Journal of Scientific & Industrial Research
Vol. 81, September 2022, pp. 963-971
DOI: 10.56042/jsir.v81i09.55455

A Novel Approach to Detect Copy Move Forgery using Deep Learning

Mamta*, Anuradha Pillai & Deepika Punj

Computer Engineering Department, J C Bose University of Science and Technology, YMCA, Faridabad 121 006, Haryana, India

Received 28 September 2021; revised 25 August 2022; accepted 25 August 2022

With the development of readily available image editing tools, manipulating an image has become a universal issue. To check the authenticity, it is necessary to identify how various images might be forged and the way they might be detected using various forgery detection approaches. The importance of detecting copy-move forgery is that it identifies the integrity of an image, which helps in fraud detection at various places such as courtrooms, news reports. This article presents an appropriate technique to detect Copy-Move forgery in which to some extent an image is copied and pasted onto an equivalent image to hide some object or to make duplication. The input image is segmented using the real-time superpixel segmentation algorithm DBSCAN (Density based spatial clustering of application with noise). Due to the high accuracy rate of the VGGNet 16 architecture, it is utilized for feature extraction of segmented images, which will also enhance the efficiency of the overall technique while matching the extracted patches using adaptive patch matching algorithm. The experimental results reveal that the proposed deep learning-based architecture is more accurate in identifying the tempered area even when the images are noisy and can save computational time as compared to existing architectures. For future research, the technique can be enhanced to work on other forgery detection techniques such as image splicing and multi-cloned images.

Keywords: Adaptive patch matching, CNN, Copy move forgery detection, DBSCAN, VGGNet

Introduction

With the arrival of various powerful devices such as Digital Camera and various image editing tools such as Adobe Photoshop and GNU Image Manipulation Program (GIMP), manipulating an image has become much easier. An image is not always manipulated with unfair intentions; image manipulation may be needed sometimes to enhance the quality or for some fair purposes. However, sometimes forgers do it with fraudulent intentions and may pose several issues in social media, courtroom, medical diagnosis, and insurance claims. Consequently, in the last two decades, there has been a substantial increase in the techniques which can enhance the credibility of an image by identifying its originality and this study of images is known as Digital Image Forensic.[1] Image forensics is used for authenticating images and provides credibility to the images about their origin. There are two approaches to detect the forgery in image[2], Active Approach (non-blind) and Passive Approach (blind).

Active approaches are less researched because they need prior information about the image. To check authenticity in an image, the features are extracted in the form of a signature and then the extracted signature will be compared with the existing information of image to know the credibility of the image. These types of approaches that perform pre-processing on the image belong to the class of non-blind approaches. Main examples of active approaches are Digital watermarking[3], digital signature[4], and steganography.

However, in Passive approach forgery is detected based on features extracted from the image, the type of tempering done on an image, or from the device through which the image has been captured. When the detection process solely depends upon the post-processing of an image, then it belongs to the class of blind approach of forgery detection. Passive detection is further divided into two categories: independent and dependent forgery. In independent forgery, manipulations are done on the same image whereas, in the case of dependent forgery, manipulations are done by partially copying and pasting an image onto the other image of the same type. Some examples of independent forgery are image retouching, scaling, resizing and compression, etc. Examples of dependent forgery include image splicing, copy-move image forgery are shown in (Fig. 1). This paper emphasizes on Copy-Move image Forgery Detection (CMFD), in

*Author for Correspondence
E-Mail: mamtamahiya@gmail.com

which some region of the image is copied and pasted on the same image at a different location with malicious intent.

In image retouching, some properties of the image are adjusted in such a way that the modification is difficult to detect. Image resampling changes the resolution of the image so as to increase the dimensions of the image, specifically used for banners and billboards, or to minimize the dimensions of the image used as an attachment on emails and websites. Image splicing is another forgery in which two or more images are merged into one image by the forger so that it looks real. A bar graph depicting the number of research done on various passive image forgeries based on Google scholar articles is shown in Fig. 2. The bar graph displays various publications titled different passive forgery techniques in the last two decades. As per the observation, it can be seen that CMFD is the most researched topic in the last decade and considerable attention is given to image splicing whereas image retouching and resampling have been less researched because it makes the least impact on the picture and hence does not lead to serious forgery.

A clear case of copy-move image forgery is shown in Fig. 3. Here one of the tigers is copied and pasted in the same picture and, as it is a crowd it cannot be detected that some part of the picture is copy-pasted because the essential features of the image such as color, contrast, and noise properties do remain the same as the pasted part is copied from the image itself.

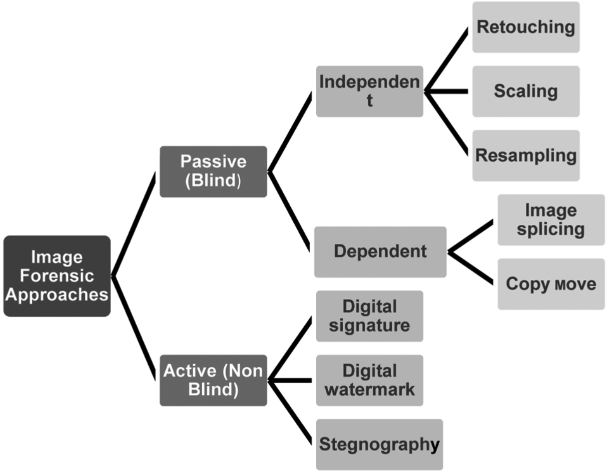

Fig. 1 — Type of Image Forgery Detection approach

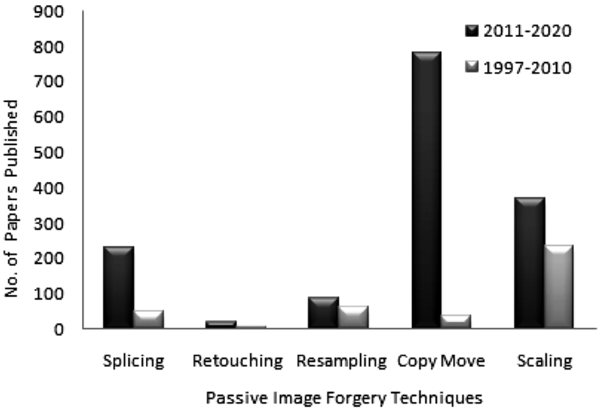

Fig. 2 — Bar graph depicting the number of research done on various passive image forgeries based on Google scholar articles in last two decades

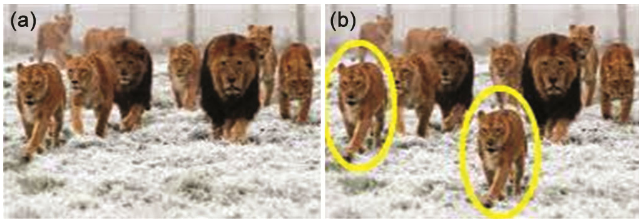

Fig. 3 — Depicting an illustration of copy move image forgery

Literature Review

In the previous decade, much research has been done on image forensic detection and Copy-Move forgery is the most discussed among all the techniques. CMFD technique needs to be developed based on various factors such as the accuracy and reliability of the results achieved so far. The method used should be able to achieve better speed and computational complexity than the existing techniques.

Recently, many reviews, surveys, and techniques have been proposed on CMFD. Teerakanok & Uehara.[5] provided up-to-date information associated with the present advancement in CMFD. It explained a new CMFD process pipeline. Warif *et al*.[6] presented the detailed process of CMFD, various feature extraction techniques, matching methods, and publically available datasets. Cristline *et al*.[7] created a dataset supporting copy-move forged images and a software framework to detect the manipulation of the image. They experimented with the dataset using the block-based and key point feature extraction techniques and reliably identified the copied region within the image. Badal *et al*.[8] gave an in-depth review of all the CMFD techniques employed by researchers over a decade and also addressed variation in database, issues, and challenges. Zhang *et al*.[9] reviewed two models of the CMFD technique and did the performance evaluation of the models.

Meena & Tyagi[10] presented a detailed survey of all the CMFD techniques and the information about research conducted in this field thus far.

There is tremendous research based on various deep learning techniques that have been done on CMFD. Bilal *et al.*[11] have proposed a CMF detection and localization technique using the fusion of SURF (speeded up robust features) and BRISK (binary robust invariant scalable keypoints) descriptor and grouped the matched features using DBSCAN clustering. The SURF features can handle forgery due to rotation and noise and the BRISK detects the scale-invariant forged region and poorly localized key points in a forged image. The workflow surpasses the prevailing techniques used for localization and detection of forged region. Bi. *et al.*[12] proposed multiscale feature extraction and adaptive patch matching algorithm. They applied SIFT (scale-invariant feature transform) algorithm for the extraction of features from all the patches from a segmented image and proposed an adaptive patch matching algorithm which later helps in matching the suspicious forged region in the image. The algorithm performs well in many promising conditions, such as image rotation, scaling and noise addition.

Aggarwal & Verrna[13] have provided an efficient CMFD technique. They have used the SLIC segmentation technique for image preprocessing and CNN-based VGGNet architecture for the features extraction from the segmented patches. Using the adaptive patch matching algorithm, they compared each block of the image and successfully detected the tempered region in the image. This article has achieved good accuracy in comparison to other existing techniques.

Presently, there are various research articles which make use of different techniques to identify the Copy-Move Forgery, but the framework used in this paper works well even when the image is noisy and is better in terms of time complexity. The novelty of the approach used in this paper is that it uses the DBSCAN algorithm, which is a technique for image segmentation, but the algorithm has been modified in this research and utilized to segment as well as refine the image. It can refine the superpixels obtained from the input image and hence provide more accurate results. This algorithm along with VGGNET 16 architecture has not been utilized earlier for Copy-Move forgery detection. Also, the VGGNET 16 architecture used to extract the features uses less memory and less time as compared to other non-CNN-based techniques. The reason behind focusing on Copy-Move forgery detection is that the forgery done using this approach may cause serious fraud. However, other techniques such as retouching and scaling can only change the properties of an image, not the meaning of the image.

Experimental Details

Image forgery can be achieved by using several image editing softwares. Our method focuses on the detection of the forged area. The flow of work proposed can be seen in Fig. 4

The idea is to divide the image into segmented patches using the modified DBSCAN segmentation technique and then, refine the pixels in order to achieve high accuracy. The segmented patches obtained after superpixel refinement are given as input to VGGNet. Then, the deep learning based VGGNet -16 architecture is used for the extraction of multiscale features from the segmented patches. Then depth of pixels is reconstructed which help in lessening the variance among the original and forged patches. The Adaptive patch matching algorithm compares the segmented patch with another patch. After this process is carried out, most of the tempered region is detected and displayed in the output.

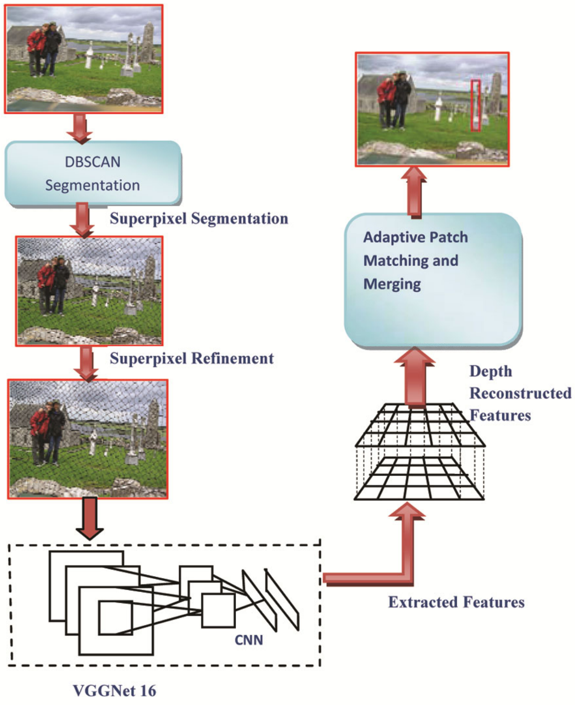

Fig. 4 — The proposed architecture to identify tempered area

Image Clustering using DBSCAN

DBSCAN is an image clustering technique that produces outstanding results for spatial clustering applications.[14, 15] There are various clustering algorithm such as SLIC[16], LSC[17], ERS[18], SEEDS[19], Mean Shift[20], Turbo-pixel.[21] Every algorithm has its advantage and disadvantage; however, it is really difficult to attain the properties such as better boundary adherence, regular shapes, dense constraints, and low computational complexity in a single algorithm. Turbo-pixel clustering cannot achieve better boundary adherence, LSC, ERS, and SEEDS clustering are alike in terms of boundary adherence but LSC maintains more regularity and perceptual satisfaction.[22] LSC performs better than SLIC in terms of boundary adherence but the computational time is more with LSC when compared to SLIC.[22]

This literature uses DBSCAN clustering which is proposed by Martin *et al*.[23] Given a dataset of points, it groups spatially closed keypoints and finds the cloned regions. It separates the lower density regions from higher density regions. The areas having noise have a lower density than the other areas of the image.[23] The reason for choosing DBSCAN over the other algorithms is that it does not have a necessity to mention the number of clusters to be formed and this algorithm provides better boundary adherence and is good to detect and segment poor quality images. The DBSCAN algorithm divides an image into small compact regions of homogeneous appearance and this procedure is known as clustering in which each cluster has superpixels (formed by a perceptual grouping of the pixels) of unique features such as color and shape.

The DBSCAN is comprised of two stages - clustering stage and merging stage.[22] In the clustering stage, the pixels of the image are divided into two sets; candidate set and labeled set. The topmost pixel in the left corner of given input image is taken as a label for the first seed and put into the labeled set. Then all the corresponding seeds are added into the labeled set one by one from left to right and from top to bottom. Then find four neighboring pixels which are unlabeled with respect to the labeled set and then calculate the combined distance between each unlabeled pixel concerning its center pixel. If the distance obtained is less than the threshold value then add the seed into the candidate set. In the last step, update the labeled set by replacing it with the candidate set. Repeat this process to meet the termination condition.

To meet the termination condition there are two aspects: One, the number of pixels in the cluster should be more than the threshold value S/N where, S depicts the size of a given image and N is the maximum number of superpixels that can be there in a cluster (value given by the user). Two, the labeled set should become an empty set i.e the pixels conjoining the previously labeled set are at the boundary region.

The distance function is calculated using the seed distance item and the neighbor distance item. The seed distance item assures that each superpixel accommodates the same type of pixel and the neighboring distance item takes care of the weak boundary and the flat region around the pixel.

In the clustering stage, initial superpixels L(p) are obtained and the merging stage merges all those initial superpixels which create small sized fragments which are generated at some of the edges of the objects. To eliminate the smaller fragment, add it to its neighboring fragments to form a larger and clearer superpixel.

In the merging stage, find out the average number of pixels in total superpixels, and then if the count of pixels obtained in any of the superpixel Sp is less than the average value, then it will be considered as a smaller cluster that can create a cluttering effect in the image. To get the refined superpixels, merge those superpixels with the neighbor having the shortest distance and then the final refined superpixel with common shapes can be acquired. The formulation of the stages can be seen in the algorithm below:

Algorithm: *Modified DBSCAN clustering to get segmented and refined superpixels*

Input: Host image

Output: Segmented and Refined Superpixels Sp.

1. Load the host image and set pixels label as 0 for each image in I. Let initially
   Threshold ← φ, Label set L← empty, Candidate set C← empty, Seed Z ← empty, Superpixel Sp.
2. Start from left to right and top to bottom. Pick the top left pixel as seed Z and add it label set L. L← Z
3. Find each neighboring pixel x around label set pixel y and insert them into unlabeled set U and then calculate clustering distance d(x,y) between unlabeled pixel from its center pixel and seed item Z.
4. If No. of pixel in Sp is < φ. where, φ= S/N where, S← Image Size, N← #of Superpixel(user input)
5. If d(x,y) > φ OR L=empty;

```
    Threshold $\varphi$ = S/N where, S← Image Size, N← #of Superpixel
    Then set C = L terminate;
    Else set C← x;
6.  Repeat step 2–4;
    To get refined Superpixel
7.  Find average no of pixels(Ap) obtained in each cluster. For each superpixel Sp
    If( No of pixels in superpixel Sp < Ap )
    Then add those super pixel's seed items to merge set M.
8.  For each pixel l in superpixel Sp around pixel p in merge set. Compute the merge distance D'(l,p).
    $D'(l,p)=D_c(l,p)+D_s(l,p)$
    where, $D_c(l,p)=\sqrt{}(R_1–R_2)^2 +(G_1–G_2)^2 + (B_1–B_2)^2$
    $D_s(l,p)=\sqrt{}(x_1–x_2)^2 + (y_1–y_2)^2$
9.  Select the pixel which has minimum distance among all Pixels around p and merge the two superpixels.
10. Repeat the step 6–8 until the merge set is empty.
```

After applying the algorithm, the whole image is segmented into initial superpixels but the clustering stage generates very small fragments at some of the edges of the objects as shown in Fig. 5 a. The reason for the smaller fragments in the image is due to the use of distance-based pixel formation which is sensitive to local color features. Therefore, the merging stage is performed to eliminate the fragments to get refined superpixels. The result obtained after using DBSCAN algorithm is shown in Fig. 5 b.

Feature Extraction from Segmented Input Image

After the superpixel segmentation, VGGnet 16 architecture is used for feature extraction from the segmented image. VGG was named after the visual geometric group lab of Oxford University. It is a convolution neural network that was first presented by Simonyan & Zisserman[24] in their paper. The VGG 16 architecture can take a fixed size input image having dimensions 224 × 224. For feature extraction, VGG uses the combined features of convolution layer and max-pooling layer. The superpixels obtained after applying DBSCAN are binary masked with the input image and then the image is inputted to a stack of convolution layers where a filter of size 3 × 3 is used.

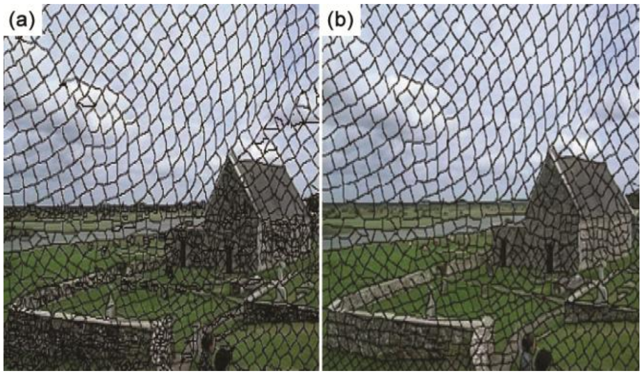

Fig. 5 — Result obtained after applying modified DBSCAN algorithm

As shown in Fig. 6 the structure of VGGNet 16 consists of 16 layers. Here 16, represents the number of layers having some parameter values. The extraction of multiscale features is done by convolution layer whereas max-pooling layer helps in reducing the size of image hence it extracts the dense features. The extracted features help in identifying the tempered region in the image. The reason for using this architecture is that it helps to extract the features with less memory usage and in less time as compared to other non-CNN based techniques.

Training and Configuration Structure of VGGNet 16

To train the network, first input the image into the first layer of VGG and to minimize the loss function the training is carried out using stochastic gradient descent with a momentum of 0.9. The input image should be of size 224. The batch size of 256 is taken with a learning rate of $10^{–3}$ then it is decreased by a factor of 10 in each iteration and stops learning after 74 epochs when the validation set accuracy stopped improving. As shown in Fig. 7, a stride of size 1 pixel and spatial padding of 1 pixel are used for every 3 × 3 convolution layer matrix. Five max-pooling layers perform Max pooling over a 2 × 2 pixel window and hence reduces the size of the image. Three Fully connected layers are having 4096 channels each. Last layer is the soft-max layer. All hidden layers are provided with rectification non-linearity. [25] To get the minor variations and detailed information between the original and forged image, the multiscale features of the image need to be extracted because the single-

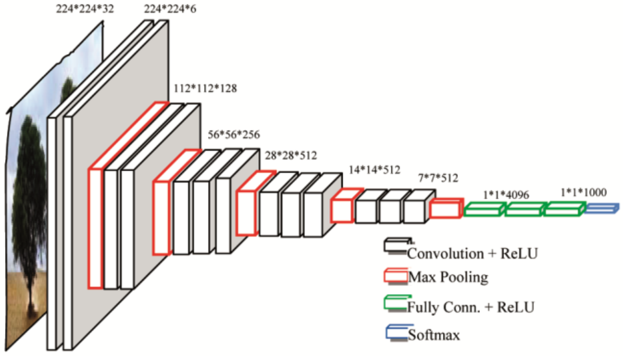

Fig. 6 — VGG net16 architecture

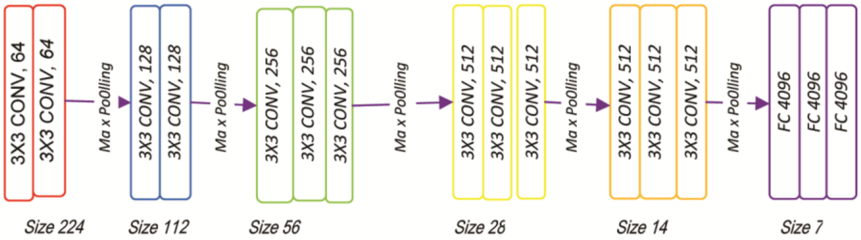

Fig. 7 — The configuration of VGGNet 16

scale features were not able to extract the minute features of the images.

The multiscale features obtained after the feature extraction step will help in the depth reconstruction of the image and will further increase the accuracy of the overall proposed work.

Feature Matching using Adaptive Patch Matching Algorithm:

Before matching the keypoint obtained after feature extraction, first the dense depth of the patches is reconstructed and then, to identify the suspicious region, the reconstructed patches are matched with the other patches. Based on the feature vector obtained after feature extraction from VGGNet, a dense depth map is created.

$$DDM=\{ v_{x,y}| x\epsilon1\ldots w, Y\epsilon1\ldots..h\} \quad \ldots (1)$$

where, $v_{x,\ y}$ represents the color value of pixels obtained after feature extraction step at location x, y whereas height and width of the feature vector are represented as h and w respectively.

Using laplace operator the depth for pixel x, y is calculated as:

$$4DDM_{x,y} -DDM_{x+1,y} -DDM_{x-1,y} -DDM_{x,y+1} -DDM_{x,y-1} = 0 \quad \ldots (2)$$

Then calculate the median difference of dense depth map and compare it with median threshold value th, if the difference obtained is lesser than median threshold then new difference is calculate using formula:

$$ND_{pq}=|In_{p}-In_{q}| \quad \ldots (3)$$

where, In represent the intensity of a patch p and q. Then the patches are given the binary value using:

$$ND(x,y)=\{0, ND_{pq}\leq th\} \quad \ldots (4)$$
$$\{1, ND_{pq}>th\}$$

All the 0's represent the similarity among the patches so the connected zeroes identifies the suspicious region. In this way the dense depth reconstruction helps to reduce the dissimilarity among the tempered and original patches.

After the depth reconstruction, the similar keypoints available in every patch are extracted using the adaptive patch matching algorithm proposed in paper. [12]

Now using reconstructed patches find the correlation among patches by calculating the correlation coefficient of each reconstructed patch.

Let R_p represent reconstructed patch then the correlation is calculated using:

$$CR_{p}=\{CR_{p}^{1},CR_{p}^{2},\ldots\ldots\ldots\ldots\ldots CR_{p}^{n}\} \quad \ldots (5)$$

where, CR_p represents the correlation coefficient of the reconstructed patch. Using CR_p calculate the threshold for all the patches and the similar patches can be found by using the threshold values of all the patches as given in the equation:

$$TR_{p}=\{TR_{p}^{1}, TR_{p}^{2}, TR_{p}^{3},\ldots\ldots.. TR_{p}^{n}\} \quad \ldots (6)$$

$$S_{p}=\{S_{p}^{1}, S_{p}^{2}, S_{p}^{3},\ldots\ldots\ldots\ldots\ldots S_{p}^{n}\} \quad \ldots (7)$$

where, TR_p represent the threshold for reconstructed patch and S_p represent the similar patch pair with respect to their threshold. And then the matched keypoint are extracted using similar patch pair are given as :

$$M_{kp}=(M_{kp}^{1}, M_{kp}^{2}, M_{kp}^{3},\ldots\ldots\ldots. M_{kp}^{n}\} \quad \ldots (8)$$

At the end the matched keypoints are merged with the segmented patches and the suspected forged area can be exposed. If all the suspected forged areas in all scales are combined using the "OR" operation then the miss rate can be reduced and the error detection rate can be enhanced. To do this remove the entire wrongly identified forged region while merging. Now calculate the pixel appearance time as:

$$T = \{T_{min},T_{min+1},\ldots\ldots\ldots T_{max}\} \quad \ldots (9)$$

where, T_{max} is the max value for which the pixel appears in all scales. To calculate the probability of the random variable T, calculate the standard deviation (σ) and mean (μ) of T using the formulas:

$$T\mu= \frac{1}{max - min} \sum_{i=min}^{max} \text{ti} \qquad \dots (10)$$

$$T\ \sigma= \overline{\frac{1}{max - min} \sum_{i=min}^{max} (ti - T\mu)\ 2} \qquad \dots (11)$$

Now to take off wrongly identified pixels the merging threshold is taken as Tμ-2Tσ and all the pixels having appearance time less than the threshold value are removed. To merge the entire suspected forged region in all scales the equation used is:

$$R(x,y)= \{1, T\mu\text{-}2T\sigma \leq \sum_{i=1}^{n} \text{fi}(x, y) \leq T\max \}$$
$$\{0, 0\leq \ {}_{i=1}^{n} fi(x, y) < T\mu\text{–}2T\sigma\} \qquad \dots (12)$$

R(x,y) represent the merged region, and fi(x,y) is the suspected forged area in the ith scale. Using the above equation the tempered areas of the image are determined.

Results and Discussion

The result obtained during the experiment conducted using the proposed framework is shown in Fig. 8. The experiment was conducted using deep learning based VGGNet 16 architecture and MICC-F220 dataset.

To measure the capability of the proposed architecture, a comparison is made with several recently developed algorithms: SIFT algorithm based detection[29], Zernike moment based method[26], hybrid algorithm[28], and Invariant feature algorithm.[30] Various performance metrics to assess the proposed framework are True negative rate, False positive rate, Precision, Recall, F-Score and Accuracy.

The results obtained through the performance metrics are shown in Fig. 8. The detailed explanation of them all is as follows:

True Negative Rate: specifies the negative event rate. If the outcome is closer to 1 then it is considered better. VGGNet has maximum true negative rate of 0.974. TNR formula is:

$$\text{TNR}= \frac{\text{TN}}{\text{TN} + \text{FN}} \qquad \dots (13)$$

False Positive Rate: shows the images which are not detected as forged and were originally not forged. VGG has got minimum of 0.55 FPR whereas other [26, 28-30] has got 0.65, 0.62, 0.7 and 0.82 respectively. The FPR is calculated using:

$$\text{FPR}=\frac{\text{FP}}{\text{FP} + \text{TN}} \qquad \dots (14)$$

False Negative Rate: It shows that amount of forged region which the proposed work failed to detect. VGGNet has the minimum FNR of 0.093 as compared to other existing techniques. FNR formula is:

$$\text{FNR}=\frac{FN}{FN + TP} \qquad \dots (15)$$

Precision: is the ratio of correctly detected tempered pixels to the number of total detected tempered pixels. As can be seen in Fig. 8 the precision rate achieved by VGG is 98.01 which

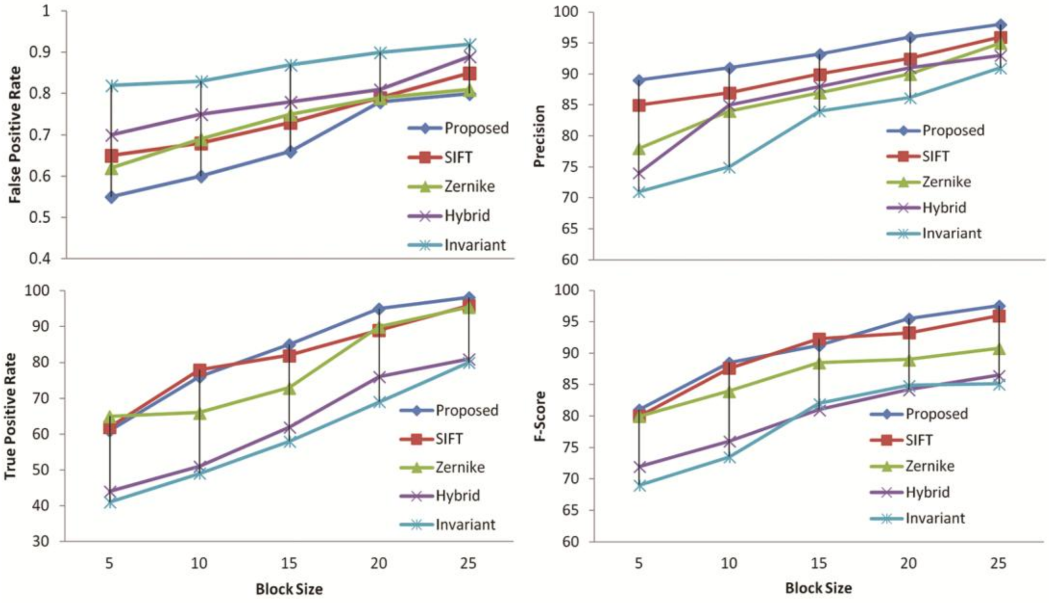

Fig. 8— Graphical representation of results achieved

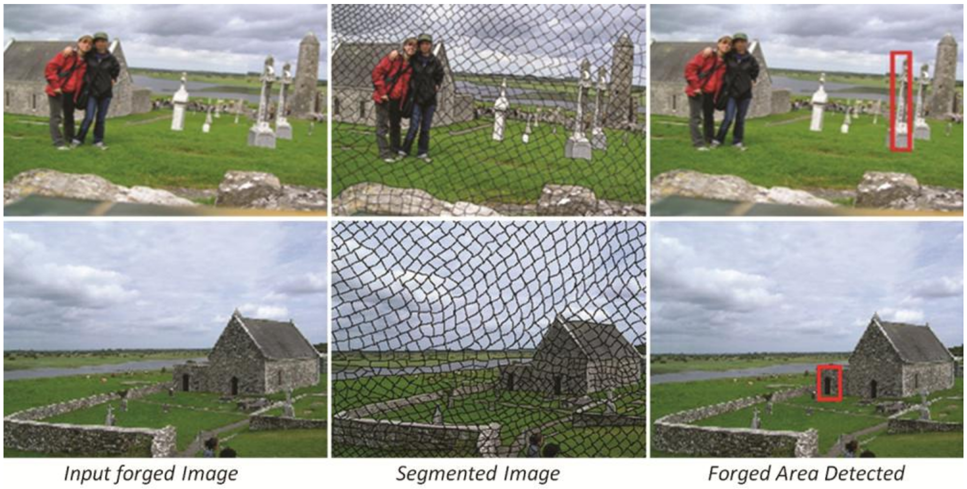

Fig. 9 — Showing results obtained on copy move forged images

is the highest among all. Precision is calculated using:

$$P = \frac{TP}{TP + FP} \quad \dots (16)$$

Recall: the ratio of accurately identified tempered pixels to the real number of tempered pixels in the forged image. VGG has achieved the maximum recall rate i.e. 89.54. Recall is same as True positive rate Recall is considered for performance calculation. Recall is calculated using equation:

$$R= \frac{TP}{TP + FN} \quad \dots (17)$$

FScore: It combines the precision and recall in a single value. The calculated F-score value of the proposed technique is 98.02 which is highest among all. F score can be calculated using the formula:

$$FScore = 2(\frac{P*R}{P+R}) \quad \dots (18)$$

Accuracy: It indicates the number of images which are correctly detected forged image out of all the images. Accuracy achieved by the proposed algorithm is 96%. Accuracy is calculated using:

$$A= \frac{TP+TN}{FP+FN + TP+TN} \quad \dots (19)$$

As can be seen in Fig. 9, the final result obtained after using dbscan to get the refined superpixel and then after extracting the pixels using VGGNet 16 the patches formed will be matched by adaptive patch matching algorithms. Hence, we are able to identify the forged area. The proposed work has better performance than various existing copy-move forgery techniques due to the usage of DBSCAN algorithms which find out the superpixels and also refine them to achieve the highest level of accuracy. The multiscale feature extraction done using VGGNet architecture also picks up the best features that are further used for the matching algorithm to detect the tempered area. The short come of this proposed work is that if there are various copies of the same part that is pasted in the picture then the patch matching algorithm gets confused to match the multiple patches. The proposed work is compared with various existing CMFD techniques (Invariant feature-based method[30], SIFT based method[29], hybrid feature-based method[28], and Zernike moment-based method[26]). The graph shows that the proposed technique performs better than all of the existing techniques.

Conclusion

With the increase of forgeries in images, it is extremely important to come up with a solution. There are various editing softwares which can easily temper an image and various types of transformations can be made which are difficult to detect. Keeping this in view the author has proposed a deep learning-based framework to detect such forgeries. DBSCAN clustering used in the proposed framework helps in reducing the search space and computational cost and most of the forged area is successfully detected with the least false matches. After making a comparison with various other techniques that detect the Copy-Move Forgery, it can be concluded that the proposed work performs much better than the existing solutions in terms of accuracy achieved in detecting forgery. However, the weakness with the current approach is that it does not work well if multiple cloned attacks are made by the forger. In the future, this approach can be enhanced to work for multi-cloned images as well as for the other forgery types such as image splicing in which multiple images are merged to make a forged image.

References

1 Farid H, Image forgery detection, *IEEE Signal Proces Mag*, **26(2)** (2009) 16–25.

2 Tyagi V, Understanding digital image processing, *CRC Press* (2018), ISBN978131512390.

3 Begum M & Uddin M S, Digital image watermarking techniques: A review, *Information*, **11(2)** (2020) 110.

4 Evsutin O, Melman A & Meshcheryakov R, Digital steganography and watermarking for digital images: A review of current research directions, *IEEE Access* (2020) 166589–166611.

5 Teerakanok S & Uehara T, Copy-move forgery detection: A State-of-the-Art technical review and analysis, *IEEE Access*, **7** (2019) 40550–40568.

6 Warif N B, Wahab A W, Idris M Y, Ramli R, Salleh R B, Shamshirband S & Choo K, Copy-move forgery detection: Survey, challenges and future directions. *J Netw Comput Appl*, **75** (2016) 259–278.

7 Christlein V, Riess C, Jordan J & Angelopoulou E, An evaluation of popular copy-move forgery detection approaches, *IEEE Trans Inf Forensics Secur*, **7** (2012) 1–26.

8 Soni B, Das P K & Thounaojam D M, CMFD: A detailed review of block based and key feature based techniques in image copy move forgery detection, *IET Image Proc*, **12(2)** (2018) 167–178 .

9 Zhang Z, Wang C & Zhou X, A survey on passive image copy move forgery detection, *J Inf Process Syst*, **14(1)** (2018) 6–31.

10 Meena K B & Tyagi V, Image forgery detection: Survey and future directions, in *Data Engineering and Applications*, edited by R K Shukla, J Agrawal, S Sharma, G Singh Tomer (Springer, Singapore) 2019, 163–194, https://doi.org/10.1007/978-981-13-6351-1_14.

11 Bilal M, Habib H A, Mehmood Z, Saba T & Rashid M, Single and multiple copy–move forgery detection and localization in digital images based on the sparsely encoded distinctive features and DBSCAN clustering, *Arab J Sci Eng (AJSE)*, **45(3)** (2019) 2975–2992, https://doi.org/10.1007/s13369-019-04238-2.

12 Xiu B, Pun L, Yuan C-M & Chen X, Multi-scale feature extraction and adaptive matching for copy-move forgery detection, *Multimed Tools Appl*, **77(1)** (2016) 363–385.

13 Agarwal R & Verma O P, An efficient copy move forgery detection using deep learning feature extraction and matching algorithm, *Multimed Tools Appl*, **79** (2019) 7355–7376.

14 Jain P, Bajpai M & Pamula R, A modified DBSCAN algorithm for anomaly detection in time-series data with seasonality, *Int Arab J Inf Technol*, **19**(2022).

15 Hegazi A, Taha A & Selim M M, An improved copy-move forgery detection based on density-based clustering and guaranteed outlier removal, *J King Saud Univ - Comput Inf Sci*, **33(9)** (2021) 1055–1063.

16 Singh K N, Singh J N & Kumar K W, Image classification using SLIC superpixel and FAAGKFCM image segmentation, *IET Image Process*, **14** (2020) 487–494.

17 Chen J, Li Z & Huang B, Linear spectral clustering superpixel, *IEEE Trans Image Process*, **26(7)** (2017) 3317–3330.

18 Liu M Y, Tuzel O, Ramalingam S & Chellappa R, Entropy rate superpixel segmentation, in *Proc IEEE Con Comput Vis Pattern Recognit*, 2011, 2097–2104

19 Bergh V D, Boix M, Roig X G, SEEDS: superpixels extracted via energy-driven sampling, *Int J Comput Vis*, **111** (2015) 298–314.

20 Xing Y, Zhong L & Zhong X, Study of clustering algorithm in object tracking and image segmentation, *Wirel Commun Mob Comput* (2022) 1–15. https://doi.org/10.1155/2022/7205929.

21 Levinshtein A, Stere A, Kutulakos K N, Fleet D J, Dickinson S J & Siddiqi K, TurboPixels: fast superpixels using geometric flows, *IEEE Trans Pattern Anal Mach Intell*, **31(12)** (2009) 2290–2297.

22 Shen J, Hao X, Liang Z, Liu Y, Wang W & Shao L, Real-Time superpixel segmentation by DBSCAN clustering algorithm, *IEEE Trans on Image Processing*, **25(12)** (2016) 5933–5942.

23 Martin D, Fowlkes C, Tal D & Malik J, A Database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, *Proc IEEE Int Conf Comput Vis*, **2** (2001) 416–423, doi: 10.1109/ICCV.2001.937655.

24 Simonyan K & Zisserman A, Very deep convolutional networks for large-scale image recognition, (2014), arXiv 1409.1556.

25 Krizhevsky A, Sutskever I & Hinton G E, Imagenet classification with deep convolutional neural networks, *NIPS*, (2012), 1106–1114.

26 Ryu S J, Kirchner M, Lee M J & Lee H K, Rotation invariant localization of duplicated image regions based on Zernike moments, IEEE Trans *Inf Forensics Secur*, **8(8)** (2013) 1355–1370.

27 Chou C L & Lee J C, Copy-move forgery detection based on local gabor wavelets patterns, in *Int Conf Security Intell Comput Big-Data Services* (2017), 47-56.

28 Yang F, Li J, Lu W & Weng J, Copy-move forgery detection based on hybrid features, *Eng Appl Artif Intell*, **59** (2017) 73–83.

29 Yang B S, Guo X, Xia H Z & Chen X, A copy-move forgery detection method based on CMFD-SIFT, *Multimed Tools Appl*, **77(1)** (2018) 837–855.

30 Al-Qershi O M & Khoo B E, Enhanced block-based copy-move forgery detection using K-means clustering, *Multidim Syst Sign Process*, (2018), 1–25.