International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

Copy-Move Forgery Detection in Digital Images using Neural Network

Jigna J. Patel, Ninad S. Bhatt

Abstract: ***Due to easy availability of image editing software applications, many of the digital images are tempered, either to hide some important facts of the image or just to enhance the image. Hence, the integrity of the image is compromised. Thus, in order to preserve the authenticity of an image, it is necessary to develop some algorithms to detect counterfeit parts of an image, if there is any. Two kinds of classic methods exist for the detection of forgery: the key- point based method in which major key points of the image is found and forged part is detected and the block based method that locates the forged part by sectioning the whole image into blocks. Unlike these two classic methods that require multiple stages, our proposed CNN solution provides better image forgery detection. Our experimental results revealed a better forgery detection performance than any other classic approaches.***

Keywords: Copy-Move forgery detection; Convolution Neural Network; Tempered Digital Images, Pixel-Based Image Forgery Detection, Block Based Image Forgery Detection.

I. INTRODUCTION

As the image enhancing application are easily available, many digital images are tempered either for concealing some part of the image or just for fun. Also, these fake images can be used for forensic purpose or utilized by newspaper editor to be published in the newspaper. Therefore, authentication of such images is a must. Images can be forged in many ways such as removal or replacement of some parts of the digital image. It may also include slicing of an image, which involves creation of one image from various images. Among all these image forgery techniques, Copy-Move Forgery Detection (CMFD) is one of the most common technique in which the content of the image is copied from one region and pasted into another area of the same image. The forged part may be geometrically transformed or some post-processing operations, such as rotation, scaling, JPEG compression, noise addition, etc., may be done before pasting. As the tempered region is similar in terms of color and texture, it is very difficult for the human eyes to distinguish the forged part from the original one. Hence, two techniques exist for the detection of copy-move forgery: block-based and key-point based techniques. The block-based methods usually extract features from overlapping blocks of the image, whereas key-point based method extract the points of interest from the entire image.

Revised Manuscript Received on February 28, 2020.
***** Correspondence Author
Jigna J. Patel*, Gujarat Technological University, Chandkheda, Gujarat, India, Email: jigna2012me@gmail.com
Dr.Ninad S. Bhatt, Electronics and Communication Department, CKPCET, Surat, Gujarat, India, Email:ninad.bhatt@ckpcet.ac.in.

But recently, deep learning approach became more popular for the detection of forgery with various tempering approach, like scaling and rotation or a combination of multiple forgery.

II. RELATED WORK

The general framework of copy-move forgery detection consists of standard steps as follows:

i) Pre-processing which generally converts the image into gray scale and then resized.
ii) Feature extraction is the second step into which the image is usually converted into a set of features of interests
iii) Feature matching wherein the similarity or distance between two features is measured and; finally,
iv) Post- processing that includes the use of a set of heuristics to further improve the detection of forged part, e.g. considering holistic matching between set of features on a higher level of consistency to reduce the rate of false alarms and to improve the number of true positives.

Copy-move forgery detection can be further classified in terms of block based approaches, which includes various research works in Polar harmonic transform [1], PCA feature [2,3], Zernike moments [4,5], DCT [6,7], and key-point based methods that comprises of SIFT [8,9], ORB [10,11], SURF [12,13], and irregular region-based methods [14,15].

Every method has its own advantages and disadvantages in the Copy-Move Forgery Detection. For instance, the block-based methods, even though simple, are computationally expensive. On the contrary, key-point based methods are fast and robust against affine transformation but they often fail to identify the forged areas when copied and pasted regions are homogeneous. Essentially, both methods have advantages and disadvantages; key-point based methods may not be suitable for copied smooth regions, whereas block-based methods work well in such cases, yet, it will bring high computation cost. In the recent years, deep neural network (DNN) has been used for the digital image forgery detection research. In [16], the authors introduced a new end-to-end deep neural network, which was robust against several assaults. Meanwhile, in [17], the authors utilized deep neural network for extracting features on behalf of the copy-move forgery identification. Later, in [18], authors used a DNN-based patch classifier for the recognition of counterfeited regions. Furthermore, in [19], the authors proposed an end-to end DNN for splicing detection and localization.

III. METHODOLOGY

3.1 Overview

Since there are several drawbacks of the CMFD methods, as discussed in Sec. 2 of the paper, our goal is to design an end-to-end DNN pipeline that is capable of detecting the forged sections in a digital image. To achieve this, an end-to-end DNN should be created in such a way that it is capable of attaining distinct similar features of the tampered regions, wherein both the copied region and the source regions have more similarity than those of the other pristine regions. Consequently, as inspired by [20], a deep neural network was proposed with an architecture capable of distinguishing all types of tempered regions, which has undergone various transformations before forgery. The proposed DNN architecture is shown in Fig. 1. In the processing module, there was either a set of standard DNN layer or custom DNN layer; hence, cascaded together. Each of these modules has been discussed in detail below.

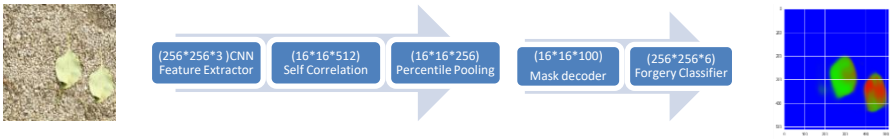


Fig. 1. Overview of the proposed DNN-based CMFD solution

3.2 CNN Feature Extractor

Extracting block-like features for a given image is the primary objective of using the convolutional feature extractor. In comparison with the traditional CMFD approach, such as block-based CMFD or key-points based CMFD, the resulting CNN feature identifies the appropriate feature representations for the image copy-move forgery detection problem from training the data automatically. Apart from the numerous existing feature extractors, the most common extractor, VGG16 CNN feature extractor [23], was selected. In the first four blocks of the VGG16 model, the convolution and max-pooling layers were utilized. As a result, if we input image X of size 256 × 256 × 3, we get as an output, a feature tensor f_s^X of size 16 × 16 × 512 from the VGG16 convolutional feature extractor.

3.3 Clone Detection

The Clone detection branch starts with feature representation via the CNN Feature Extractor. The cloning detection further produces a feature tensor f_s^X of size 16×16×512, which can be seen as 16×16 patch-like features, i.e. $f_s^X = \{f_s^X [ir, ic]\}_{ir, ic \in [0\ldots 15]}$, each with 512 dimensions. Since the objective of the research was to locate the cloned/forged regions, there was the need to search for useful information in order to decide what the matched patch-like features are. For doing so, all-to-all feature similarity score was calculated using Self-Correlation, and then, meaningful statistics were obtained to identify matched patches via Percentile Pooling. Given the two patch-like feature $f_m^X[i]$ and $f_m^X[j]$ where i= (ir, ic) and j = (jr, jc), the Pearson correlation coefficient ρ was utilized to compute for the feature similarity as shown in Eq. (1)

$$\rho(i, j) = \left(\check{f}_m^X[i]\right)^T \check{f}_m^X[j] /512 \quad (1)$$

where $(\cdot)^T$ is the transpose operator, and $\left(\check{f}_m^X[i]\right)$ is the normalized version of $f_m^X[i]$ and it is shown in Eq. (2)

$$\check{f}_m^X[i] = (f_m^X[i] - \mu_m^X[i]) / \sigma_m^X[i] \quad (2)$$

where $\mu_m^X[i]$ is the mean and $\sigma_m^X[i]$ is the standard deviation.

Afterwards, a score vector $S^X[i]$ was established by repeating the process over all with possible $f_m^X[j]$ for a given $f_m^X[i]$. As an output from Self- Correlation, a tensor S^X of shape 16×16×256 was produced. Hereafter, the Percentile Pooling was employed, which first sorted the score vector $S^X[i]$ to $S'^X[i]$ in a descending order, wherein the sorted vector depicted an abrupt drop at some point which can be used to decide on the feature matching. One can decide what feature is matched using this sorted score vector. However, the only drawback of score vector is that its length is dependent on the input size.

Considering this consequence, Percentile Pooling was used to remove this dependency and standardize the sorted score vector through selecting those scores at the percentile ranks of interests. Aside from these, the Percentile pooling aids in reducing the dimensions, as it keeps only a small portion of all scores. Thereafter, we engage the Mask Decoder to gradually up sample feature P^X to the original image size as d_s^X. Finally, Binary Classifier produced a copy-move mask M_s^X to fulfill the task of predicting the similarity between two forged regions, which is as simple as a single Conv2D layer with 1 filter of kernel size (3,3) followed by the softmax activation.

Excluding Self-Correlation and Percentile Pooling modules, all the other modules are either standard or can be fabricated from standard layers. Implementation of Eqs. (1) and (2) was required for Self-Correlation. On the other hand, Percentile Pooling was just a pooling layer, which had no trainable parameters but only a deterministic pooling function. In this regard, backpropagation was conducted similar to the one performed in standard MaxPooling in which only the neuron corresponding to the max receives the gradient.

3.4 Forgery Classifier

The classifier was activated during the training period. The output given was the cloned regions that are manipulated or the similar pixels of interests, respectively. It was, then, assumed that the input image X was of size 256×256×3. Afterwards, features were extracted from the image X using CNN Feature Extractor; the similarity between features was calculated with the aid of Self-Correlation module and lastly, collected useful statistics through Percentile Pooling. Next, the Mask Decoder was utilized,

which unsampled the feature maps to its original image size and finally, binary classifier was initiated to produce a copy-move mask M_s^X at the same resolution of the input image.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

For training, all parameters were initialized using weights from a pre-trained model, which was taken from [20], the VGG16 on ImageNet for CNN Feature Extractor in Clone detection section. In addition, the Adam optimizer and binary cross entropy loss were used. But, for the primary task, the Adam optimizer with categorical cross entropy loss was utilized. At the same time, precision, recall and F1 scores were collated in order to account the CMFD's performance [20]. For the testing image, four values were computed: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). Subsequently, the computation of precision, recall, and F1 scores was accomplished to evaluate the overall performance of the network. The CoMoFoD dataset [21], which contains 200 base forged images and 25 categories (total 5000 images), was utilized. Each category was made by applying post-processing or attacks to the original image category to conceal the falsification in the image. A detailed description of the attacks and settings can be found in [21]. Finally, the data were evaluated using ground truth masks, thereby, distinguishing the forged part from the pristine regions so as to analyze the network's degree of discernibility.

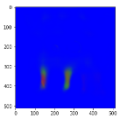
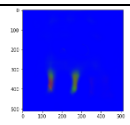
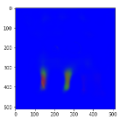
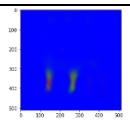
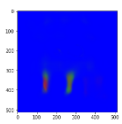
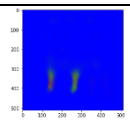
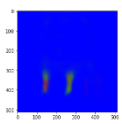
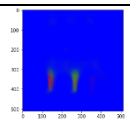
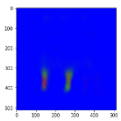
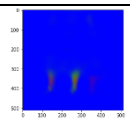
Table 1, as presented below, illustrates the overall performance based on the CoMoFoD dataset.

Table.1 Overall performance on CoMoFoD Dataset

	CoMoFoD Dataset *(Image Level Evaluation Protocol)*
Precision	73.65
Recall	72.9
F1 Score	99

From the table above, it can be concluded that the proposed network provided a 73.65% accuracy over the CoMoFoD dataset.

In order to evaluate the network's robustness against several assaults or post-processing, it was tested against the CoMoFoD dataset as shown in Table 1. Meanwhile, the total number of correctly detected images, which underwent five different attacks and contained 200 samples each, is presented in the Table 2. An image is considered as correctly detected if its pixel-level F1 score is higher than 0.5 [22]. Hence, based on the evaluation of the network performance, the system is quite robust for the detection of various attacks for all images. The overall performance of the CNN model is also summarized in Table 2. Moreover, the proposed network reflected a 73.65% overall accuracy. The performance analysis based on the Image type is illustrated in Figure 2.

Image Type	Original Image	Our Prediction	Image Type	Original Image	Our Prediction
BC1 ***Brightness change*** (lower bound, upper bound) = [(0.01, 0.95)]			***CA1Contrast adjustments*** (lower bound, upper bound) = [(0.01, 0.95)]		
BC2 ***Brightness change*** (lower bound, upper bound) = [(0.01, 0.9)]			CA2***Contrast adjustments*** (lower bound, upper bound) = [(0.01, 0.9)]		
BC3 ***Brightness change*** (lower bound, upper bound) = [(0.01, 0.8)]			CA3***Contrast adjustments*** (lower bound, upper bound) = [(0.01, 0.8)]		
CR1 ***Color reduction*** intensity levels per each color channel = [32]			IB1 ***Image blurring*** averaging filter = [3*3]		
CR2 ***Color reduction*** intensity levels per each color channel = [64]			IB2 ***Image blurring*** averaging filter = [5*5]		

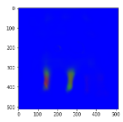
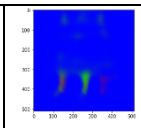
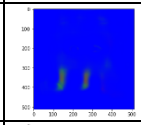
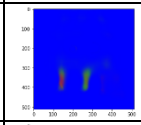
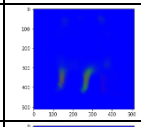
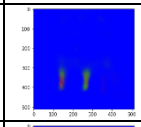
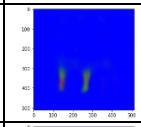
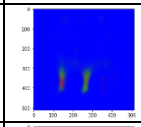
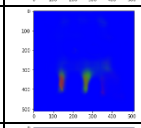
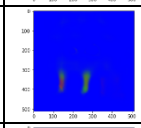
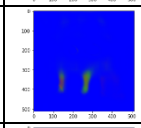
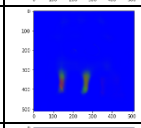
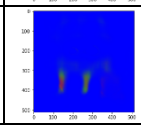
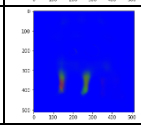
CR3 ***Color reduction*** intensity levels per each color channel = [128]			IB3 ***Image blurring*** averaging filter = [7*7]		
NA1 ***(Noise added)*** μ = 0, σ2 = [0.009]			JC4***(JPEG compression)*** factor = [50]		
NA2***(Noise added)*** μ = 0, σ2 = [0.005]			JC5***(JPEG compression)*** factor = [60]		
NA3***(Noise added)*** μ = 0, σ2 = [0.0005]			JC6***(JPEG compression)*** factor = [70]		
JC1 ***(JPEG compression)*** factor = [20]			JC7 ***(JPEG compression)*** factor = [80]		
JC2***(JPEG compression)*** factor = [30]			JC8***(JPEG compression)*** factor = [90]		
JC3***(JPEG compression)*** factor = [40]			JC9***(JPEG compression)*** factor = [100]		

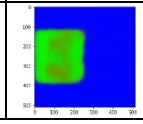
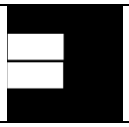
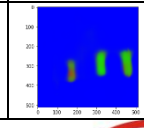
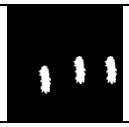
Figure 2. Performance analysis based on the Image type

The total Images correctly detected according to the Image Type is shown in the table below:

Table 2. Performance analysis based on the Image Forgery type

Image Type	Total Correctly Detected	Image Type	Total Correctly Detected	Image Type	Total Correctly Detected
BC1	115	NA3	102	JC2	77
BC2	114	CA1	118	JC3	85
BC3	110	CA2	116	JC4	102
CR1	116	CA3	115	JC5	98
CR2	117	IB1	111	JC6	100
CR3	115	IB2	103	JC7	106
NA1	99	IB3	94	JC8	108
NA2	101	JC1	61	JC9	105

The results of our experiment based on the Image Forgery type are shown in Figure 3

Forgery Type	Forged Image	Our Results	Ground Truth	Forged Image	Our Results	Ground Truth
Translated(1-40)						

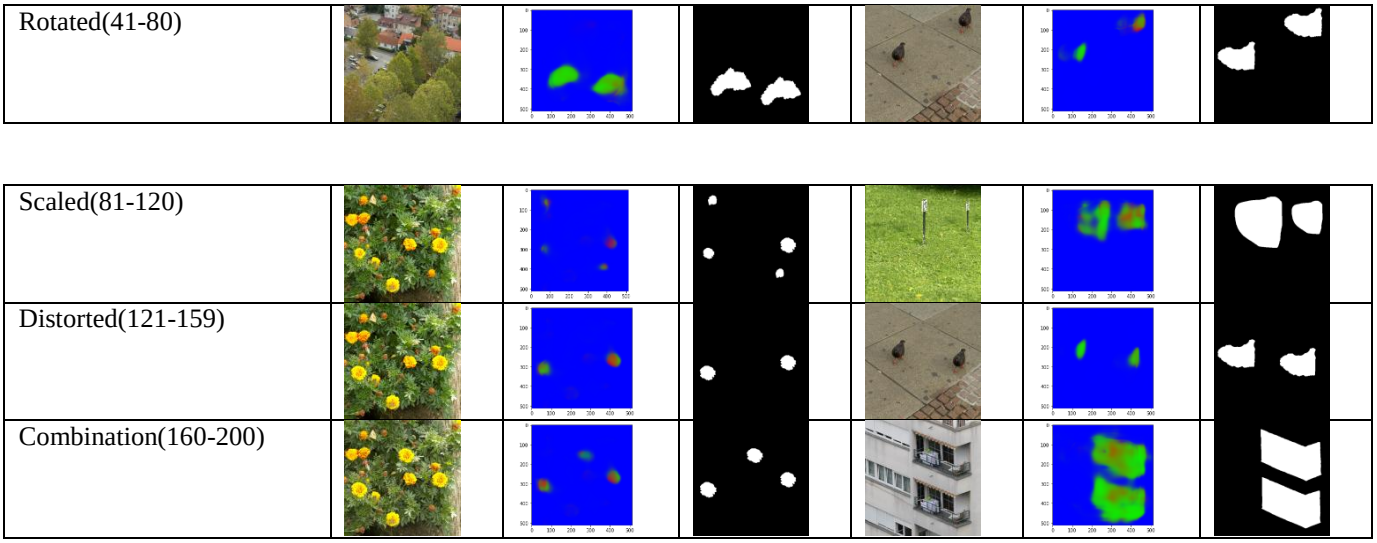


Figure 3. Results based on the Image Forgery type

V. CONCLUSION AND FUTURE WORK

In this paper, an end-to-end CNN solution for detecting copy-move forged images was established. Our evaluation results demonstrated that our method was robust against various known Copy-Move attacks, and also detects several types of forgery that cannot be identified by state-of-the-art algorithms. Hence, this approach can be further optimized for image copy-move forgery detection tasks, like image splicing detection. Also, LSTM can be combined with CNN approach to further optimize the results.

REFERENCES

1. Li L, Li S, Zhu H, Wu X. "Detecting copy-move forgery under affine transforms for image forensics", Computers & Electrical Engineering Volume 40, Issue 6, Pages 1951-1962, Elsevier'August 2014.
2. Huang, D.Y., Huang, C.N., Hu, W.C., Chou, C.H.: Robustness of copy-move forgery detection under high jpeg compression artifacts. Multimedia Tools and Detection Applications 76(1), 1509–1530 (2017)
3. Alaa Hilal ; Taghreed Hamzeh ; Samer Chantaf,"Copy-move forgery detection using principal component analysis and discrete cosine transform", Sensors Networks Smart and Emerging Technologies (SENSET)-IEEEXplore'Dec'2019
4. Seung-Jin Ryu,Min-Jeong Lee,Heung-Kyu Lee," Detection of Copy-Rotate-Move Forgery Using Zernike Moments"pp. 51–65, Springer'2010.
5. Detection of Copy-Rotate-Move Forgery Using Zernike Moments Seung-Jin Ryu, Min-Jeong Lee, Heung-Kyu LeePublished in Information Hiding 2010.
6. Mahmood, T., Nawaz, T., Irtaza, A., Ashraf, R., Shah, M., Mahmood, M.T.: Copy-move forgery detection technique for forensic analysis in digital images. Mathematical Problems in Engineering '2016.
7. Ashima Gupta, Nisheeth Saxena, S.K Vasistha,"Detecting Copy move Forgery using DCT" ,International Journal of Scientific and Research Publications, Volume 3, Issue 5,pp-1-4, May 2013.
8. Hesham A. Alberry,Abdelfatah A. Hegazy, GoudaI Salama, A fast SIFT based method for copy move forgery detection,pp-159-165, Future Computing and Informatics Journal'2018.
9. Yang, B., Sun, X., Guo, H., Xia, Z., Chen, X.: A copy-move forgery detection method based on cmfd-sift. Multimedia Tools and Applications pp. 1–19 (2017).
10. Rajdeep Kaur, Amandeep Kaur,Copy-Move Forgery Detection Using ORB and SIFT Detector,pp-804-813 Volume 4, Issue 4,IJEDR'2016.
11. Zhu, Y., Shen, X., Chen, H.: Copy-move forgery detection based on scaled orb.Multimedia Tools and Applications 75(6), 3221–3233 (2016).
12. Guang-qun Zhang,Hang-jun Wang,"SURF-based Detection of Copy-Move Forgery in Flat Region", International Journal of Advancements in Computing Technology(IJACT) pp-521-529,Volume4, Number17, September'2012.
13. Silva, E., Carvalho, T., Ferreira, A., Rocha, A.: Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes. Journal of Visual Communication and Image Representation 29, 16–32(2015).
14. Li, J., Li, X., Yang, B., Sun, X.: Segmentation-based image copy-move forgery detection scheme. IEEE Transactions on Information Forensics and Security 10(3), 507–518 (2015).
15. Pun, C.M., Yuan, X.C., Bi, X.L.: Image forgery detection using adaptive oversegmentation and feature point matching. IEEE Transactions on Information Forensics and Security 10(8), 1705–1716 (2015).
16. Yue Wu, Wael Abd-Almageed, and Prem Natarajan, "Image Copy-Move Forgery via an End-to-End Deep Neural Network", Winter Conference on Applications of Computer Vision, IEEE'2018.
17. Liu, Y., Guan, Q., Zhao, X.: Copy-move forgery detection based on convolutional kernel network. Multimedia Tools and Applications pp. 1–25 (2017).
18. Bunk, J., Bappy, J.H., Mohammed, T.M., Nataraj, L., Flenner, A., Manjunath, B., Chandrasekaran, S., Roy-Chowdhury, A.K., Peterson, L.: Detection and localization of image forgeries using resampling features and deep learning. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on.pp. 1881–1889. IEEE (2017).
19. Wu, Y., Abd-Almageed,W., Natarajan, P.: Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection. In: Proceedings of the 2017 ACM on Multimedia Conference. pp. 1480–1502. MM'17 (2017).
20. Wu, Yue, and AbdAlmageed, Wael and Natarajan, Prem," BusterNet: Detecting Image Copy-Move Forgery With Source/Target Localization", European Conference on Computer Vision (ECCV),Springer'2018.
21. Dijana Tralic, Ivan Zupancic, Sonja Grgic, Mislav Grgic: Comofodnew database for copy-move forgery detection. In: ELMAR, 2013, 55th international symposium. pp. 49–54.IEEE (2013).
22. Tralic, D., Zupancic, I., Grgic, S., Grgic, M.: Comofodnew database for copy-move forgery detection. In: ELMAR, 2013 55th international symposium. pp. 49–54.IEEE (2013).
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)

AUTHOR'S PROFILE

Mrs. Jigna J Patel, received Bachelor's degree in Computer Science from L. D. Engineering college, Ahmedabad and Master's degree in Computer Science from Sarvajanik College of Engineering and Technology, Surat under Gujarat Technological University. She is pursuing her PhD in area of Image Processing. Her area of interest includes Wireless Communication and Image Processing. She is working as Assistant professor in Dr. S. & S.S. Ghandhy Government Engineering College, under Gujarat Technological University, Surat, India.

Dr. Ninad Bhatt, has received his B.E.degree in Electronics Engineering from VNSGU, Surat, Gujarat in 2001, M.E. in Electronics and Communication Systems from DDU, Nadiad, Gujarat in 2007 and Ph.D.in the area of Speech and Audio compression from VNSGU, Surat, Gujarat. He is currently serving as Professor and Head of Electronics and Communication Department of C.K.Pithawala College of Engineering and Technology, Surat, Gujarat affiliated with Gujarat Technological University, Ahmedabad, Gujarat.He has forty nine research publications in his domain in various international conferences and reputed journals.