

# Image Manipulation Detection Through Laterally Linked Pixels and Kernel Algorithms

K. K. Thyagarajan and G. Nirmala\*

RMD Engineering College, Kavaraipettai, Chennai, India

\*Corresponding Author: G. Nirmala. Email: nirmalaganapathy@yahoo.co.in

Received: 17 May 2021; Accepted: 01 July 2021

**Abstract:** In this paper, copy-move forgery in image is detected for single image with multiple manipulations such as blurring, noise addition, gray scale conversion, brightness modifications, rotation, Hu adjustment, color adjustment, contrast changes and JPEG Compression. However, traditional algorithms detect only copy-move attacks in image and never for different manipulation in single image. The proposed LLP (Laterally linked pixel) algorithm has two dimensional arrays and single layer is obtained through unit linking pulsed neural network for detection of copied region and kernel tricks is applied for detection of multiple manipulations in single forged image. LLP algorithm consists of two channels such as feeding component (F-Channel) and linking component (L channel) for linking pixels. LLP algorithm linking pixels detects image with multiple manipulation and copy-move forgery due to one-to-one correspondence between pixel and neuron, where each pixel's intensity is taken as input for F channel of neuron and connected for forgery identification. Furthermore, neuron is connected with neighboring field of neuron by L channel for detecting forged images with multiple manipulations in the image along with copy-move, through kernel trick classifier (KTC). From experimental results, proposed LLP algorithm performs better than traditional algorithms for multiple manipulated copy and paste images. The accuracy obtained through LLP algorithm is about 90% and further forgery detection is improved based on optimized kernel selections in classification algorithm.

**Keywords:** Machine learning; copy move forgery; support vectors; kernel; feature extraction

## 1 Introduction

Recently, free image editing tools available in internet leads to duplication of image and detecting duplication in image is a major problem for many researchers. In this internet world, day-by-day digital photo plays a vital role in various e-commerce applications such as sales and marketing. Furthermore, sharing of digital image in social media is increasing exponentially. However, identifying original image needs efficient software tools based on type of duplication of image such as copy-move, splicing, digital watermarking, digital signature, and image compression and re sampling duplicated images. Among the above duplication of image, copy-move duplication is more in social media, due to availability of many



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

free editing tools and easy to manipulate. Moreover, according to statistics many forensic cases are related to copy-move duplicated images [1]. The traditional algorithms for detecting copy-move forgery performs based on any one of the following procedures such as key-points in pixels, image block based detection, feature vectors, feature matching, decomposing transform co-efficient, testing and training based detection. However, copy-move forgery detection by traditional algorithms is implemented in standard forgery image data sets. The efficiency of traditional algorithm needs to be checked with various software based manipulations in images such as blurring, rotation, translation, JPEG compression, noise addition, brightness change and Hu-adjustment. The software manipulations need to be performed with standard tools such as adobe, spark and pine tools. Furthermore, efficiency of traditional algorithms need to be checked with fabricated multiple manipulations in single image. The fabricated manipulations need to be performed during acquisition of images through the manipulations in camera lens. The traditional algorithm which performs better for standard data sets never performs for image of the same data sets image with multiple manipulations. Similarly, dataset images with fabricated photo manipulations needs efficient and robust algorithm to detect such forensic images.

## 2 Contributions

1. The proposed LLP algorithm performs copy and move detection in single image with multiple manipulations through free online software. The algorithm performs forgery detection in single image with multiple manipulations through single linear linked pixels. The single linear linked pixels differentiate software manipulation in images through the property of automatic feature extraction in CNN.
2. The software manipulations needs feature extractions through dimensional-reduced feature matching to differentiate copy and move pixels for reducing time complexity and improves duplication accuracy.
3. The single and multiple manipulations in single image detection and classification of duplicate and original image is performed through selecting optimum kernel in support vector machines. The kernel selection for the multiple manipulated images is performed through various functions such as cost function, sigmoid, linear, polynomial and radial basis function.

### 2.1 Related Work

From [Tab. 1](#) it is understood that block based and key point based methods uses feature extraction as a separate method and classification based on features are done. This gives a way support vector machines to be combined with LLP algorithm for bringing the improvement in terms of classification accuracy. Another novel idea to find the images with single and multiple manipulations in addition to standard forgery dataset was implemented.

### 2.2 Inferences from Literature Survey

Till now, copy-move forgery detection in high resolution images are performed with various algorithms such as CNN, LBP, SVM, DCT and DWT. However, performance of above algorithms needs to be evaluated for low and medium resolution copy move forgery images. The existing algorithms require change in thresholding levels for DWT, DCT and such level changes in algorithms leads to high computational complexity. Moreover, deep learning algorithms needs change in network architecture and which leads to more false identification due to less number of data sets for learning and training. Furthermore, SVM algorithm needs changes in kernels to detect low and medium resolution images for more accuracy. From traditional algorithms, manipulated images such as blurred, noise added (single) which lead to low and medium resolution images need more accurate selection of kernels or threshold levels. To avoid the

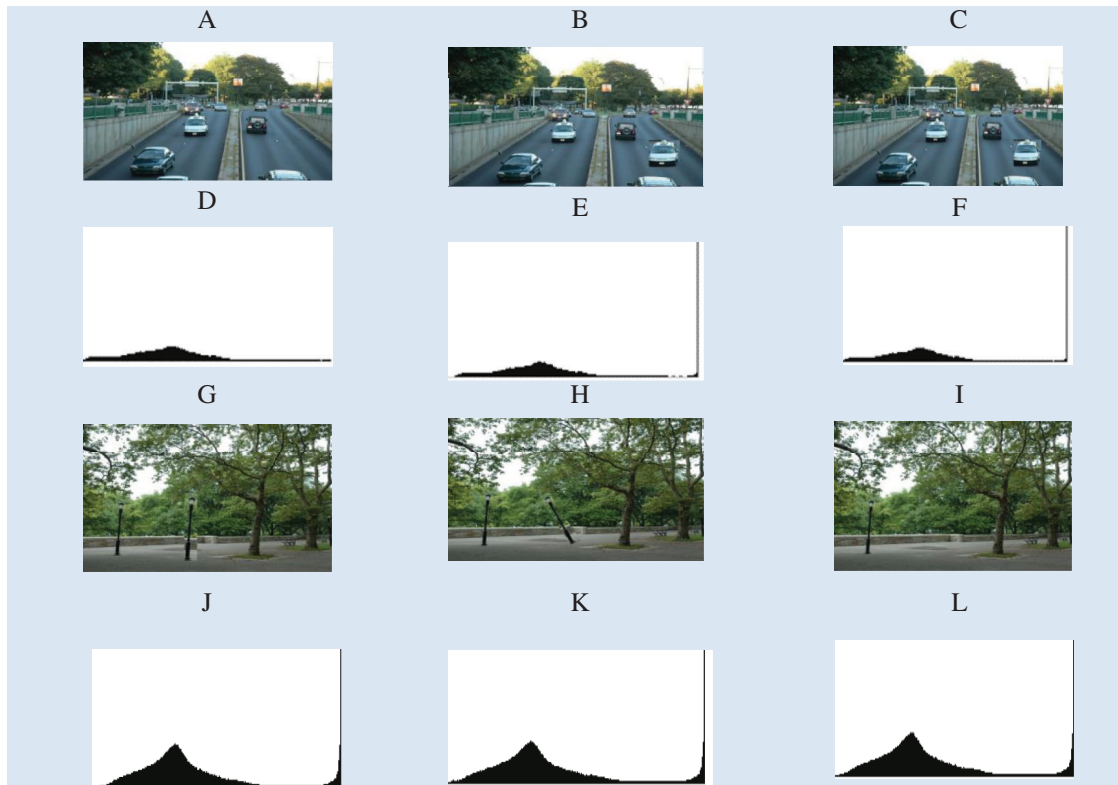
above problem, LLP algorithms are proposed based on selected properties such as linked pixels and kernel modifications. The proposed algorithm discriminates pixels in copy move region with high and low resolutions.

**Table 1:** Literature review

Reference/ Year	Data set	Algorithm	Advantages and disadvantages
2020 [2]	CASIA Columbia Data set	DCT and SVD, SVM and K-means	Accuracy of 90% for manipulation such as rotation, scaling, noise and JPEG Compression included in standard images.
2020 [3]	Lenova, Moto, samsung	SVM and SURF	Performs for manipulation such as blend, scale, joined, bicubic and crop operations in the images of datasets.
2020 [4]	CASIA 2 COLUMBIA	DCT and LBP with Mean for feature extraction and SVM classifier	Improved accuracy of above 92% for combining DCT and LBP algorithms in the images.
[5] 2020	Columbia casia v1.0 casia v2.0	Deep CNN for classification and SVM for block matching using local features to detect the region duplication.	No manipulation in the dataset images.
[6] 2020	Pedestrian Data set INRIA, Diabler ETH	HOG, SVM and NRULBP.	Online Standard Data set with no manipulations.
[7] 2018	COVERAGE COMOFOD	PCNN for Feature extraction and Correlation coefficient for classification.	Standard Dataset with no manipulations.
[8] 2019	COMOFOD, MICC	Brich clustering for classification.	No manipulation in the dataset image.
[9] 2018	CASIA v1	LBP, RLBP, SURF.	Manipulation in image through rotation invariant version of LBP (RLBP) on each key point of SURF to detect the forgery.
[10] 2019	GRIP MICC-F2000	PCNN and a fast local feature detector named Harris-Hessian	No manipulation in the dataset image.
[11] 2011	COVERAGE, GRIP	SIFT.	No manipulation in the dataset image.
[12] 2013	GRIP, MICC-F20, COMOFOD	Listed various methods of forgery detection with the steps involved in it.	No manipulation in the dataset image.
[13] 2018	CASIA v1.0, CASIA v2.0	ZM-polar is used to locate forged region. SVM for classification.	Detection of image manipulation in the image through rotated by 45° and 135°
[14] 1990	COLUMBIA Image splicing data set	Feature linking by PCNN.	No manipulation in the dataset image.
[15] 2018	Casia v1 Casia v2 CoMoFoD COLUMBIA	Review on copy move forgery detection techniques as block based, key point based, active and passive methods.	No manipulation in the dataset image.
[16] 2019	COLUMBIA	Near Duplicate (ND) detection by key point matching and entropy measure to classify.	No manipulation in the dataset image.
[17] 2019	COMOFOD	SVM with compact representation of Kernel.	Efficient classification of larger dataset.
Proposed	GRIP MICC-F220 COMOFOD	LLP and kernel trick.	Forgery image dataset & Standard forgery image dataset with fabricated multiple manipulated images are detected for forgery.

### 2.3 Materials and Methods

The growth of internet technologies increases forgery images and identifying forgery images is a challenging task due multiple manipulations in a single image. Copy-move forged images consist of small the portion copied from different image and pasted in different part of image. Figs. 1A–1L is copy-move forged image from MICC-200 dataset. B is tampered image of A. but both looks similar.



**Figure 1:** (A–L) Copy-move forged image from MICC-200 dataset with histogram

Figs. 1A–1L does not show any difference in original image and forged image through visual interpretation and similarly histogram shows no different in bin for all the images such as forged and original image. In Fig. 1L are a multiple manipulated image and its histogram which never show difference for forged and original image. However, Fig. 1A with single manipulation shows negligible difference in histogram, whereas for multiple manipulation images as in Fig. 1J shows no difference in bins of histogram. The multiple manipulations in image need efficient algorithms for detecting forgery images. Along with images in standard dataset different multiple manipulations of image through different software are also taken as input images. The manipulated images never show difference in image between original and multiple manipulated one.

In proposed LLP block as given above in Fig. 2 initially, input image is converted into  $M \times N$  matrix and from matrix initialize the values of  $S_{ij}$ , decay term for feeding, linking and threshold  $\alpha$   $T$  is obtained. Initial magnitude scaling term is assigned, linking strength is obtained and from number of iterations values such as Feeding Pixel (FP), Feeding component (FC), Output Pixels (OP), Output Pixels (OC), Mean (M), Weight (W) are calculated, updates feeding and linking input for each iteration i.e.,  $F_{ij}[n] = S_{ij}$  Lining field ( $L_{ij}[n] = \sum_{kl} y_{kl}(n-1)$ ) getting  $L_{ij}$  as 0 or 1 and compute for performing LLP,

Compute  $U_{ij} = S_{ij} * (1 + (0.1 * L_{ij}))$ ,  $T_{CI_{ij}} = TPI_{ij} * \exp(-\alpha T) + (5 * YPI_{ij})$ ,  $YPI_{ij} > 0$  then  $L_{ij} = 1$ , else 0 and if  $U_{ij} > T_{CI_{ij}}$  then  $Y_{ij} = 1$  else 0 update the threshold and updates the activity and explained in the further section.

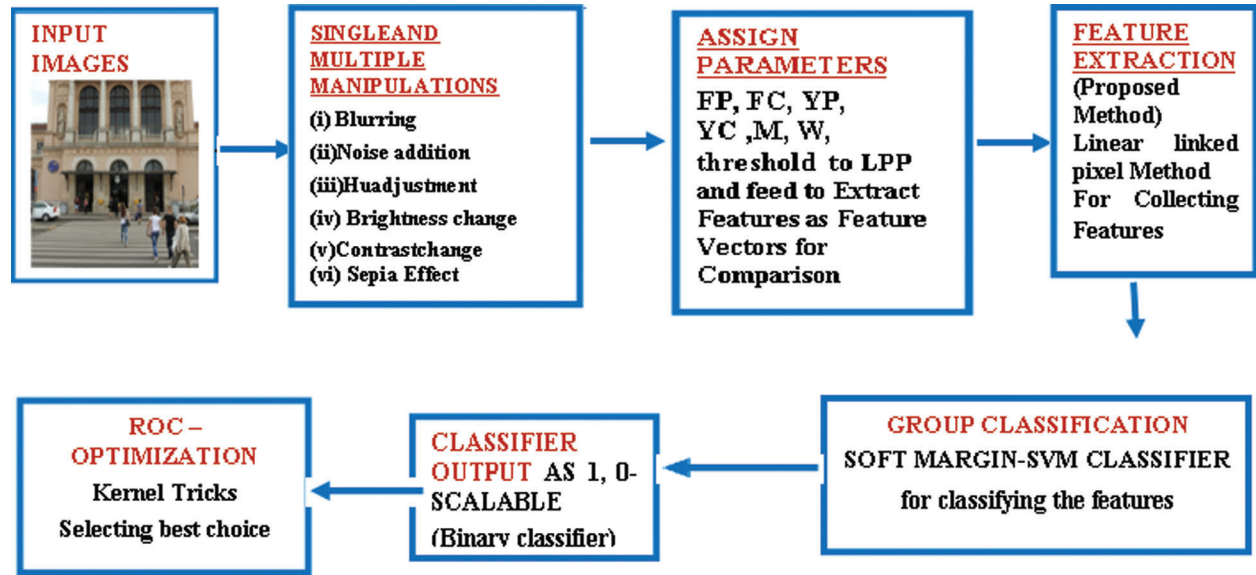


Figure 2: Block diagram of the proposed LLP algorithm

### 3 LLP Algorithm (Linear Linked Pixels Unit Linking PCNN Model)

The model of LLP is represented in the above Fig. 3 with input as  $y$  consisting pixels as rows and columns and feature vectors are extracted as output from the block. In LLP, input signal  $S_{ij}$  (external stimulus) is intensity of pixel at  $(i, j)$  position. LLP is feeding component (input stimuli) to neuron. Each pixel in input image is connected to a neuron. Each neuron connects with outputs of its neighboring neurons, for receiving local stimuli from them. The external and local stimuli are combined (multiplied) in an internal activation system  $U_{ij}$  (membrane potential). At one stage, internal activity is larger than threshold and hence neuron fires, which makes output of the neuron ‘1’. Moreover, threshold decays exponentially in each successive iteration; when its value becomes below internal activity a specific iteration, output will become ‘0’. This creates sequence of 1s and 0s and it is called as time series of pixel (image) created by neuron(s). It is called as temporal series of pulse outputs. The temporal series of pulse outputs contain information of input images and used for various image processing applications, such as image segmentation and feature generation. The linking field value is computed as given below according to neuron model shown in Fig. 1. In this paper, LLP follow the method suggested in [18] for estimating the value of  $L_{ij}[n]$  as given below. If a set of neurons is in neighborhood of neuron  $(i, j)$  and if set is denoted as  $N(i, j)$ , linking value is given by.

$$L_{ij}[n] = \begin{cases} 1 & \sum_{kl \in N(i,j)} Y_{kl}(n-1) > 0 \\ 0 & \sum_{kl \in N(i,j)} Y_{kl}(n-1) = 0 \end{cases} \quad (1)$$

where each  $(k, l)$  is neighbor neuron’s position of center neuron  $(i, j)$ . Eq. (1) indicates, if any one of neighboring neurons fires, then  $L_{ij}(n) = 1$  otherwise it will be zero. If  $Y_{kl}$  is output of neuron found at  $(k, l)$  position, use a  $3 \times 3$  ( $k=3, l=3$ ) or  $5 \times 5$  ( $k=5, l=5$ ) square linking neuron set with “ $\times$ ” pattern or “ $+$ ” pattern linking as shown below in the matrix respectively. Eight-link neuron set is used as below.

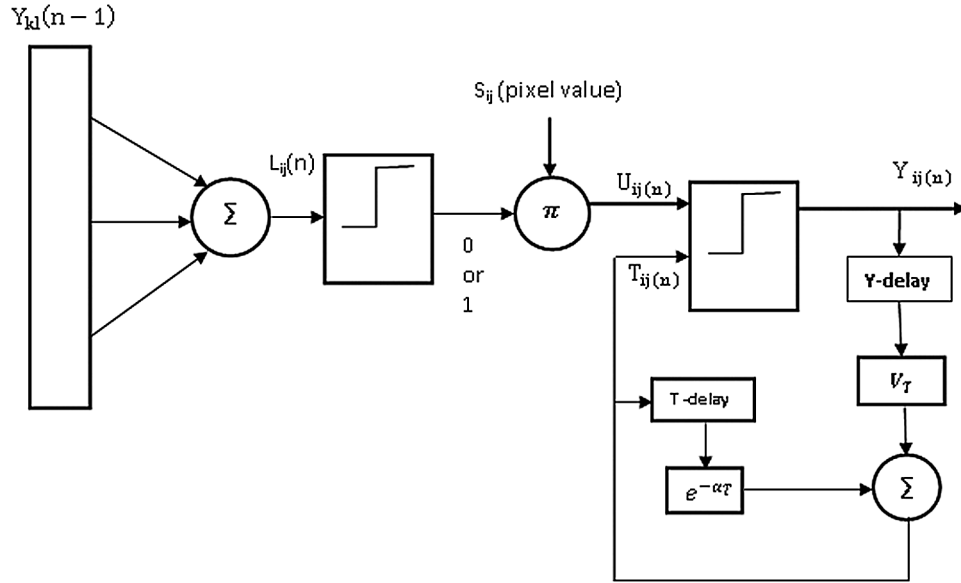


Figure 3: Model of LLP

```

0 1 0 1 0 1 1 1 1
1 1 1 0 1 0 1 1 1
0 1 0 1 0 1 1 1 1

```

(a) '+' Linking (b) 'X' Linking (c) 8-linking

$$U_{ij}[n] = F_{ij}[n](1 + \beta L_{ij}[n]) \quad (2)$$

$$T_{ij}[n] = T_{ij}[n-1]e^{-\alpha\tau} + V_T Y_{ij}[n-1] \quad (3)$$

The values of  $U_{ij}[n]$  and  $T_{ij}[n]$  is to decide output of neuron at different iterations. The first pulse cycle of neuron has 'n' number of iterations. This cycle starts with value of  $T_{ij}[n] = V_T$ , when next pulse cycle started at  $(n+1)^{th}$  iteration, gets residual value  $V_T e^{-n\alpha\tau}$  from previous cycle. If  $\alpha_T$  or 'n' is chosen high value, residual value becomes zero and next pulse cycle will also have 'n' number of iterations otherwise number of iterations in next pulse cycle will be more. But, if 'n' is to be high  $\alpha_T$  should be small to make threshold value slowly decaying

$$Y_{ij}[n] = \begin{cases} 1, & \text{if } U_{ij}[n] > T_{ij}[n] \\ 0, & \text{else} \end{cases} \quad (4)$$

The output  $Y_{ij}$  of a set of neurons corresponding to an image of size  $M \times N$  is a time series matrix of order  $M \times N$ . When whole image is provided as input to LLP, time series produced is called global time signature or global image icon. If only a portion of whole image is provided as input, LLP(UL-PCNN) produces signature of corresponding region and it is called as local time signature or local image icon. Dividing an image into smaller blocks and finding local signatures is necessary in applications such as object detection, navigation and authentication. The local time signature will reflect local changes and used for image forgery detection or image authentication. The change in neuron's output from 0 to 1 and again to 0 produces an oscillation. The frequency of oscillation depends on values assigned to above parameters of PCNN. The amplitude of this oscillation in each iteration is sum of outputs of all neurons.

$$G(n) = Y_{ij}(n)$$

Time series  $G(n)$  is rotation, translation, scaling and distortion in variance. The length of feature vector is defined as total number of the LLP iterative steps after analyzing the various methods for feature matching [19]. The LLP transforms an image into a series of binary images. The binary image sequence contains lots of information about shape, edge and texture features of original image. Similar images should show same features i.e., should produce same frequency and amplitude. Still, a duplicated image will have small group of pixels with intensities changed due to changed color or changed illumination levels or changed size of objects in image. Therefore, small difference exhibited by the forged pixels in their features. The  $G(n)$  depends on number of pixels in image i.e., if size of the image is changed,  $G(n)$  will change, so it is proposed to normalize  $G(n)$  against size of image.

LLP is used for identification of forensic through binary images obtained at various values of  $G(n)$ . The recognition precision is to estimate through percentage against number of iterations. Furthermore, Duplication in image is identified with 100% precision, whereas more than 45% change seen in texture of image and never classified. In the proposed LLP method extract features of images and does not require any training. Furthermore, intensity of pixel  $(i, j)$  i.e.,  $S_{ij}$  is normalized against maximum intensity level in image, so that any pixel can have a value from 0 to 1. The input  $S_{ij}$  to a particular neuron  $(i, j)$  is constant throughout all iterations. The LLP-Algorithms steps are as below. The LLP algorithm identification is explained for each manipulated images in further sections.

---

### LLP-Algorithm

---

*Step 1: Read the input image*

*Step 2: Initialize the values of  $S_{ij}$ , decay term for feeding, linking and threshold  $\alpha T$*

*Step 3: Assign the initial magnitude scaling term as  $Vt$  for 5*

*Step 4: Specify the linking Strength (Beta) as 0.1*

*Step 5: Mention the number of iterations*

*Step 6: assign the initial values for  $F, L, Y, U, T$*

*Step 7: Adjust the values to lie within 0 and 1*

*Step 8: update the feeding and linking input for each iteration*

$$F_{ij}[n] = S_{ij}$$

$$L_{ij}[n] = \sum_{kl} y_{kl}(n-1) \text{ getting } L_{ij} \text{ as } 0 \text{ or } 1$$

*Step 9: Compute the following*

$$U_{ij} = S_{ij} * (1 + (0.1 * L_{ij}))$$

$$TCI_{ij} = TPI_{ij} * \exp(-\alpha T) + (5 * YPI_{ij}),$$

**if  $YPI_{ij} > 0$  then  $L_{ij} = 1$  else 0**

**Step 10: if  $U_{ij} > TCI_{ij}$  then  $Y_{ij} = 1$  else 0 update the threshold and update the activity**

*Step 11: Feature vectors of 100 iterations are received as output for each image*

---










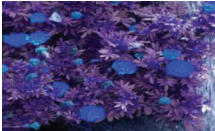
Irrespective of the type of image, if a neuron is excited by a pixel with an intensity value of  $S_{ij} = 1$ . The number of iterations required between two consecutive firing is known as pulse cycle. Since, the value for threshold-decay parameter ( $\alpha_T$ ) is constant, if a neuron is excited by a pixel with an intensity value of  $S_{ij} = 0$ ,

then more number of iterations will be taken by that neuron to complete one pulse cycle. This indicates that a high intensity pixel will have more influence on output or feature vector of LLP, than a low intensity pixel. When number of iterations is manually fixed to less value, then neurons with low level of activation will not contribute anything in output or feature produced. At the same time if more number of iterations are used, neurons with high activation will produce ‘1’ frequently and may mislead final conclusion. Therefore, it is necessary to derive expressions to know minimum number of iterations required within all neurons and will be fired.

### 3.1 Single Manipulated Copy-Move Forgery Identification

The images from benchmark data sets are taken for color and rotation changes, these changes in image done through free online image editing tools like adobe spark, pine tools for giving additional inputs to model for feature extraction. These changes give intentionally more number of copied images as innovative idea to predict originality of image. The below given [Tab. 2](#) shows a set of changes performed from original image in data set.

**Table 2:** Single and multiple manipulated images given as input to LLP

SINGLE & MULTIPLE MANIPULATIONS	SOFTWARE/HARDWARE (TOOLS AND METHODS)	ORIGINAL IMAGE	MANIPULATED IMAGE
<b>BLURRING</b>	Contrast changes Photo shop Adobe illustrator Select original image and blur with user interface menu		
<b>NOISE ADDITION</b>	Pixelitor HSV Noise Noise added to the original Image as single manipulation		
<b>NOISE ADDITION AND CONTRAST ADJUSTMENT</b>	Pine tools Resize APP Multiple Manipulation on the same image		
<b>BLURRING AND SEPIA EFFECT</b>	Easy Paint studio Canva for Enterprise Multiple Manipulation		
<b>HU ADJUSTMENT WITH BRIGHTNESS CHANGE</b>	Ease paint water remover ACD see Adobe Spark Multiple manipulation		



### 3.2 Multiple Manipulated Copy-Move Forgery

It is verified experimentally that more than one change done on same image are also taken as input for feature extraction to proposed LLP algorithm. Multiple manipulations perform through original image and performing the blurring, noise addition with contrast adjustment on the same image after doing multiple changes in single image, noise is added image.

Again contrast is increased for making to look image with minor variations that cannot be detected through eye. For the above single and multiple manipulations, LLP features are obtained as in [Tab. 3](#)

**Table 3:** Feature vectors received as output form LLP Algorithm

Image Name	Iteration1	Iteration2	Iteration 3	Iteration 4	Iteration ...	Iteration100
001_O.png	0.8158493	0.181953	0.079769	0.398647	...	0.23455
001_F_BC1.png	0.81788635	0.179626	0.109638	0.435841	...	0.23925
001_F_BC2.png	0.82730484	0.170502	0.162735	0.525455	...	0.396111
001_F_BC3.png	0.85306931	0.200623	0.367996	0.487068	...	0.366302
001_F_CA1.png	0.81394577	0.183971	0.057476	0.354473	...	0.292744
001_F_CA2.png	0.80683136	0.190975	0.051125	0.285225	...	0.210354
.	...	...	...	...	...	...
.						
.						
.					...	
200_O_NA3.png	0.7517395	0.246788	0.106705	0.225769	...	0.13557

[Tab. 3](#) shows the feature vectors extracted using LLP as time series vector from iteration 1 to 100 for all 10,000 images from CoMoFoD, GRIP and MICC-220.

### 3.3 SVM and Kernel Trick

Support vector is a supervised Machine learning classification algorithm. Kernel trick is applied for finding the optimization boundary by converting the data points for high dimensional data of feature vectors. There are seven types of kernel of which four types are implemented such as linear kernel, polynomial kernel, sigmoid kernel and radial basis kernel. Each kernel features are explained in the algorithm given below.

#### Kernel Trick Classifier (KTC) Algorithm

1. Load the CSV file containing  $G$  feature vectors for 100 iterations of images
2. Mapping input space into infinite dimensionless space. Plot each data item as a point in  $n$ -dimensional space where  $n$  represents 100 features
3. Initialize Kernel parameters, gamma value and cost function with kernel trick converting from low dimension space to high dimension space.
4. **Linear Kernel**  $k(x, y) = x^T y + c$   
**Polynomial kernel**  $k(x, y) = (\alpha x^T y + c)^d$

(continued)

---

**Algorithm: (continued)**


---

*Sigmoid kernel or Hyperbolic Tangent Kernel*  $k(x, y) = \tanh(ax|Ty + c)$

*Radial basis Function Gaussian Kernel*  $k(x, y) = \exp$

$$z(y, y|i) = \exp\left(-\text{gamma} * \sum(x - x_i)\right)$$

*gamma ranges from 0 to 1.*

---

#### 4 Experimental Results and Discussion

In this section, feature vectors obtain from LLP is applied for kernel trick for forgery image identification. In kernel tricks, initially find optimum number of LLP-Features and optimum value of threshold (matching accuracy) to be used. From selected LLP features, a heuristic rule for selecting optimum number of features and matching accuracy values are applied. This heuristic rule on forgery image datasets such as CoMoFoD [20], GRIP, MICC-220 [21] is to justify performance of proposed LLP method and parameter selection compared with other existing approaches. All experiments are conducted on Intel (R) Core (TM) i5 2410 M, 2.3 GHz with turbo boost up to 2.9 GHz and 4 GB RAM. Our method was implemented using 64-bit MATLAB V2020 run under Windows 7 Home premium 64-bit operating system. Further, G(n) features from LLP as times series are given input to kernel trick classifier algorithm and classify with parameters. The kernel trick classifier algorithms is as shown below. By using the kernel trick classifier accuracy was improved and produces better accuracy.

Tab. 4 represents GRIP image dataset with single manipulation dataset with poly kernel reaches accuracy of 70% which is less, when compared to multiple manipulated image with an improved accuracy of 85%. From simulation results, it is conveyed that sigmoid and RBF kernel are producing better accuracy in classification between poly and linear. Linear kernel is better for multiple manipulated images (GRIP dataset).

**Table 4:** Performance measures for GRIP as output from KTC

Kernel optimizer	Accuracy	Precision	Recall	F1-Score	Support	Type of Forgery detected
Poly	0.7083	0.6666	0.7826	0.71	48	Single manipulated
Linear	0.8541	0.8636	0.8260	0.85	48	Multiple manipulated images with dataset
Sigmoid	0.79166	0.8095	0.7391	0.79	48	Single manipulated
Rbf	0.8125	0.8500	0.7391	0.81	48	Multiple manipulated images with dataset

---

Tab. 5 gives the results received from MICC-220 dataset with single and multiple manipulated images with reference to [22]. It is seen from result RBF kernel provides 90% accuracy for MICC-220, for rotated, blurred and single manipulated images have lesser classifier accuracy with 83%, 86% and 65% for poly, linear and sigmoid kernels respectively. So it is proved that RBF kernel is the best classifier for MICC-220.

**Table 5:** Performance measures for MICC-220 as output from LTC

Kernel optimizer	Accuracy	Precision	Recall	F1-Score	Support	Type of Forgery detected
Poly	0.8333	0.7428	0.9285	0.91	66	Single manipulated
Linear	0.8636	0.9523	0.7142	0.86	66	Multiple manipulated images with dataset
Sigmoid	0.6515	0.5959	0.5714	0.65	66	Single manipulated
Rbf	0.9090	0.9230	0.8571	0.91	66	Multiple manipulated images with dataset

In [Tab. 6](#) results shows for CoMoFoD dataset with multiple manipulated image and produces greater accuracy in RBF kernel of 85.18% when compared to single manipulated image with dataset of 74%, 77%, 77% in poly, linear and sigmoid kernels respectively.

**Table 6:** Performance measures for CoMoFoD as output from SVM

Kernel optimizer	Accuracy	Precision	Recall	F1-Score	Support	Type of Forgery detected
Poly	0.7407	1.0	0.4166	0.71	27	Single manipulated
Linear	0.7777	0.875	0.5833	0.77	27	Multiple manipulated images with dataset
Sigmoid	0.7777	0.8	0.6666	0.71	27	Single manipulated
Rbf	0.8518	0.8333	0.8333	0.85	27	Multiple manipulated images with dataset

In [Fig. 4](#), MICC-220 with multiple Manipulated images provides accuracy of 95.39% with RBF kernel and CoMoFoD with ROC curve is 89.2%, Sigmoid and RBF is about 87.71%. For GRIP dataset Linear kernel is 85.91%. The range of ROC value varies from 73.39% to 85.91%.

[Tab. 7](#) shows precision values of proposed LLP with other algorithms. Comparatively, proposed LLP accuracy is about 87.5%, 92.3%, 86.36% and 97.8%. The proposed LLP algorithm shows better performance with different datasets including coverage [\[23\]](#).

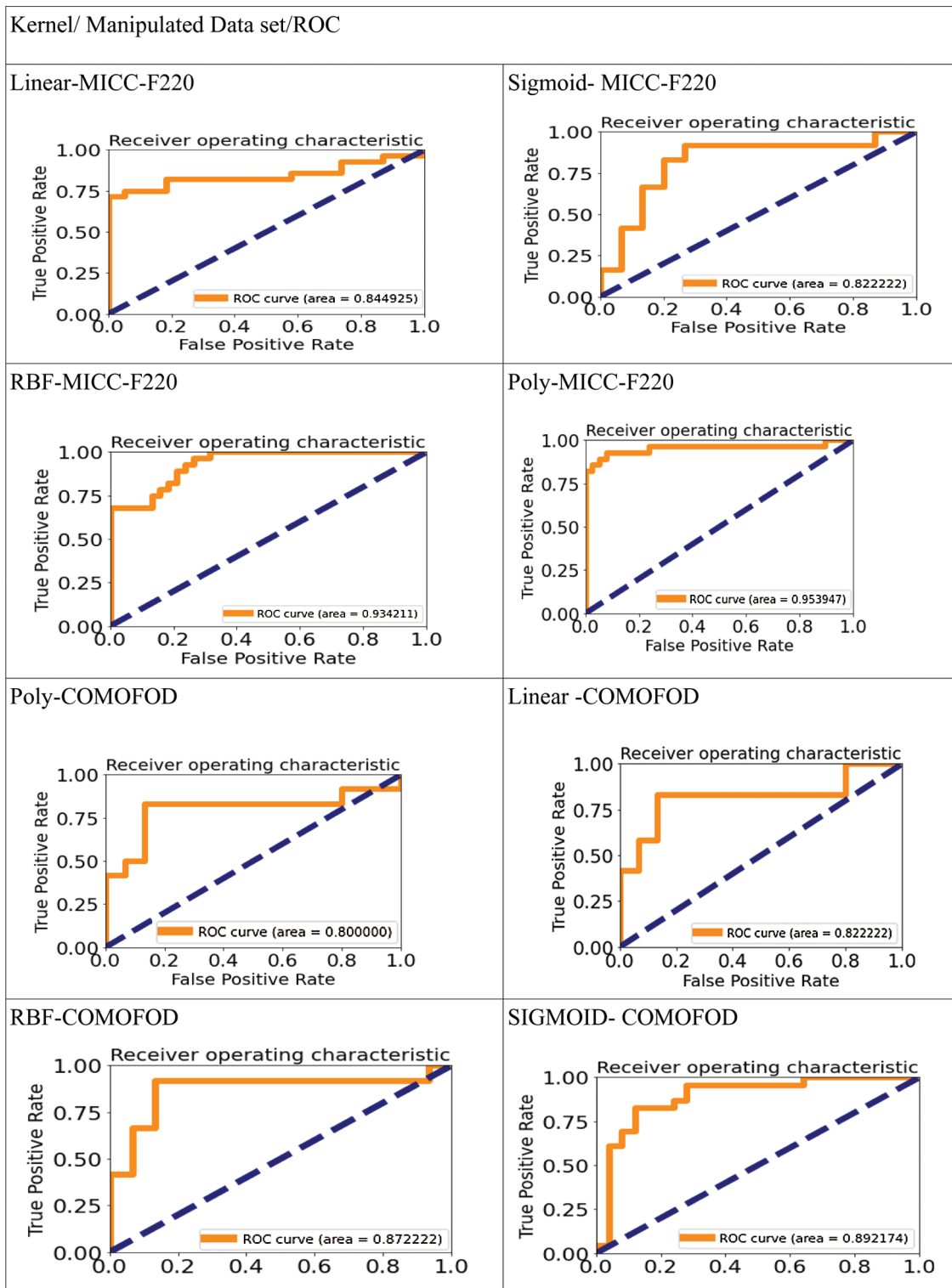


Figure 4: Continued

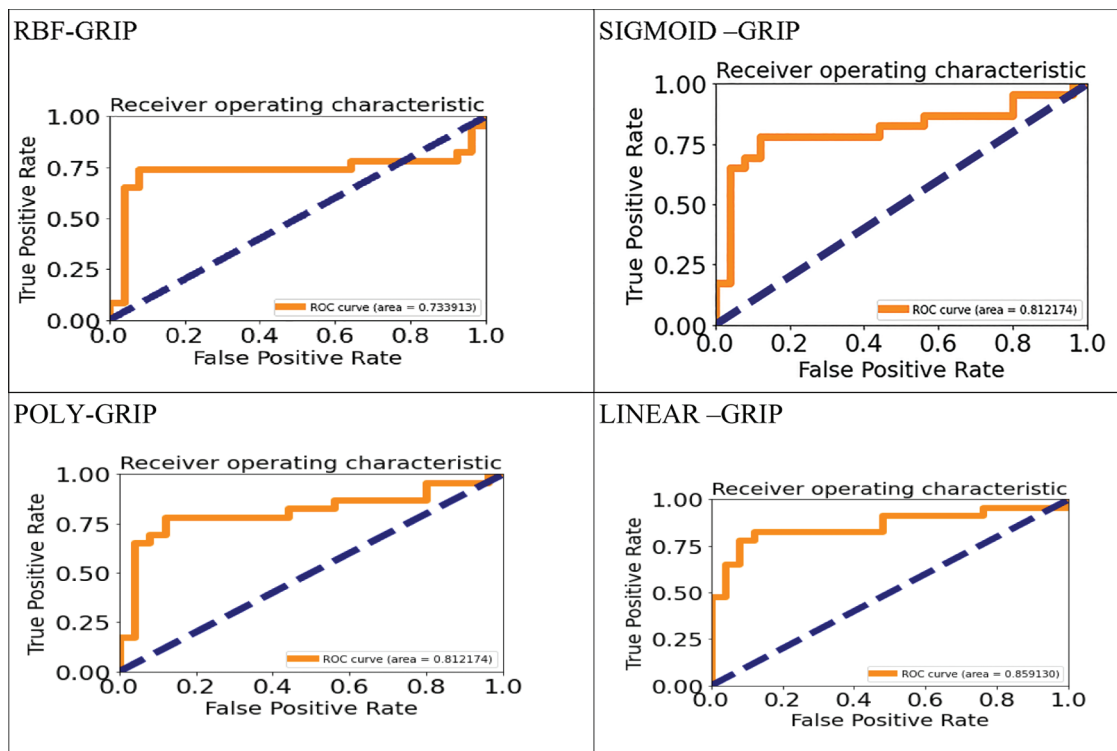


Figure 4: ROC curve for various kernel and dataset

Table 7: Comparison of performance metric for proposed LLP with other authors

Image transformation data set	SIFT	SURF	Dense field	SGO	PCNN	Proposed system UL-PCNN+SVM
CoMoFoD	77	51.5	72	70.1	84.1	87.5
MICC	76	85.2	62	75	87.5	92.3
GRIP	71	82	52	-	84.2	86.36
Coverage	50.5	58.6	71.8	73	94.4	97.8

### 5 Conclusion

The proposed LLP algorithm obtained through pulsed neural network and combines with kernel tricks for detection of manipulations in forged image. LLP algorithm linking pixels detects image multiple manipulation due to correspondence between pixel and neuron. Neuron is connected with neighboring field of neuron for detecting forged images with multiple manipulations in the image along with copy-move, through kernel trick classifier (KTC). The proposed system with LLP algorithm is implemented by selecting optimum feature parameter from LLP. The LLP applied in forgery standard dataset such as GRIP, COVERAGE, MICC-220 particularly for copy move forgery and same dataset images apply with single and multiple manipulated through standard software tool, checked for performance of LLP and

KTC. The proposed LLP method provide better performance with suitable precision and recall values for manipulated images along with images in standard and created data set. The proposed LLP algorithm shows better performance in terms of accuracy about 87.5%, 92.3%, 86.36% and 97.8% in various data sets. The images which that are duplicated with simple changes, multiple changes on the same image, changes made in different images and pasted on original image have correlation value for matching percentage range about 0.8, 0.4, 0.3 respectively. The results are more useful for forgery and forgery manipulated image detections. Furthermore, hardware based forgery image acquired images need to be check with proposed LLP and KTC algorithms.

**Acknowledgement:** The authors are sincerely grateful to the anonymous referees and the editor for their timely effort in providing constructive and value comments and suggestions that have led to a substantial improvement in the paper.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the Present study.

## References

- [1] G. K. Birajdar and H. Vija Mankar, "Digital image forgery detection using passive techniques: A survey," *Digital Investigation*, vol. 10, no. 3, pp. 226–245, 2013.
- [2] G. Priyanka, G. Singh, G. KulbirSingh and K. Singh, "An improved block based copy-move forgery detection technique," *Multimed Tools Applications*, vol. 79, pp. 3011–13035, 2020.
- [3] S. Dhivya, J. Sangeetha and B. Sudhakar, "Copy-move forgery detection using SURF feature extraction and SVM supervised learning technique," *Soft Computing*, vol. 24, pp. 14429–14440, 2020.
- [4] M. M. Islam, G. Karmakar, J. Kamruzzaman and M. Murshed, "A robust forgery detection method for copy-move and splicing attacks in images," *Electronics 2020 MDPI*, vol. 2, pp. 1–22, 2020.
- [5] N. Jindal and A. Thakur, "Hybrid deep learning and machine learning approach for passive image forensic," *IET Image Processing*, vol. 14, no. 10, pp. 1952–1959, 2020.
- [6] K. Kumar and R. K. Mishra, "A heuristic SVM based pedestrian detection approach employing shape and texture descriptors," *Multimedia Tools Applications*, vol. 79, pp. 21389–21408, 2020.
- [7] K. K. Thyagarajan and G. Kalaiarasi, "Pulse coupled neural network based near duplicate detection of images (PCNN–NDD)," *Advances in Electrical and Computer Sciences*, vol. 18, no. 3, pp. 87–96, 2018.
- [8] G. Nirmala and K. K. Thyagarajan, "A modern approach for image forgery detection using brich clustering based on normalized mean and standard deviation," in *Proc. of ICCS-Int. Conf. on Communication and Signal Processing*, Saveetha Engineering College, Chennai, 2019.
- [9] A. Roy, R. Dexit, R. Naskar and R. S. Chakraborty, "Copy move forgery detection with similar but genuine objects," *Digital Image Forensics*, vol. 775, pp. 65–77, 2020.
- [10] H. Xie, K. Gao, Y. Zhang, S. Tang, J. Li *et al.*, "Efficient feature detection and effective post-verification for large scale near-duplicate image search," *IEEE Transactions on Multimedia*, vol. 13, no. 6, pp. 1319–1332, 2011.
- [11] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1841–1854, 2011.
- [12] V. Christlein, C. Riess, J. Jordan, C. Riess and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, 2012.
- [13] C. S. Prakash, A. Kumar, S. Maheshkumar and V. Maheshkumar, "An integrated method of copy-move and splicing for image forgery detection," *Multimedia Tools and Applications*, vol. 77, pp. 26939–26963, 2018.

- [14] R. Eckhorn, H. J. Reitboek, M. Arndt and P. Dicke, "Feature linking via synchronization among distributed assemblies: Simulations of results from cat visual cortex," *Neural Computing*, vol. 2, pp. 293–307, 1990.
- [15] B. Soni, P. K. Das and D. M. Thounaojam, "CMFD: A detailed review of block based and key feature based techniques in image copy-move forgery detection," *IET Image Processing*, vol. 12, no. 2, pp. 167–178, 2020.
- [16] W. L. Zhao and C. W. Ngo, "Scale-rotation invariant pattern entropy for key point-based near-duplicate detection," *IEEE Transactions on Image Processing*, vol. 18, no. 2, pp. 412–423, 2009.
- [17] J. Dass, V. Sarin and R. N. Mahapatra, "Fast and communication-efficient algorithm for distributed support vector machine training," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 5, pp. 1065–1076, 2018.
- [18] Y. Chen, S. K. Park, Y. Ma and R. Ala, "A new automatic parameter setting method of a simplified PCNN for image segmentation," *IEEE Transactions on Neural Networks*, vol. 22, no. 6, pp. 880–892, 2011.
- [19] M. A. Qureshi and M. Deriche, "A bibliography of pixel-based blind image forgery detection techniques," *Signal Processing: Image Communication*, vol. 39, part A, pp. 46–74, 2015.
- [20] Copy Move Forgery Detection (CoMoFoD) <http://www.vcl.fer.hr/comofod>, Queen Mary University of London, <http://qmro.qmul.ac.uk/jspui/handle/123456789/8150>.
- [21] J. L. Johonson, "Pulse-coupled neural nets: Translation, rotation, scale, distortion and intensity signal invariance for images," *Appl. Opt.*, vol. 33, no. 26, pp. 6239–6253, 1994.
- [22] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to ROC, informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.
- [23] B. Wen, Y. Zhu, R. Subramanian, T. T. Ng, X. Shen *et al.*, "COVERAGE-a novel database for copy-move forgery detection," in *IEEE Int. Conf. on Image Processing*, Phoenix, AZ, USA, pp. 161–165, 2016.