Journal of Engineering Science and Technology
Vol. 17, No. 3 (2022) 2000 - 2019

COPY-MOVE FORGERY DETECTION - A HYBRID APPROACH

JIGNA J. PATEL[1,*], NINAD S. BHATT[2]

[1]Gujarat Technological University, Nr.Vishwakarma Government Engineering College
Nr.Visat Three Roads, Visat - Gandhinagar Highway,
Chandkheda, Ahmedabad - 382424 – Gujarat, India
[2]C.K Pithawalla College of Engineering and Technology Near Malvan Mandir Via
Magdalla Port, Dumas Rd, Surat, Gujarat 395007, India
*Corresponding Author: jigna2012me@gmail.com

Abstract

Copy-paste (move) forgery is image manipulation by copying one region and pasting it within the same image but to another location either by transforming the copied region or simply by pasting it to another location. The proposed hybrid network in this paper is a combination of LSTM (Long short-term memory) as well as CNN (Convolution Neural Network) to detect copied portion that is pasted in the same digital image. Localize copied and pasted parts in the digital image were detected using a proposed model, that is a combination of SRM (steganalysis rich model) filter, LSTM, and CNN, and finally SVM (Support Vector Machine) classification. For resampling features, artifacts such as upsampling or downsampling of replicated parts, as well as rotation and shearing, patch extraction and rotation of extracted patches was employed. Firstly, patches were extracted from pristine as well as tampered images from the dataset. Secondly, features are extracted on these overlapping patches of the image, as they pass through an LSTM-CNN based network. Finally, use SVM Classifier was done for analysing the distinct attributes that discriminate manipulated regions from non-manipulated regions. The detection/localization were performed on two datasets CASIA and CoMoFoD and the experimental results depict that the technique utilized is effective and an accuracy of 94.7% was achieved with patch rotation as well as 82.8% was achieved without rotating the patches for CASIA dataset. Further, it yielded 84.8% accuracy for CoMoFoD dataset with patch rotation and 73.5% accuracy without patch rotation.

Keywords: Convolution neural network, Copy-move forgery, Hybrid architecture, Long short-term memory, Patch extraction, Support vector machine.

1. Introduction

Because of the extensive usage of digital cameras, smartphones and tablets, there has been an exponential growth in the number of digital images that are posted on social media like Instagram, WhatsApp, Twitter, Facebook, etc, and can be forged/tampered with by anyone using social media. Also, easily available tools and software like PaintShop Pro, Photoshop, and smartphone applications like Instasize, Snap seed, etc. made it too easy for these users to manipulate or enhance the images. In today's scenario, as there is a close resemblance between the tampered and pristine images, the authenticity of the image is too difficult. However, in almost every image that is manipulated, the tampered regions have a typical characteristic that exhibits distinguishing features between the boundaries shared among tampered and non-tampered regions. There are different ways to alter an image like the addition, alteration, or deletion of some remarkable features from an image in such a way that no traces of forgery will be visible in the image. These forgeries include Copy-Move/Copy and Paste forgery, Image splicing and resampling [1].

- Copy-Move/Copy and Paste Forgery: In copy-move cloning, portion of any digital image of various shapes and sizes is copied and pasted to other region of the same picture. As the copied and pasted part belongs to the same image, it's essential attributes like color, noise as well as and texture wouldn't change resulting in making the process of recognition of tampered regions troublesome.
- Splicing: Image splicing involves two different image regions pasted into a single one, to create a tampered image. The borders among the spliced regions can be invisible if splicing is performed precisely making it much more difficult to detect.
- Image Resampling: Resampling is a method that transforms the digital image by selecting portions of the image, and transforming it by geometric transformations such as skewing, rotation, flipping, stretching, scaling, etc.

There are different techniques with which the forged image can be detected. To date, numerous algorithms for detecting digital image forgeries based on artifacts from copy and paste, resampling, lighting, camera forensic photos, JPEG compressed images, and other factors have been proposed. They can be grouped as block-based and keypoint based techniques. Block-based methods include dividing an image by overlapping blocks. Mainly the block-based techniques differ in the feature extraction used for matching the blocks. On the other hand, the key point-based methods include extracting and matching features that are extracted from the overall image to identify the forged regions. The block-based methods and key-point-based technique that work with overlapping blocks or extraction of features from entire images are computationally expensive but the recall rate of both these methods is low. All of these techniques highly depend on a threshold value to be computed experimentally for feature matching. Hence, the robustness of these algorithms is reduced, and thus overall accuracy also reduces [1].

Moreover, it needs much trial and error to implement this classic approach for detection of copy-move forgery as such algorithms need to tune up various parameters that depend on data. For example, deciding the number of key points to be extracted for each image, or size of a block to be used, and the ratio of overlapping blocks during the unit stage of feature representation. Also, it may require tuning of parameters in post-processing. For example, if one uses the comprehensive approach for matching, the minimum distance between matching

candidates, the minimal number of correspondences, area threshold, rules for splitting one unit group into two different groups or merging two different groups into one group, and threshold to fill a hole, etc. [2] would be necessary to be specified. The set of parameters that works well for one type of image may fail for the other if these kinds of approaches are used. Also, the selection of parameters that working well for each testing image on a trial-and-error basis is possible only for a small number of testing images. Recently, for copy and paste forgery detection, deep neural networks (DNN) have been introduced.

In this research, an algorithm that takes into account artifacts caused by copy and paste forgery was offered, which is the most common type of forgery when any digital manipulations like scaling, rotation, or geometrical transformations are performed. Almost many of the deep learning algorithms for detecting copied and pasted regions uses convolution layers. Instead, a unique hybrid net that exploits LSTM layer followed by convolution layers for detecting forged region in the tampered image was proposed by us. However, an LSTM_CNN architecture was proposed in a way that it can focus on the local regions of the artifacts and learn to recognize them. Firstly, image patches were extracted from the authentic and tampered images of the dataset used. Then, an end-to-end network for detecting and localizing the digitally manipulated regions is presented by us. Initially, SRM filters were applied to the extracted patches which produce the noise features. Output from filters is transformed linearly so that it can have the same dimensions as the input dimension of the LSTM_CNN model.

For the LSTM layer, an input size of 320 and a hidden layer of 64 was utilized. Also, 9 CNN layers have been used along with max pooling. Finally, an SVM classifier was used for the classification of forged and pristine parts of the image. For developing our hybrid network, convolutional layers together with LSTM, batch normalization, and max-pooling was used. In the next section, the technical issues of the proposed hybrid architecture employed for the detection of pixel-level copy and paste (move) forgery detection shall be discussed.

The remaining sections of the paper are organised as follows: In Section 2, Digital Image Forgeries-related work carried out by researchers will be highlighted. In Section 3, a detailed description is provided for our end-to-end machine so as to locate the tampered part of an image. In Section 4, our experimental results as well as discussion are presented, in Section 5, the Training, in Section 6, Testing and finally, conclusion and future scopes are presented in Section 7.

2. Related Work

Various methods for forgery detection in an image including copy-paste(move) forgery detection, object removal and image splicing, JPEG artifacts, the machine learning algorithms as well as convolutional neural networks have been proposed from which a few techniques have been discussed in this paper. In the recent years much of research is carried out for the detection of various types of manipulations or tempering which include resampling, content-manipulations in the image, JPEG artifacts and other different types of tempering. In this section, will be discussed in brief, few existing techniques used previously for the detection of Copy-paste (Move) Forgeries in digital images.

Malviya and Ladhake [3] proposed a technique that extracted features of the forged image using the colour information of the images. They utilized Auto Color Correlogram (ACC) which is a feature extraction technique for content-based image retrieval. ACC is a low complexity technique for extracting features from an image. Hence, they obtained feature vectors from the forged image with the help of ACC. Forged portion of the image is detected with considerable accuracy using Color Moments, HSV color space and Auto Color Correlogram.

Li et al. [4] proposed a framework in which they employed a block-based method to segment the images into non-overlapped patches and discovered suspicious matches by matching these patches utilizing a transform matrix in the first stage. Later, confirmed the existence of copy move image forgery by refining the transform matrix in second stage.

Sridevi et al. [5] proposed an algorithm that could detect forgeries for real-time applications. They had converted the input image to grayscale while dividing it into overlapping blocks of fixed size. From each block, they extracted intensity features. For storing the location of each block, two additional columns were utilized. Using radix sort, lexicographical sorting was applied to these feature vectors and forged block was localized. But their algorithm could not be applied over color images.

Amerini et al. [6] proposed a SIFT feature-based method for detecting manipulations in an image. Their method individualized the tampered regions and was capable of detecting multiple cloned regions. Bianchi and Piva [7] and Luo et al. [8] presented an approach that exploits JPEG blocking artifacts for detection of manipulated regions. An algorithm based on rotation-invariant features that were computed densely on an image was proposed for accurate detection and localization of copy-paste forgeries. To achieve good results on the forged regions that were geometrically transformed before pasting, they utilized invariant features like Circular Harmonic transforms [9].

Huang and and Ciou [10] extracted key points along with their descriptors with the help of the scale-invariant feature transform (SIFT) algorithm. Using descriptors, they matched the pairs by calculation of similarity between key points. They grouped these matching pairs on the basis of geometric constraints and spatial distance via Helmert transformation for obtaining the coarse forgery regions. To obtain final results, they refine the coarse forgery regions. They have chosen CMH3 and D2 datasets for their experimental work and their algorithm exhibited good results for scaled, rotated, and compressed forged regions.

Armas Vega et al. [11] proposed a technique for copy and paste forgery detection using the discrete cosine transform. Using DCT, they acquired the transfer vectors and grouped them. With the help of a tolerance threshold, they determined the forged region in a tampered image. They used CASIA TIDE v2.0, CMFD GRIP, CoMoFoD datasets for their experimental work.

Park and Choeh [12] proposed a technique in which they extracted the keypoints and their descriptors using Scale Invariant Feature Transform (SIFT). Further they performed an improvised matching operation capable of multiple forgery detection. Their algorithm was robust against geometric transformation and additive white Gaussian noise applied to images.

Recently, it seems to be an increase of interest in the detection of digitally manipulated images by the implementation of many machine learning algorithms

as well as computer vision. Bappy et al. [13] proposed a new algorithm to detect and localize manipulated regions in an image, utilizing both spatial contexts as well as frequency-domain features. They exploited encoder-decoder network and Long-Short Term Memory (LSTM) cells that segmented tampered regions from non-tampered regions. They also used resampling features for capturing artifacts like upsampling/ downsampling, JPEG quality loss, shearing, and rotation. Wu et al. [2] proposed a two-branch deep learning model along with a fusion module capable of localizing potential tampered regions with the help of visual artifacts and copy and pasted regions through visual similarities. Their algorithm was capable of localizing source and target portions of the digital images.

Table 1 gives comparison of different CMFD techniques, its results and analysis. From Table 1, it can be seen that various algorithms have been proposed earlier based on block-based, key-point based and, Deep learning techniques. However, a technique capable of detection in images that have undergone various attacks have been proposed. For this, the CoMoFoD dataset has been utilized and a detailed experimental evaluation for all the images separately has been done. As discussed earlier, the block-based methods and key-point-based methods are highly dependent on a threshold value that needs to be computed experimentally for feature matching reducing the overall accuracy and robustness of the algorithm [1]. Hence, a hybrid LSTM-CNN based method of forgery detection was proposed and discussed in detail in Section 3.

Table 1. Comparison of different CMFD methods and its performance analysis.

Ref.	Used method	Observations	Accuracy/ Precision
[3]	**Feature extraction using with Color Moments, AutoColor Correlogram, and HSV color space**	Good results against scaled and translated images.	90%
[4]	**Segmentation based method**	Effective in detection of most CMF attacks but high time complexity	average precision 86%
[5]	**parallel block matching algorithm**	Effective in various CMF attacks with reduction in execution time.	Not available
[6]	**Scale Invariant Features Transform (SIFT) based detection method/ g2NN**	Effective for detection of multiple clones in an image.	TPR of 93%
[7]	**DCT coefficients**	Not good for double JPEG compressed image.	94.5%
[9]	**PatchMatch algorithm and Fast approximate nearest-neighbor search algorithm**	Robust for various type of geometrical distortions with reduced time complexity.	Not available
[13]	**Encoder-decoder network and Long-Short Term Memory cells for image segmentation**	classifies the manipulated and non-manipulated regions effectively.	Average precision 93.4%
[2]	**DNN architecture, BusterNet**	localizes source/target regions, robust against various attacks.	CoMoFoD-80.49 % and CASIA dataset-76%.
[10]	**discrete cosine transform and transfer vectors**	locate duplicate zone of the image with high precision, less computational complexity.	Average Precision 93%
[11]	**Feature extraction with SIFT, Superpixel segmentation and the Helmert transformation**	robust against JPEG compression and other transformations but yields poor detection against symmetric, recurring, and smooth patterns.	Average precision 98%
[12]	**key points extraction with SIFT Algorithm**	Effective for transformations like rotation, scaling, additive white Gaussian noise, etc.	80%

3. Implementation of the Proposed Work

As depicted in Fig. 1, a novel approach that combines convolution layers along with the LSTM network was used for pixel-level manipulations in the image and finally SVM classification was utilized that resulted in higher accuracy. A unique way of detecting copy-move forgery from forged images has been presented by us. Firstly, the use of SRM filter kernels was done for producing the noise features for better detection and input them to the LSTM-CNN network. Then LSTM-CNN architecture was utilized in a way that it can focus on the local regions of the artifacts and learn to recognize them using image patches that have to be extracted from the dataset used. A 3-stacked layer of LSTM was used as the first layer of our architecture to know the correlation between pristine and manipulated regions in the image at the frequency domain. Then the CNN layer was utilized to capture spatial information. For training, our fully connected layers with softmax that learns the parameters in the proposed net by back-propagation with the use of ground-truth mask information was used. Finally, using SVM classifier for classification as it performs well for binary classification, pixel-level detection of copied and pasted regions of a digital image. The technical particulars of proposed architecture for the detection of copy-move forgery will discussed in detail in the following section.

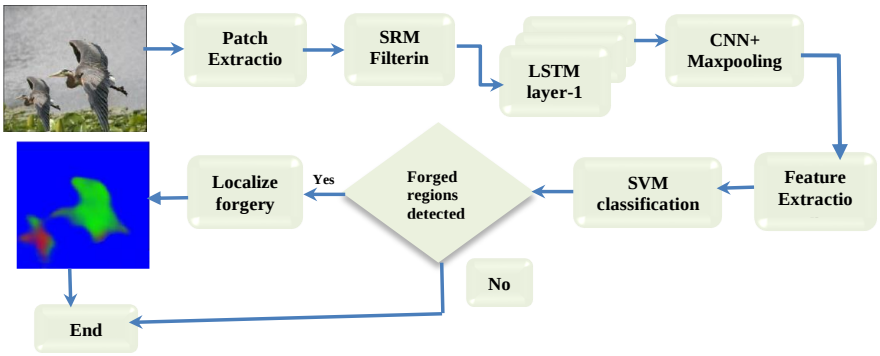


Fig. 1. Block diagram of the proposed method.

Initially patches with rotation and without rotation are extracted from the original as well as forged images. Secondly, in training phase SRM filters are applied to these extracted patches. It outputs 30×3×5×5 filter tensor which goes as an input to 3-stacked LSTM layer and CNN layer with maxpooling as depicted in Fig. 2 of proposed implementation. Training phase returns a feature map after final convolutional layer. Training done with 150 epochs, learning rate of 0.005 and batch size of 64 is used for optimal results. Training loss and training accuracy files are generated, and trained model's learned parameters are saved. Cross-entropy loss is used as loss function and SGD optimizer is utilized in our experimental work. For testing, features are extracted from the patches and feature vectors are generated. Finally, SVM classification yields the confusion matrix as well as misclassified data. If image is tampered, forged region is localized. A detailed explanation of proposed implementation blocks is described further.

Here, a detailed explanation of proposed end-to-end network that is used to detect and localize the tampered regions in digitally forged images is presented. Fig. 2 depicts the block schematic of the proposed system. Description of all the steps used in the framework as given:

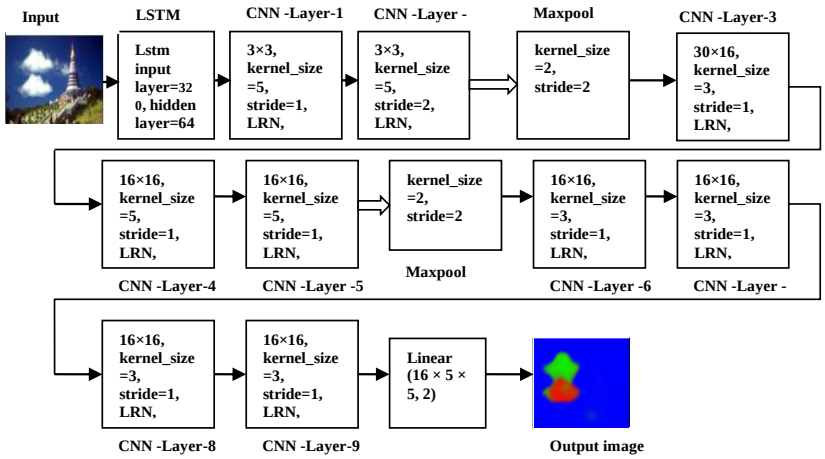


Fig. 2. Architecture of the proposed LSTM-CNN network. Local response normalization is abbreviated as LRN.

3.1. Patch extraction

Firstly, 128 x 128 overlapping patches of the tampered image as well as the pristine image were extracted. Also, patches with rotation of 90,180 and 270 are extracted for each of these authentic and tampered images. For each tampered image, two random patches from the forged region were extracted. Also, from each tampered image, two random patches from the forged region are extracted. Extraction of rotated patches was done from both tampered and authentic images to further get rotation- invariant features from the images whenever needed as shown in Fig. 3.

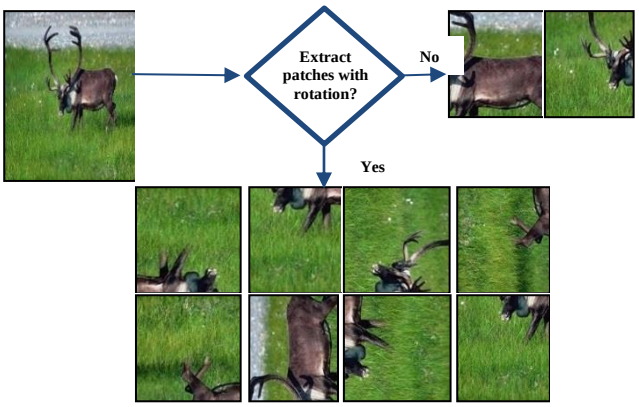


Fig. 3. Extracted 128×128 overlapping patches from original image.

Next, for each patch generated, features were computed on these patches as the initial step of the process. These features are further utilized to identify any resampling that has been applied to the patch. From this, at all the parts at which the patches were extracted, a multi-channel feature map is produced, one channel per resampling characteristic. A correlation between points in the feature map that

represents the patch centered at that pixel and a pixel from the original image and can be interpreted by us. These patches are densely extracted overlapping patches (stride of 8). Finally, by post-processing this feature map, it is further used for producing a binarized localization map and a pixel-level detection score. While selecting patch size, larger patch sizes yield more detection of resampling rather than the smaller size. Selection of 128×128 sizes is done such that they can detect reasonably well. For computational efficiency, extraction of patches with a stride of 8 is done.

3.2. SRM filtering

When some image part is copied and then pasted in some part of the same image, the noise features among the source as well as target images are not likely to match. For proposed work, the RGB image is converted to the noise domain first for the use of these local noise features. There exist several various ways of doing this. However, taking into reference, steganalysis rich model (SRM) for classification of manipulation [14], SRM filter kernels were used for producing noise features and input them to the LSTM-CNN network, which helps in improving the generalization ability.

Moreover, it accelerates the convergence of the network. Based on previous studies [14], it is easier to capture the artifact introduced by various image processing operations in an image residual domain. Thus, the model proposed here first transforms the input image into residuals with some high pass filters.

In proposed experiment, 30 basic filters are utilized and nonlinear operations like maximum and minimum from the neighbouring pixels, the basic noise features are given by SRM. It produces the final features by quantifying and truncating the output from these filters and by extracting the nearer co-occurrence data. These features can be considered as a local noise descriptor. The size of the kernel of the SRM filter layer is 5×5×3 and the output channel size are 3. Further, the output features from these filters are fed into the LSTM-CNN layer. Using 30x5x5 different SRM high pass filters as input, 30×3×5×5 tensor was created. The filters used by us are given in Table 2.

Table 2. SRM filters used in proposed work.

1'st Order filter	2'nd Order filter	3'rd Order filter	3×3 SQUARE	3×3 EDGE	5×5 EDGE	5×5 SQUARE
[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[-1, 2, -2, 2, -1]	[-1, 2, -2, 2, -1]
[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 1, 0, 0]	[0, -1, -2, -1, 0]	[0, -1, 2, -1, 0]	[2, -6, 8, -6, 2]	[2, -6, 8, -6, 2]
[0, 0, -1, 1, 0]	[0, 1, -2, 1, 0]	[0, 1, -3, 1, 0]	[0, 2, -4, 2, 0]	[0, 2, -4, 2, 0]	[-2,8, -12, 8, -2]	[-2, 8, -12, 8, -2]
[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 1, 0, 0, 0]	[0, -1, -2, -1, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[2, -6, 8, -6, 2]
[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[-1, 2, -2, 2, -1]

3.3. The LSTM_CNN network

The experimental work has been carried out using the LSTM_CNN network, using Pytorch [15], and it performed remarkably better for the detection of rotations, noise

addition, upsampling, downsampling, and differences in JPEG compression quality factors apart from all other types of tempering. Also, the experimental results of proposed method gave better accuracy for patches that were extracted by rotating. The experimental results on both the CASIA dataset and the CoMoFoD dataset, with various types of images are discussed in Section 4. The comparison graph of the proposed model for both the datasets with and without rotating the patches is depicted by plotting the graph in Section 4 of this paper.

The natural statistics of an image are distorted whenever it is manipulated, especially in the boundary region. The LSTM_CNN architecture that focuses on the local regions of the artifacts and learns to recognize them was used. For this, image patches of size 128×128×3 have to be extracted from the dataset used. It means that there is a 128×128 patch for every color channel. Extraction was performed by applying a patched-size sliding window with a stride equal to 64 for the whole image. Following that, the tampered patches are discriminated against the non-tampered ones. As far as the tampered patches are concerned, a comparison of each patch is done with the equivalent patch (from the same region of the image) of the mask of this image and the ones that contain part of the tampered region are kept. From input images, LSTM extracts the features. This feature vector is then transformed linearly to keep the same dimension as the input dimension of the CNN network.

LSTM network is used for learning correlation between blocks of resampling features. So, for using LSTM, firstly the 2D resampling feature map is divided into blocks. It can be done by splitting into 8×8 blocks, and every block has an 8×8 (total 64 pixels) size. Each block is fed to each cell of the LSTM sequentially and learning distance block dependency of the image blocks. Correlation of the neighbouring blocks with the current block is done by the LSTM cells. In proposed approach, an LSTM network having 3 stacked layers and 64-time steps was used. In the last layer, a 256-d feature vector is provided by each cell which is supplied to the softmax classifier.

For learning the boundary transformation amongst different blocks, LSTM was used. It gives the distinct features that discriminate between tampered and non-tampered patches. In this section, all the parameters of an LSTM cell will be discussed briefly. For building an LSTM network, a cell is the most important entity. In the cell, the flow of information is controlled by gates. There are three gates (1) input gate, (2) forget gate, and (3) output gate that link a cell to neighbouring cell. Let us denote cell state as C_t, output state as Z_t, and current cell as t. These gates control the cell state and output states. Every gate has a value that ranges from 0 to 1, wherein the sigmoid function is used for activation and they decide the amount of information to be passed through. Higher the value, the more information flows. Each of these cells produces a new candidate cell state $\bar{C}_t$.

The updated cell state C_t with the use of previous cell state and $\bar{C}_{t-1}$ and $\bar{C}_t$ can be written as [16],

$$C_t = f_t \circ C_{t-1} + i_t \circ \bar{C}_t \quad (1)$$

where $\circ$ signifies the pointwise multiplication. Finally, the output of the current cell h_t is obtained, and represented as [16],

$$h_t = o_t \circ tanh(C_t) \quad (2)$$

In Eqs. (1) and (2), *f*, *i*, and *o* denotes forget, input, and output gates. The output from the LSTM layer is directly provided to the convolutional layer.

For proposed experimental work, any kind of resampling detected in the extracted patch can be differentiated by a 3-stacked layer of an LSTM network and a CNN layer. The SVM classifier is trained for checking any kind of resampling present in the patch. The six resampling characteristics are JPEG compression, rotation clockwise, rotation counter-clockwise, or any kind of affine transformation in the image. To train the model for every task, two standard datasets of raw uncompressed images which are available publicly are used. The first CASIA TIDEv2.0 dataset [17] contains 7491 pristine and 5123 manipulated colour images and the CoMoFoD dataset [18] that contains 200 base images that were forged as well as 25 categories (totally 5000 images). Each of these categories is made through the application of post-processing/attacks to the base images categories for hiding the manipulations (e.g., Noise addition JPEG compression, etc.) done to an image. Details about the attacks descriptions and their settings can be found in [18].

About 20,064 patches without rotation and almost double that with the rotation that was extracted from about 10,000 images from the CoMoFoD dataset. Similarly, for CASIA dataset about 41416 patches without rotation and double of that with rotation and found out those rotations in patches give better accuracy. A set parameter, like multiple JPEG compressions, noise addition, and affine transformations, etc., that are generated randomly is used for the transformation of few patches. Half of the dataset should include these transformations while the other half should not.

Mutually exclusive classifiers are not used, and they are trained independently. An artificial neural network containing nine hidden layers and an LSTM layer gives the best classification results for this task. Finally, a mask that shows the tampered regions with the use of filtered resampling feature maps using the proposed hybrid model was acquired.

The proposed technique that consists of the LSTM- CNN model as discussed earlier in Section 3. The proposed model consists of a 3-stacked LSTM layer with 400 input sizes and 64 hidden sizes, which is followed by a CNN network which consists of 9 convolutional layers, 2 max-pooling layers as well as the fully-connected layer along with a softmax classifier. The extracted patches of 128×128×3 (128×128 patch, 3 color channels) size were input to the LSTM. The first and second convolutional layers have 3 kernels of 5×5 with stride 1 and 2 respectively, the third layer has 30 kernels of 3×3 with stride 1 while all other layers have 16 filters (kernels) of 3×3 size.

Rectified Linear Units (ReLU) as an activation function was applied to neurons. With the help of these, they can selectively respond to useful signals of the input [19]. The second and fifth convolutional layers are followed by a max-pooling with a filter of size 2×2 that discards 50% of the activations and resizes the input spatially. This can be done as a max-pooling operation that can help in retaining more texture features and it also improves the convergence performance [20].

Also, for improving generalization, local response normalization (LRN) is applied to the feature maps before the pooling layer in order to normalize the central value in each neighbourhood of the surrounding pixel values. Finally, the 400-D features (5×5×16) that are extracted, passes to the fully-connected layer softmax

classifier, and a "dropout" [17] with the probability of 0.5 is applied that sets to zero the neurons in the fully-connected layer.

3.4. Convolutional neural network

As discussed in Section 3, a convolutional neural network with 9 convolutional layers, 2 max-pooling layers, and a fully connected layer having a softmax classifier was proposed. The first and second convolutional layers have 3 kernels of 5×5 with stride 1 and 2 respectively, the third layer has 30 kernels of 3×3 with stride 1 while all other layers have 16 kernels of size 3×3. Local Response Normalization (LRN) was used to square-normalize the pixel values of the feature map and Rectified Linear Units (ReLU) were used as activation function to neurons.

3.5. Soft-max layer

In proposed network, the patch classification task is done using a softmax layer. For the accomplishment of this task, prediction of labels is done at the final layer of the CNN network. The features are obtained from the given patch; these features are used for the prediction of the manipulated class with the use of the softmax function. L*et* $\mathcal{W}$ be the parameter associated with the feature. Then the softmax function ($\mathcal{F}$) can be written as [16],

$$P(y|k) = \frac{\exp^{(\mathrm{W^t})\mathcal{F}}}{\sum_{k=1}^{N_c} \exp^{(W^t)\mathcal{F}}} \tag{3}$$

where N_c denotes the number of classes (ϵ $R^{2\times1}$) - tampered vs non-tampered. W_L^k indicates the weight vector associated with class *k*. Eq. (3) was used for computing the probability distribution over various classes. Now, by maximizing $P(y|k)$ with respect to *k* the labels can be predicted, and further the predicted labels can be obtained by [16],

$$\hat{y} = \underset{k}{arg\,max}\ P(y|k) \tag{4}$$

4. Training the Network

In this paper, to determine whether an image is tampered or not, patch classification is performed. Given ground-truth patch labels, firstly, cross-entropy loss is computed for this task. The patches extracted in the patch extraction phase are used as input to the stacked LSTM and 9 convolution 2D layers, 2 max-pooling layers for feature extraction. Finally, the softmax layer is used for the prediction of the manipulated class. As an optimizer, stochastic gradient descent was used and a learning rate of 0.005 was used. Training time was quite fast for learning rate 0.005, and reasonably fast from 0.5 and 0.1. Also, for the proposed work, it yielded a better accuracy rather than using any other (e.g., 0.1, 0.5, etc.) learning rates.

Patch classification

Prediction of patch labels is done at the end of the LSTM network. Let us denote, $\theta^p = [\theta^1, \mathcal{W}]$, the weight vector that is associated with the classification of the patch. Here, θ^1 comprises the parameter of the first two convolution layers and LSTM layers. $\mathcal{W}$ is the parameter of the softmax layer of patch classifier. Now, computation of cross-entropy loss for the patch classifier can be done as [16],

$$L_p(\theta_p) = -\frac{1}{M_p}\sum_{j=1}^{M_p}\sum_{k=1}^{N_p} 1\,(Y^j = k \backslash x^j; \theta \quad (5)$$

where 1 (.) is the indicator function, and it is equal to 1 if j = k, else it is 0. Y^j and X^j indicates the label (tampered or pristine) associated with patch and the feature of the sample j. M_p indicates the number of patches.

5. Testing the network

The testing of the image was performed using an SVM classifier which classifies according to True positive, false positive, true negative, and false negative based on the features extracted by our training model given as input.

SVM classifier

After the training of the network, the next step is to train the Support Vector Machine (SVM) classifier. Determination of the decision boundaries is done by SVM in the training step. SVM classification is done on basis of the decision plane concept which defines the decision boundaries. This decision plane separates out a set of objects that belong to various classes. Support vectors that define the widest separation of classes are found by SVM. For classification purposes, first extraction of every possible 128x128 patch was done from both the original and the tampered images using a sliding window with a stride of 64 to scan the whole image. This process results in 2 new patches per image which are passed through the LSTM_CNN network resulting in n feature representations *Y_i (400-D).*

These representations need to be fused into a single $\hat{Y}[k]$ representation for each image before being passed as an input to the SVM. Max pooling was applied to each dimension of Y_i over all the n patches extracted from each image. The resulting 400-D feature vector is then used by the SVM to classify images either as pristine or tampered.

If the feature map in the convolution layer n is denoted by $F^n(X)$, kernel and bias by W^n and B^n respectively, the convolutional layer can be computed using the following formula [21]:

$$F^n(X) = pooling\,(f^n(F^{n-1}(X) * W^n + B^n) \quad (6)$$

where $F^0(X) = X$, is the input data, $f^n(\cdot)$ is a non-linear activation function applied to every element of the input and the pooling operation which reduces the dimensions of the data via a max or mean operation.

Radial basis function (RBF) was utilized by us as the SVM kernel. For dividing a dataset into training and testing sets, ten-fold cross-validation was employed which is widely used and most suitable method [22]. In this method, entire dataset is divided into 10 equal-sized mutually exclusive folds and the ratio of the real and forged images in each fold was kept roughly the same as in the total dataset. Each time nine folds were used for training and validation, and one-fold was used as the test set.

6. Results and Discussion

6.1. Performance Criteria

The performance of the proposed technique was carefully observed through the following [19]:

True Positives (TP) - Image is actually forged and identified as forged,
False Positives (FP) - Image is actually pristine but identified as forged,
True Negatives (TN) - Image is pristine and identified as pristine, and,
False Negatives (FN) - Image is actually forged but identified as pristine

Accuracy is the total number correctly predicted images divided by the total number of images in the dataset. The best accuracy is 1.0, whereas the worst is 0.0.

Accuracy = (TP + TN) / (TP + TP + FP + FN).

In experimental work of proposed work, calculation was done considering tampered images as positive class and pristine images as negative class. Hence, *TP* or *FP* represents the forged class, while original/pristine images are represented by *TN* or *FN*.

Evaluation Data

For evaluation, the standard CoMoFoD dataset [18], which contains 200 base forged images and 25 categories (total 5000 images) were used. All CoMoFoD database contains 260 image sets, out of which 200 images belonging to the small image category (512×512) were used. Following transformations are done with the images [18]:

Translation where the copied part is translated to the new location of the image without doing any transformation on it, rotation where a copied part first undergoes rotation, scaling where the copied part is first scaled and then pasted, distortion where the copied part is first distorted and then pasted to a new location and combination where two or more transformations are applied on the copied portion before pasting it to the new location.

Moreover, post-processing like JPEG compression, Noise adding, Image blurring, Brightness change, Color reduction, Contrast adjustments are applied on all pristine as well as forged images. For experimental work, a 3.2 GHz Intel Core i5 processor with integrated Intel UHD Graphics 610 and 8GB RAM was utilized. Python 3.7.2 with Pycharm 2019.3.5.was used for experimental work.

To evaluate proposed CNN-LSTM network's performance, image data are needed to be tested with ground truth masks. The CASIA and the CoMoFoD datasets were used for experiments. Further, the performance of the model was tested by extracting patches without rotation and with rotation and found out that with patch rotation our model yields accuracy of 94.8% with patch rotation and 82.8% without rotating the patches for CASIA dataset.

Further, it yields 84.8% accuracy for CoMoFoD dataset with patch rotation and 73.5% accuracy without patch rotation. From our experimental work, it was derived that proposed model yields a better performance with the CASIA dataset in contrast to the CoMoFoD dataset. Figure 4 depicts the detection of forged regions for the CASIA and CoMoFoD dataset.

The CASIA and the CoMoFoD datasets were used for testing the efficiency of the hybrid LSTM_CNN based approach. Comparison of CNN-LSTM methods on both the datasets can be seen in Table 3. The classification result is depicted in Table 3. The results in Table 3 and Fig. 4 confirm that the proposed hybrid method is much more effective for the classification of manipulated regions.

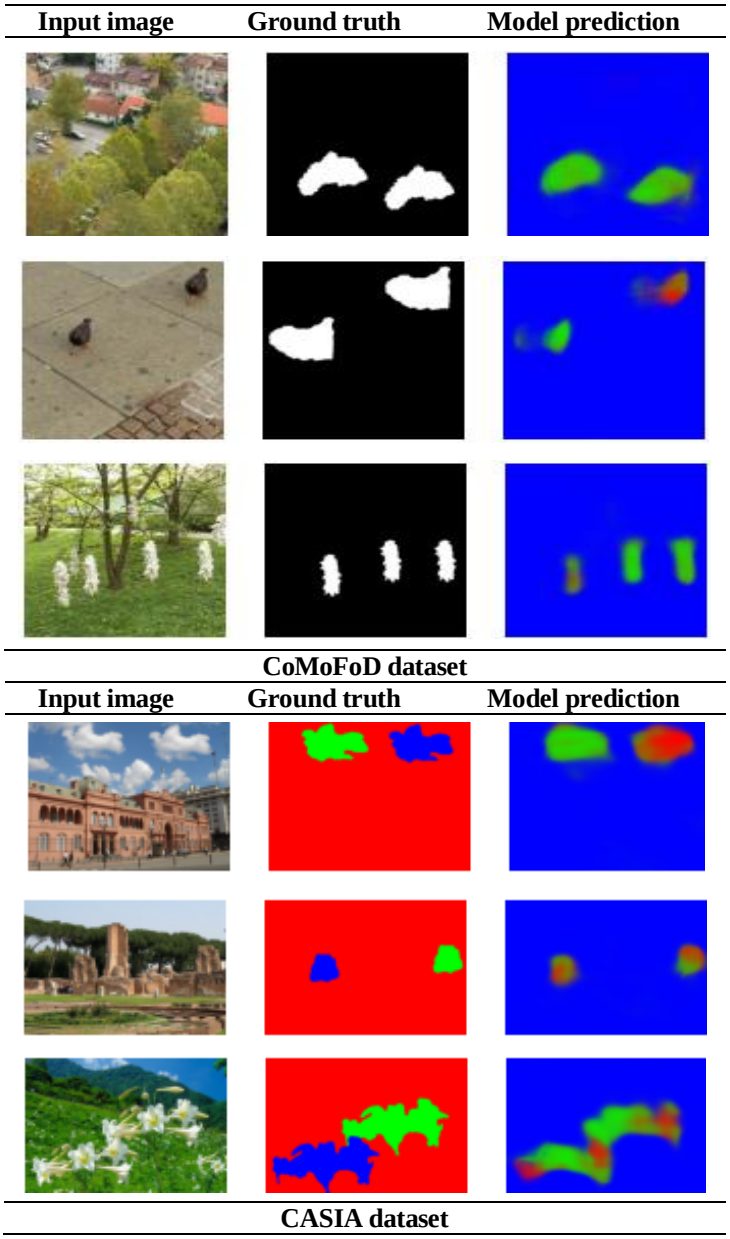
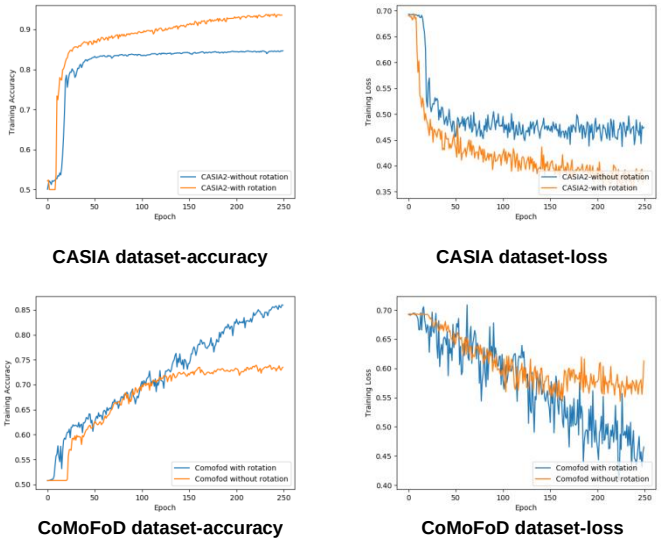


Fig. 4. Detection of forged regions for the CASIA and CoMoFoD dataset.

Table 3. Performance analysis (accuracy) using SVM classifier.

Proposed_Method(HybridCNN-LSTM)	CoMoFoD Dataset	CASIA Dataset
Accuracy (with patch rotation)	84.8%	94.8%
Accuracy (without patch rotation)	73.5%	82.8%

The training accuracy and loss for the CASIA and CoMoFoD datasets have been presented in Fig. 5. The graph in Fig. 5 shows the comparison between the CASIA and CoMoFoD datasets (with itself) in terms of accuracy and loss function with rotation and without patch rotation. It clearly depicts that the performance of the CASIA dataset gives better accuracy over the CoMoFoD dataset for our approach.



CASIA dataset-accuracy **CASIA dataset-loss**

CoMoFoD dataset-accuracy **CoMoFoD dataset-loss**

Fig. 5. Comparison of training accuracy and loss function.

From our experiments, the trend in AUC 0.96 on 128×128 patches of the image can be observed in Fig. 5. Further, an increase in accuracy can be noticed as patch size increases. In addition, by utilizing different patch sizes for the same datasets and comparing the same, the best patch size can be obtained. Further, a comparison of the proposed work can be done with the deep network in [2], which has a two-branch architecture for the detection and manipulation of forged regions in an image. Also, SIFT based method [4] was utilized for further comparison; and it was found that the proposed algorithm has an increase in accuracy of almost 5% with rotated patches for the CoMoFoD dataset. The CNN-based [2] technique and SIFT-based [4] techniques were used for comparison with propose work as both the techniques use the CoMoFoD dataset having images of various types.

Use of an SGD optimizer with a learning rate of 0.005 and cross-entropy loss was done and good results were achieved on training data for 150 epochs. For pixel-level evaluation, computation of True Positive, True Negative, False Positive, and False Negative numbers was done over the whole dataset as well as individual image types. The testing image, for which any number of pixels are detected as forged is labelled as forged. Furthermore, for evaluating overall performance, the accuracy and loss function were used. This evaluates the overall ability of the hybrid network to distinguish between the two classes. Two methods, a deep learning end-to-end network [2] and SIFT [4] based detection was used for comparison as the baseline.

For evaluation of the robustness of the proposed hybrid network against all the attacks, the proposed method was tested with the baseline method on the CoMoFoD dataset. The proposed hybrid network outperforms the baseline method on overall image datasets. For performance analysis further, the experiments were conducted with entire datasets and compare them with state-of-the-art methods. Table 4 shows the comparison of the proposed work with other work on the CoMoFoD dataset. It is noteworthy that the overall performance of the proposed network has increased by almost 6% on the performance over the entire dataset as shown in Table 4.

Table 4. Comparison of proposed work on CoMoFoD dataset

Technique used	Accuracy (%)
Proposed Method (without rotation)	73.5%
Proposed Method (with rotation)	86.1%
CNN based [2]	80.49%
SIFT based [4]	71.55%

Time complexity for the proposed algorithm. SIFT on 10 images 14.006 seconds, CNN based algorithm is 35.320 seconds and that of proposed algorithm was 735.87 seconds as detection of image forgery include the process of patch extraction, training, feature extraction and testing.

Overall comparison of all the images of CoMoFoD dataset, individually is depicted in Table 5. The results in the Table 5 show the performance in percentage accuracy for different image types. Table 5 depicts the accuracy of various methods under each attack (200 samples each). The results in Table 5 and Fig. 4 confirm that proposed hybrid algorithm is much more effective for the classification of tampered patches.

Table 5. Comparison of results of proposed hybrid network

Performance in percentage Accuracy for different types of transformation applied on images									
Image types	SIFT based [4]	CNN based [2]	Proposed network		Image types	SIFT based [4]	CNN based [2]	Proposed network	
			without rotation	with rotation				without rotation	with rotation
BC1	70.61	61.25	75.75	80	JC1	58.33	59.00	78.78	82.22
BC2	73.56	62.75	77.18	78.49	JC2	61.22	56.05	82.62	87.81
BC3	74.36	60.50	83	83.86	JC3	66.60	60.25	86.5	94.37
CA1	63.35	62.25	86.62	90.68	JC4	67.64	59.50	86	81.9
CA2	61.33	61.00	81.12	84.43	JC5	62.43	61.00	71.62	86.56
CA3	63.50	61.50	75	85.62	JC6	60.53	61.25	83.5	82.75
CR1	69.13	61.50	85.24	86.81	JC7	62.25	60.36	84	81.93
CR2	67.60	61.50	79.5	81.62	JC8	63.77	60.90	85.24	88.31
CR3	67.75	61.50	74.125	78.09	JC9	62.20	61.00	81.25	82.18
IB1	57.52	60.50	68.22	71.43	NA1	58.70	58.25	81.3	86.5
IB2	60.43	57.25	67.81	70	NA2	59.90	61.68	78.9	85.6
IB3	56.50	57.25	65.23	68.73	NA3	60	61.25	76.4	79.88

By comparison of proposed work with CNN based method and SIFT based method, it can be observed that proposed work gives good detection with JPEG compressed images as depicted in Fig. 6.

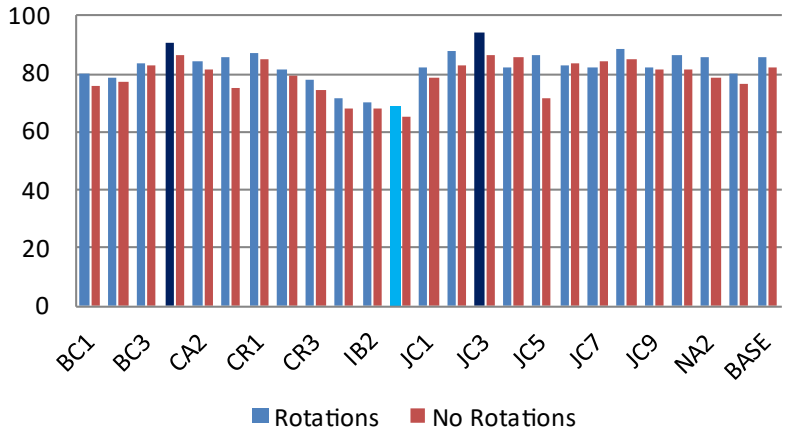


Fig. 6. Accuracy graph for different images of CoMoFoD dataset.

Further, a small dataset of 50 images with different forgery types was created for testing our model. The picture in Fig. 7 (Original Image) was forged by copying a part of the picture and pasting it in the image. It was tested with proposed model and it gives correct detection as shown in Fig. 7 (Manipulated regions). Ground Truth of the same is given in Fig. 7 (Ground Truth). An accuracy of 82.91% was achieved for the given dataset (rotated patches) with batch size of 128 and 100 epochs. Learning rate of 0.001 was utilized. As more data yields better results for LSTM-CNN technique, new forged and original images will be added to the dataset.

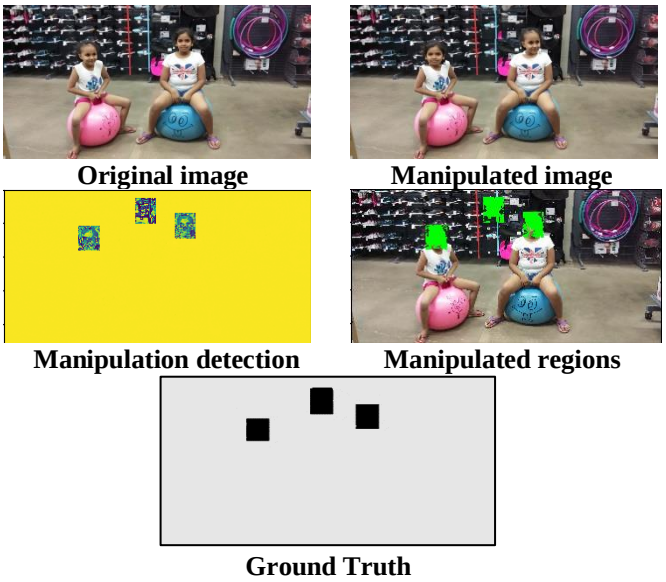


Fig. 7. Proposed model detecting the manipulated regions (i.e., Faces exchanged, and background copied from the same image and pasting in the image).

7. Conclusion and Future Scope

In this paper, a combined method to the detection and localization of copied and pasted regions in digital images is presented. Experimental results depict that combining LSTM with Convolutional Neural Network is much more effective and accurate in the detection of tampered features from a digital image as it yields 5% more accuracy than other method.

Moreover, by rotation of patches further increases the overall accuracy of the network as seen in Tables 4 and 5. Also, from the results presented in Table 5, it can be seen that it gives good detection for JPEG compressed image. However, for the proposed work, the time complexity increases in comparison to other two method.

In the future, hybrid network for other kinds of forgery detection in digital images as well as a forgery in the video can be used. Also, further work can be carried out towards reduction of time complexity for the hybrid method.

Nomenclatures

$\bar{C}_t$	New candidate cell state
$\bar{C}_{t-1}$	Previous cell state
$F^n(X)$	Feature map in the convolution layer n
h_t	Current cell
$L_p(\theta_p)$	Cross-entropy loss
N_c	Number of classes
o_t	Output gate layer
W	Parameter at the softmax layer of patch classification

Greek Symbols

θ^1	parameters of the LSTM layers and the convolution layers.
θ^p	weight vector associated with classification of patches.

Abbreviations

ANN	Neural Network
CMFD	Copy-Move Forgery Detection
CNN	Convolution Neural Network
FP	False Positive
TN	True Negative
TP	True Positive

References

1. Dixit, A.; and Gupta, R.K. (2016), Copy-move image forgery detection a review, *International Journal of Image, Graphics and Signal Processing* (*IJIGSP*), 8(6), 29-40.
2. Wu, Y.; Abd-Almageed, W.; and Natarajan, P. (2018). BusterNet: Detecting copy-move image forgery with source/target localization. *European Conference on Computer Vision* (*ECCV*), *Springer*,1-17.

3. Malviya, A.V.; and Ladhake, S.A. (2016). Copy move forgery detection using low complexity feature extraction. 2015 *IEEE UP Section Conference on Electrical Computer and Electronics* (*UPCON*), Allahabad, India, 383-390.
4. Li, J.; Li, X.; Yang, B.; and Sun, X. (2015). Segmentation-based image copy-move forgery detection scheme. *IEEE Transactions on Information Forensics and Security*, 10(3), 507-518.
5. Sridevi, M.; Mala, C.; and Sandeep, S. (2012). Copy-move image forgery detection in a parallel environment. *Computer Science & Information Technology* (*CS & IT*), 52, 19-29.
6. Amerini, I.; Ballan, L.; Caldelli, R.; Del Bimbo, A.; and Serra, G. (2011). A siftbased forensic method for copy-move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security*, 6(3), 1099-1110.
7. Bianchi, T.; and Piva, A. (2012). Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3), 1003-1017.
8. Luo, W.; Huang, J.; and Qiu, G. (2010). Jpeg error analysis and its applications to digital image forensics. *IEEE Transactions on Information Forensics and Security*, 5(3), 480-491.
9. Cozzolino, D.; Poggi, G.; and Verdoliva, L. (2015). Efficient dense-field copy- move forgery detection. *IEEE Transactions on Information Forensics and Security*, 10(11), 2284-2297.
10. Huang, H.-Y.; and Ciou, A.-J. (2019). Copy-move forgery detection for image forensics using the superpixel segmentation and the Helmert transformation. *EURASIP Journal on Image and Video Processing*, 68.
11. Armas Vega, E.A.; González Fernández, E.; Sandoval Orozco, A.L.; and Luis Javier, G.V. (2021), Copy-move forgery detection technique based on discrete cosine transform blocks features, *Neural Computing and Applications*, 33, 4713-4727.
12. Park, C.-S.; and Choeh, J.Y. (2018) Fast and robust copy-move forgery detection based on scale-space representation. *Multimedia Tools and Applications*, 77(13), 16795-16811.
13. Bappy, M.J.H.; Simons, C.; Nataraj, L.; Manjunath, B.S.; and Roy-Chowdhury, A. (2019). Hybrid LSTM and encoder-decoder architecture for detection of image forgeries. *Computer Vision and Pattern Recognition*, 28(7), 3286-3300.
14. Pan, X.; and Lyu, S. (2010), Region duplication detection using image feature matching. *IEEE Transactions on Information Forensics and Security*, 5(4), 857-867.
15. Hatcher, W.G.; and Yu, W. (2018). A survey of deep learning: Platforms. applications and emerging research trends. *IEEE Access*, 6, 24411-24432.
16. Bunk, J.; Bappy, J.H.; Mohammed, T.M.; Nataraj, L.; Flenner, A.; Manjunath, B.; Chandrasekaran, S.; Roy-Chowdhury, A.K.; and Peterson, L. (2017). Detection and localization of image forgeries using resampling features and deep learning. *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (*CVPRW*), *Honolulu, HI, USA*, 1881-1889.

17. Dong, J.; Wang, W.; and Tan, T. (2013). CASIA image tampering detection evaluation database. 2013 *IEEE China Summit and International Conference on Signal and Information Processing*, *Beijing*, China, 422-426.
18. Tralic, D.; Zupancic, I.; Grgic, S.; and Grgic, M. (2013). CoMoFoD - new database for copy-move forgery detection. *Proceedings ELMAR*-2013, Zadar, Croatia, 49-54.
19. Ghoneim, S. (2019). Accuracy, recall, precision, f-score & specificity, which to optimize on? Retrived April 2, 2019, from https://medium.com/towards-data-science/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124.
20. Feng, X.; Cox, I.J.; and Doerr, G. (2012). Normalized energy density based forensic detection of resampled images. *IEEE Transactions on Multimedia*, 14(3), 536-545.
21. Rao, Y.; and Ni, J. (2016). A deep learning approach to detection of splicing and copy-move forgeries in images. *IEEE International Workshop on Information Forensics and Security* (*WIFS*), Abu Dhabi, United Arab Emirates 1-6.
22. Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the* 14*th International Joint Conference on Artificial Intelligence*, Vol. 2, Montreal, Canada, 1137-1145.