is brightness change. The result produced by the proposed architecture on these images and comparison of the result with other state-of-the-art methods is shown in Table 8. In this table, the proposed model is compared with other state-of-the-art methods on the images of the CoMoFoD dataset with colour reduction and contrast stretching. A post-processing operation 'colour reduction' is performed on images of the dataset three times with different parameter. The average result of the proposed method and stat-of-the-art methods are shown in the table. Contrast stretching is also performed on the images of the dataset. The average result on these images with different geometric transformation is shown in this table. A comparison of the proposed method with other state-of-the-art methods is also done in this table. From these comparisons, it can be deduced that the proposed deep learning approach is far better than the conventional approaches. Although in the case of distortion (i.e. horizontal skewness and vertical skewness) result is lesser than rotation and scaling, the overall result of the proposed model is greater than other state-of-the-art methods.

Since, in almost every test case, the proposed method performs better than the compared state-of-the-art techniques (see above tables), it can be concluded that the proposed method is invariant to geometrical transformation like rotation, scaling and distortion. Also, the method is invariant to post-processing operations like brightness enhancement, colour reduction, contrast adjustment, noise addition, JPEG compression and Image blurring.

Another dataset is CMFD on which the proposed model is evaluated using different performance measure. Here also, pixels are imbalanced concerning forged and non-forged classes. Here also MCC value is much important, so except precision, recall, accuracy and F1-score other performance measures are calculated like MCC, miss rate and specificity. Visual results given by the proposed model on this dataset is also shown in Fig. 12. The geometric transformation-wise average quantitative result on the given dataset is shown in Table 9.

The visual result of the proposed model on the given dataset is shown in the following Fig. 12. In this visual result, the first row contains a direct translation of the copied region and made it tampered with using a copy-move forgery attack. The second and the third row represent rotation geometric transformation of the copied region and the fourth, fifth rows show scaling geometric transformation. The first column of visual result shows the original image, the second column shows tampered image, the third column represents the ground truth mask of a tampered image and the last column represents the visual outcome from the proposed model.

The visual result in Fig. 12 shows that the proposed model is scaling invariant. The fourth and fifth rows have tampered images in which seashell and bird is scaled up and the proposed model can identify the forged region. To analyze the performance of the proposed model on different transformed images of the given dataset line graphs of all performance measure are shown in Fig. 13. Precision, recall, accuracy and f1-score are shown in Fig. 13a and true negative rate and MCC values are shown in Fig. 13b.

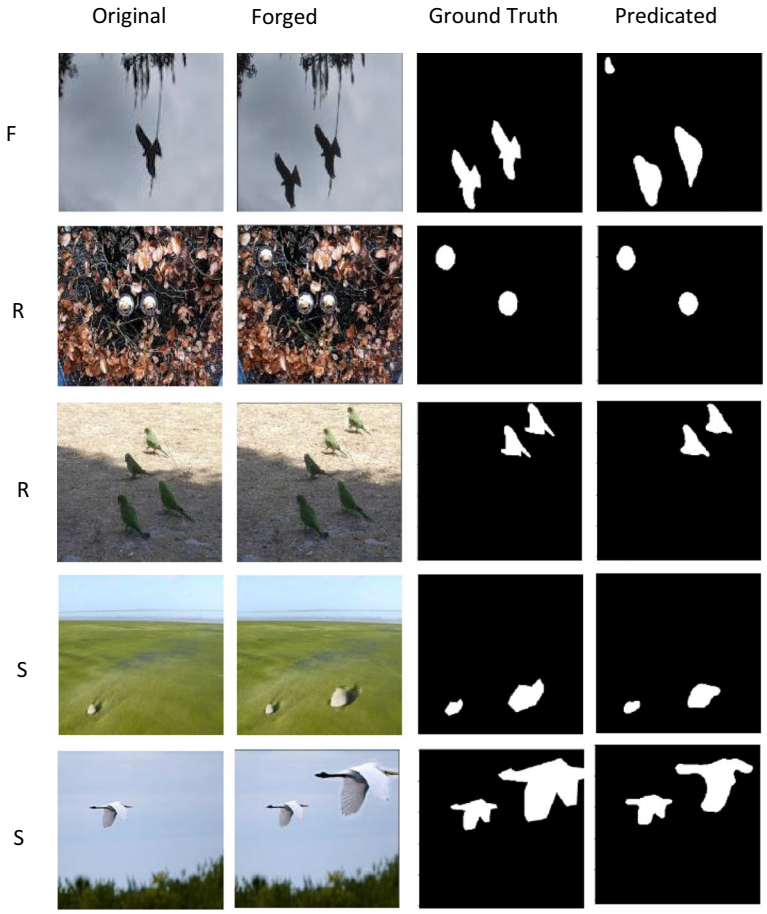


Fig. 12 The visual result of the proposed model on images of the CMFD Dataset

Table 9 The proposed model result on CMFD dataset on different transformation

Transformation	P	R	A	TNR	FNR	F1	MCC
Translation	0.9750	0.9862	0.9633	0.4475	0.0138	0.9805	0.4325
Rotation	0.9972	0.9988	0.9961	0.8825	0.0012	0.9980	0.9136
Scaling	0.9765	0.9989	0.9762	0.6290	0.0011	0.9874	0.7443

The above-mentioned results and comparisons are pixel-level analysis. In pixel-level analysis pixels of an image can be classified into forged and non-forged pixels. Hence forged region in an image can be localized in the pixel-level analysis. Another analysis is image level, in this analysis, the only image is to be checked whether it has tampered with

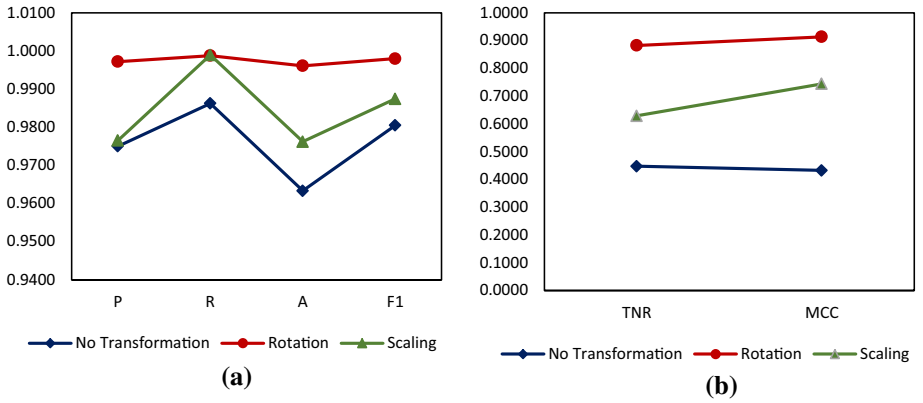


Fig. 13 Performance analysis of the proposed model using line graph on CMFD dataset **a** Performance analysis using **p**recision, **r**ecall, **a**ccuracy and **F1**-score, **b** Performance analysis using TNR and MCC values

or not. The forged region is not to be localized in the image level analysis. In this paper image-level analysis is also done based on the number of true-negative, false-positive and false-negative pixels. From both dataset 50–50 original images have been taken and given to the trained model. In the case of the model predicted zero true negative pixels and very lesser false positive and false negative pixels. Then the image is counted non-tampered image else it will be counted as tampered image. Based on these tampered and non-tampered images, confusion matrix for both the dataset are shown in Fig. 14.

Based on these confusion matrix performance measures of the proposed model is also calculated. Evaluated results obtained on both datasets are given in the following Table 10.

Block-based and key-point approaches take more time to produce localized result than data-driven approaches. However, data-driven approaches take much time during the training of the model, but a trained model takes lesser time to predict the localized result. The proposed approach takes 3.8 s to predict forged regions in ten images while CMSDNet [32] takes 2.7 s to predict forged regions in ten images. The latter one takes lesser time

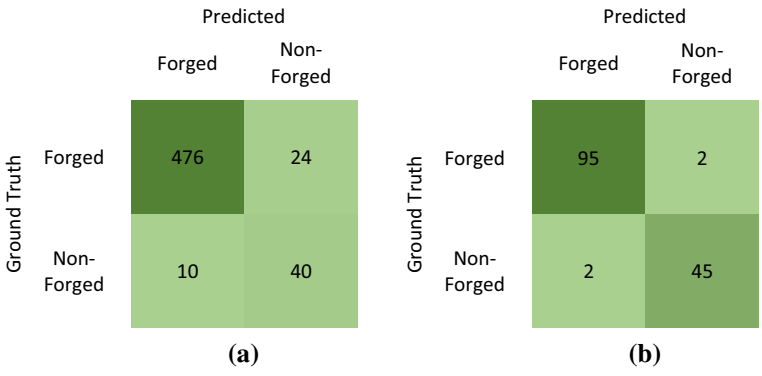


Fig. 14 Image level analysis of the proposed model on both datasets **a** Image level analysis on CoMoFoD dataset, **b** Image level analysis on CMFD dataset

Table 10 Image level analysis of the proposed model on different datasets

	P	R	A	TNR	FNR	F1	MCC
CoMoFoD	0.9794	0.9520	0.9382	0.8000	0.0480	0.9655	0.6742
CMFD	0.9794	0.9794	0.9722	0.9574	0.0206	0.9794	0.9368

but doesn't give precise information about the forged location. The proposed one takes a slightly longer time but gives precise information about the forged location. And in this application predicting the accurate location of the forged region in an image is very important.

5 Conclusion

In this paper, a data-driven approach is developed for the detection of copy-move forgery attacks in digital images. Copy-move forgery is a commonly used forgery attack in the digital image to conceal any information in the image. A lot of literature already reported for the detection and localization of such type of forgery attacks. These are block-based approaches, keypoint based approaches and data-driven approaches. Block-based approaches are suffering from the problems of transformation of forged region and computation cost. The keypoints based approach is suffering from the problem of clustering in the case of small region duplication. Data-driven approaches are either based on image-level analysis, to the best of our knowledge none of the data-driven approaches is there for localization of the forged region. In this paper, a multi-scale input based deep learning convolution neural network is developed to localize the forged region in a copy-move forged image. The multi-scale input image and its convolution features are scale-invariant this is the reason why the scaled forged region is also identified by the proposed model. The proposed model is trained and validated on two different publicly available datasets. In this paper, image-level analysis, as well as pixel-level analysis, is done. The second dataset has a small number of images to train the model. This is the reason why performance is lower in the case of the second dataset. In future, a combined generalized dataset can be made for both types of attacks i.e., spliced and copy-move forgery. This generalized dataset can be used to train a generalized model which can predict the spliced as well as copy-move forged location in the image. The segmentation task can be also performed by dictionary learning [37]. In future, a segmentation using dictionary learning can be used with the proposed deep learning technique for a better result.

Declarations

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

References

1. Friedman GL (1993) The trustworthy digital camera: restoring credibility to the photographic image. IEEE Trans Consum Electron 39:905–910. https://doi.org/10.1109/30.267415
2. Fridrich J (1999) Methods for tamper detection in digital images. Proc Work Multimed Secur, pp 19–23
3. Lin C, Chang S (2001) A robust image authentication method distinguishing JPEG compression from malicious manipulation. IEEE Trans Circuits Syst Video Technol 11:153–168
4. Fridrich J, Soukal D, Lukáš J (2003) Detection of copy-move forgery in digital images. Proc Digit Forensic Res Work. https://doi.org/10.1109/PACIIA.2008.240
5. Popescu AC, Farid H (2004) Exposing digital forgeries by detecting duplicated image regions. Dept Comput Sci, Dartmouth Coll Tech Rep TR2004-515 1–11. https://doi.org/10.1109/TSP.2004.839932
6. Weiqi L, Jiwu H, Guoping Q (2006) Robust detection of region-duplication forgery in digital image. In: International conference on pattern recognition, pp 746–749
7. Langille A, Gong M (2006) An efficient match-based duplication detection algorithm. In: Third Canadian conference on computer and robot vision, CRV 2006, pp 1–8
8. Mahdian B, Saic S (2007) Detection of near-duplicated image regions. Adv Soft Comput 45:187–195. https://doi.org/10.1007/978-3-540-75175-5_24
9. Lin H, Wang C, Kao Y (2009) Fast copy-move forgery detection. WSEAS Trans Signal Process 5:188–197
10. Zimba M, Xingming S (2011) DWT-PCA (EVD) based copy-move image forgery detection. Int J Digit Content Technol Appl 5:251–258. https://doi.org/10.4156/jdcta.vol5.issue1.27
11. Amerini I, Ballan L, Caldelli R et al (2011) A SIFT-based forensic method for copy-move attack detection and transformation recovery. IEEE Trans Inf Forensics Secur 6:1099–1110. https://doi.org/10.1109/TIFS.2011.2129512
12. Yang B, Sun X, Guo H et al (2017) A copy-move forgery detection method based on CMFD-SIFT. Multimed Tools Appl. https://doi.org/10.1007/s11042-016-4289-y
13. Yang F, Li J, Lu W, Weng J (2017) Copy-move forgery detection based on hybrid features. Eng Appl Artif Intell 59:73–83. https://doi.org/10.1016/j.engappai.2016.12.022
14. Ngo L, Cha J, Han JH (2020) Deep neural network regression for automated retinal layer segmentation in optical coherence tomography images. IEEE Trans Image Process 29:303–312. https://doi.org/10.1109/TIP.2019.2931461
15. Xiao Y, Zijie Z (2020) Infrared image extraction algorithm based on adaptive growth immune field. Neural Process Lett 51:2575–2587. https://doi.org/10.1007/s11063-020-10218-7
16. Yuan Y, Xiong Z, Wang Q (2019) VSSA-NET: vertical spatial sequence attention network for traffic sign detection. IEEE Trans Image Process 28:3423–3434. https://doi.org/10.1109/TIP.2019.2896952
17. Wang Q, Han T, Qin Z et al (2020) Multitask attention network for lane detection and fitting. IEEE Trans Neural Networks Learn Syst. https://doi.org/10.1109/TNNLS.2020.3039675
18. Elaskily MA, Elnemr HA, Sedik A et al (2020) A novel deep learning framework for copy-move forgery detection in images. Multimed Tools Appl. https://doi.org/10.1007/s11042-020-08751-7
19. Liu Y, Guan Q, Zhao X (2018) Copy-move forgery detection based on convolutional kernel network. Multimed Tools Appl 77:18269–18293. https://doi.org/10.1007/s11042-017-5374-6
20. Novozámský A, Šorel M (2018) Detection of copy-move image modification using JPEG compression model. Forensic Sci Int 283:47–57. https://doi.org/10.1016/j.forsciint.2017.11.031
21. Takahashi R, Matsubara T, Uehara K (2019) A novel weight-shared multi-stage CNN for scale robustness. IEEE Trans Circuits Syst Video Technol 29:1090–1101. https://doi.org/10.1109/TCSVT.2018.2822773
22. Mahmood T, Irtaza A, Mehmood Z, Tariq Mahmood M (2017) Copy–move forgery detection through stationary wavelets and local binary pattern variance for forensic analysis in digital images. Forensic Sci Int 279:8–21. https://doi.org/10.1016/j.forsciint.2017.07.037
23. Jaiswal AK, Srivastava R (2019) Copy-move forgery detection using shift-invariant SWT and block division mean features. Springer
24. Mahdian B, Saic S (2007) Detection of copy-move forgery using a method based on blur moment invariants. Forensic Sci Int 171:180–189. https://doi.org/10.1016/j.forsciint.2006.11.002
25. Bayram S, Sencar HT, Memon N (2009) An efficient and robust method for detecting copy-move forgery. In: IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 1053–1056
26. Pan XPX, Lyu SLS (2010) Region duplication detection using image feature matching. IEEE Trans Inf Forensics Secur 5:857–867. https://doi.org/10.1109/TIFS.2010.2078506

27. Mahmood T, Mehmood Z, Shah M, Khan Z (2017) An efficient forensic technique for exposing region duplication forgery in digital images. Appl Intell. https://doi.org/10.1007/s10489-017-1038-5
28. Agarwal S, Chand S (2018) Image forgery detection using co-occurrence-based texture operator in frequency domain. Adv Intell Syst Comput 519:117–122. https://doi.org/10.1007/978-981-10-3376-6
29. Mehta V, Jaiswal AK, Srivastava R (2020) Copy-move image forgery detection using DCT and ORB feature set. Springer
30. Aloraini M, Sha L, Sharifzadeh M, Schonfeld D (2019) Dictionary learning and sparse coding for digital image forgery detection. IS T Int Symp Electron Imaging Sci Technol 2019:1–7. https://doi.org/10.2352/ISSN.2470-1173.2019.5.MWSF-531
31. Wu Y, Abd-Almageed W, Natarajan P (2018) Image copy-move forgery detection via an end-to-end deep neural network. In: Proceedings of the 2018 IEEE winter conference on applications of computer vision, WACV 2018 2018-Janua:1907–1915. https://doi.org/10.1109/WACV.2018.00211
32. Chen B, Tan W, Coatrieux G et al (2020) A serial image copy-move forgery localization scheme with source/target distinguishment. IEEE Trans Multimed 9210:1–1. https://doi.org/10.1109/tmm.2020.3026868
33. Tralic D, Zupancic I, Grgic S (2013) GM CoMoFoD—new database for copy-move forgery detection. In: Proceedings of the 55th international symposium ELMAR-2013. http://www.vcl.fer.hr/comofod/download.html. Accessed 11 Oct 2017
34. Ardizzone E, Bruno A, Mazzola G (2015) Copy-move forgery detection by matching triangles of keypoints copy-move forgery detection by matching triangles of keypoints. IEEE Trans Inf Forensics Secur 10:2084–2094. https://doi.org/10.1109/TIFS.2015.2445742
35. Li J, Li X, Yang B, Sun X (2015) Segmentation-based image copy-move forgery detection scheme. IEEE Trans Inf Forensics Secur 10:507–518. https://doi.org/10.1109/TIFS.2014.2381872
36. Silva E, Carvalho T, Ferreira A, Rocha A (2015) Going deeper into copy-move forgery detection: exploring image telltales via multi-scale analysis and voting processes. J Vis Commun Image Represent 29:16–32. https://doi.org/10.1016/j.jvcir.2015.01.016
37. Li X, Shen H, Member S et al (2014) Contaminated by thick clouds and shadows using multitemporal dictionary learning. IEEE Trans Geosci Remote Sens 52:7086–7098. https://doi.org/10.1109/TGRS.2014.2307354