International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-1, November 2019

Image Forgery Detection using AKAZE Keypoint Feature Extraction and Trie Matching

Badal Soni, Anji Reddy. V, Naresh Babu Muppalaneni, Candy Lalrempuii

Abstract: ***Image Forgery is an illegal activity in the society as per cyber laws. There are various types of forgeries in which forgery on images is considered as an illegal activity. Image forgery may take place in different ways. One way for doing forgery on images is copy and move forgery which may result in loss of image integrity or authenticity. There are number of popular detection techniques exist such as SIFT, SURF etc., but have high complexity in detection of forgery. Here we have proposed a method to detect the forgery on images which results in loss of integrity or authenticity. In our proposed method we have used descriptor matching using Trie Data Structure The descriptor matching method of implementation using Trie data structure made the complexity of the problem to reduce to O (n log n). Using Key points approach we can verify the integrity of the image. Extracting the features with key points approach is computational expensive task. But there is KAZE method which overcomes this situation. KAZE's method of using non-linear diffusion filtering requires it to solve a series of PDEs. This cannot be done analytically forcing KAZE to use a numerical method called an AOS scheme to solve the PDEs. However, this process is computationally costly and therefore an accelerated version of KAZE was created. The Accelerated KAZE or AKAZE which creates non-linear scale space through Fast Explicit Diffusion for reduce the complexity in extracting the features.***

Keywords: KAZE, AKAZE, SIFT, SURF, FED

I. INTRODUCTION

The images are one of the secured ways to represent the information or data. They are the fastest means to convey information as they are easy to store and understand as compared to texts. There are many editing tools such as MS-paint, Adobe Photo shop etc which can easily do editing of the image. These tools are readily available and image can be forged very easily using these tools. Our visual system can easily identify forgery if the forgery is large enough but when forgeries are done to a very small scale it is hard for our visual system to identify the forged region correctly.

Revised Manuscript Received on November 30, 2019.
***** Correspondence Author
Badal Soni*, Computer Science and Engineering, National Institute of Technology Silchar, Assam, India. Email: badal@nits.ac.in
Anji Reddy.V, Computer Science and Engineering, Lendi Institute of Engineering and Technology, Andhra Pradesh, India. Email: anjisoftware@hotmail.com
Naresh Babu Muppalaneni, Computer Science and Engineering, National Institute of Technology, Silchar, Assam, India.Email: nareshmuppalaneni@gmail.com
Candy Lalrempuii, Computer Science and Engineering, National Institute of Technology Silchar, Assam, India. Email: candylalrempuii@gmail.com

Current software's allow users to create forgeries that are very difficult to be distinguished with the naked eye. It is then we need some methods to recognize it correctly. The difficult to be distinguished with the naked eye. It is then we need some methods to recognize it correctly. The main purpose of forgery is to analysis of check whether there is any manipulation in the meaningful content of an image. We can Find the forgery of any image using different methods. But the system needs to be versatile to all sorts of image forgeries. Moreover the time that the method takes to detect the forgery is also crucial in this internet age where images are transferred to various sources in no fraction of time. Finding this image forgery become Image forgery detection has become one of the main goals of image forensics which are required on various real life events like image evidence in courts, in medical records, financial document. Various known technologies and various new methods are implemented to develop an efficient method using hardware's and the software's available.

There are two types of Copy Move Forgery Detection, first is Block based and second is keypoint Approach [11]. Keypoint based approach has been adopted for this project because of some of the drawbacks of block based Approach such as large time complexity. But there is KAZE method which overcomes this situation. KAZE uses non linear diffusion filtering.

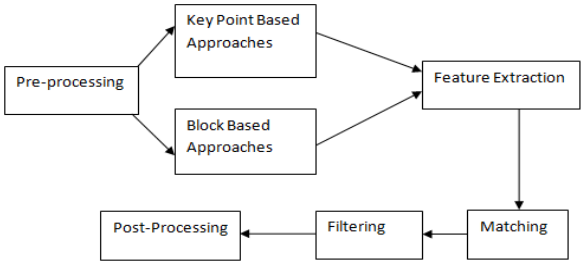

Figure1: Common pipeline processing for Copy-move forgery

1. Pre-processing:
2. Detection Techniques:
   Block Based Approach
   Key-Point Based Approach:
3. Extracting the features
4. Match the copy:
5. Process of Filtering:
6. After processing:

Objectives of this paper are:

(1) To develop a method that should do detection of image forgery using keypoint feature extraction algorithm.

(2) To develop a method that should do Copy move forgery detection even in presence of geometric transformations.

(3) To develop a method that should reduce the computational time.

II. LITERATURE SURVEY

A number of papers have been published for detection of forged regions in an image. Here, some of the related papers that have been taken into consideration and summarized.

2.1 Block-Based Approach

There are five methods in block based approach. They are moment-based; dimensionality reduction-based, frequency-based, intensity-based and textured based [10]. *Popescu et al.* [1], PCA is applied to these blocks to yield a reduced dimensional representation that is robust to minor variance in the images such as lossy compression or additive noise. The PCA representation for each block (x_i) is as follows: x^b coordinate that corresponds to S_i. Then consider all the pairs in rows S_i and S_j whose row distances $|i-j|$ is less than a specified threshold. This method is efficient in detecting tempered regions in an image with noise or wasted compression's. Generates less number of false matches. The Drawback is highly discriminative for large block size and its accuracy reduces considerably for small block sizes and low JPEG qualities. *Friedrich et al.* [2], in this paper they propose a method based on Discrete Cosine Transform (DCT). Initially, the image is divided into overlapping blocks by using a fixed size sliding window where the window moves by one pixel along the direction of top-left to bottom-right. For each block, the pixel values are stored in a row and a matrix is formed having row with pixel value corresponding to each block. Lexicographical sorting is then performed to find similar blocks. This method is known as exact match method. *Kumar et al.* [3], here they have implemented a method called improved version of DCT method. It will remove the limitations of popular DCT based block coefficients and are utilized to set a more stronger criterion to establish similarity.

Cao et al. [4], uses a method that employs DCT for copy-move forgery detection. To make an efficient detection, robust features must be extracted from the blocks. So this algorithm focuses on dimension of the feature vector and uses a circular block to represent each block quantified by DCT. For each block four features are extracted and used for finding similar blocks. This implies that the feature vectors dimension is low which means that the method has a low computational complexity. This method is capable of identifying multiple region duplications and is robust against blurring and additive noise .The drawback is that it has poor performance with poor image quality and is not robust to geometrical operations.

Elhem Mohebbian et al. [5], proposed a method that increase the efficiency of DCT method for detection of Copy-Move Forgery in complex and smooth image. Normally like any other method the pre-processing of image is done. Before moving to further steps, it recognizes the complexity of images. The images are divided into two categories: complex and smooth. For smooth images, small block are considered

2.2 Key Point Based Methods

Key Point based methods are such as SIFT, SURF, FAST, BRIEF, ORB, KAZE, and AKAZE. *Lowe et al.* [6], here they developed a method for image forgery detection based on SIFT that can take unique features from images. In which it can be used for reliable matching among different kind of objects in image [12].

Xue et al. [7], the main drawback of SIFT is low speed in computations. To overcome this drawback a method called generation algorithm was proposed. This algorithm improves the speed of computation

match.Junwen et al. [8], here a method called SURF (Speed up Robust Feature)was proposed for identify the forgery in image. This technique was very fast and robust for finding forgery in image effectively. For calculating descriptor, square We can summarize the method of copy-paste forgery detection using SURF also.

BRIEF method *Calonder et al.* [9], proposed a feature point descriptor using Binary strings. Hamming distance is used to identify the similarity of descriptors. The computation is considered faster than rest methods because binary strings are calculated directly from image patches. Pair wise intensity comparison can classify the image patches without requiring a training phase. That is block size is less while for complex images large block size is considered. Then, it recognizes the complexity of images. The images are divided into two categories: complex and smooth. For smooth images, small block are considered *i.e.* block size is less while for complex images, large block size is considered.

III. PROPOSED SYSTEM

The objective of a detection method is to accurately determine the forged region from the original image. The other methods implemented are time consuming and the accuracy is not up to the mark. More over the methods given in the previous chapter reduced the time complexity but it failed to provide good accuracy. The proposed algorithm is better in accuracy and also reduces the time of execution. The proposed method is implemented on *64-bit* Ubuntu Operating System. We used the dataset MICC-F220 and MICC-F2000. MICC- F220 and MICC-F2000 have undergone the different transformations like translation, scaling, rotation, distortion and combination.

3.1. Preprocessing: In this stage the color image is converted to black and white image for further processing make easy. Scale Space Construction: AKAZE uses non-linear diffusion filtering method. It can be described by the following partial differential equation (PDE).Scale Space Construction: AKAZE uses non-linear diffusion filtering method. It can be described by partial differential equation (PDE).

$$\delta L/\delta V = \text{div}(c\,(x, y, t).\Delta L \qquad (1)$$

Where div is divergence, δ is the gradient. *L* is luminance of an image *C* conductivity function. This is used for noising the image *t* scale parameter. Here conductivity function as follows

$$[c\,(x, y, t) = g(|\nabla L\sigma\,(x, y, t)|)] \qquad (2)$$

Where $L\sigma$ is the Gaussian smoothed version of the image $\nabla L\sigma$- gradient of $L\sigma$ Here *g* is Gaussian kernel given bellow:

$$g2 = \frac{1}{1+|\nabla L|\dot{2}/\lambda 2} \quad (3)$$

$\lambda 2$ is used for detecting finer edges. It is called contrast parameter. By using the above method we construct the scale space by using following pseudo code.

$$\forall i, \text{ is } 2...O\times S\text{: if ((mod(i,S)}\equiv 1)) \text{ then } \lambda_i = \lambda_{i-1} \times t_{cp} \quad (4)$$

$$L_i = FED\ (Li-1,\ c\ (G\ (subsample\ (L_i-1),\ \sigma),\ \lambda_i),\ t_i)$$

$$\text{else} \qquad \lambda_i = \lambda_{i-1} \quad (5)$$

$$Li = FED\ (Li-1, c\ (G(Li-1,\ \sigma),\ \lambda i),\ ti) \quad (6)$$

The scale space consists of *O* octaves and *S* sublevels. Images generated arel1...Los. Images in each octave are L1, *L1+s, L1+2s,...* and these are generated by sub- sampling the last image in previous octave. The contrast parameter for an image i is determined by multiplying the contrast parameter for previous octave with a factor t_{cp}. This is used for detecting finer edges. For producing images *L2,..Los*, we use the method. First image in the first octave is the noise of the test image L, L1=G (L, $_\sigma$) and the value of λ blurring of test image L with standard deviation σ.

The conductivity function produces by Parona and Malik model is used in the method. This function takes the current image and contrast parameter as the argument and uses the function defined in and gives the blurred version of the current scale image. Here time value t_i for the current image is calculated by using the current octave o and scale number s as in equation.

$$\sigma_i(o, s) = \frac{1}{2}2^{\left(\frac{o+s}{2}\right)^2}, \quad t_i = \frac{1}{2}\sigma_i^2 \quad (7)$$

FED is used for creating the final scale images list each scale. It uses time value t_i for the i^{th} scale, smooth version of the previous scale image. Than the result of the conductivity function will take as arguments. For approximating Gaussian filter, FED uses iterated box filter. In this method, *M* cycles of n explicit diffusion steps with varying step sizes –Tau– is used to crate scale images. That is for each one cycle (say M=1), FED implements *n* steps to find current scale image *Li* and *n* step size calculated for each step sizes are $\tau_1...\tau_n$. Each step sizes are of different size and the maximum step size τ (max) chosen to be of size 0.25 b by the algorithm.

The cycle length n for the current image *Li* is calculated according to the time value t_i of the current scale and the formulae for the same are given below

$$n = \frac{-1}{2} + \frac{1}{2}\sqrt{1 + \frac{12(t_i - t_{i-1})}{T_{max}}} \quad (8)$$

Step size τ for j^{th} step in FED cycle is determined using

$$T_j = \frac{3(t_i - t_{i-1})}{n(n+1)T_{max}} * \frac{T_{max}}{2cos^2(\pi(2*j+1)/(4*n+2))} \quad (9)$$

Each FED cycle is calculated for *G (Li-1, Sigma)*, with initial *L1 = G (L, zg).* L_i represents the temporary scale image in the jth cycle of FED matrix. Matrix A used for calculation with respect to the method specified in [10]. Scale images Li created after each n cycle is given below

$$L_i^j = (I + T_j A(G(L_{i-1}, \sigma)) L_i^{j-1}, j = 1 \ldots . n-1 \quad (10)$$

Algorithm 1 Copy-Move Forgery Detection

```
1: procedure COPY-MOVE(img)
2: image ← imread(inputImage)
3: grayImage ← cvtColor(image,grayImage)
4: AkazeObject ← Akaze.create()
5: detectAndCompute(grayImage, KeyPoints Vector,
Descriptor Vector)
6: bitset<486> bitset[no of Keypoints]
7: for i ← 0 to no.ofkeypoints do
     copyDescriptor.row(i).datatobitset.[i]
8: end for
9: trieNodeRoot ← NULL
10:  triedNodeRoot ← CreateRootNode()
11:  for i ← 0 to no.ofkeypoints do
                AddDescriptortoTrie(bitset)
12:  end for
13:  for i ← 0 to no.ofkeypoints do
                matchDescriptor(bitset,cutoff)
14:  end for
15:  drawMatchedKeyPointsOnImage(img)
16:  showImage(img)
17: end procedure
```

3.2 Keypoint extraction

Once the nonlinear space has been created, specific points can be extracted. Hessian matrix H_i of each image is calculated which is then multiplied by a normalized scale factors f_i. Here $Hi = sf_i * H_i$. This factor is different for different images. The scale factors f_i for i^{th} image L_i having octave value o is calculated by using the following method

$$Sf_i = \frac{\sigma_i}{2^\sigma} \quad (11)$$

Matrices will contain the values and those values look over for finding greater than the threshold value. Also each key point extracted from the i^{th} image will be consulted with other key points residing on the $i+1^{th}$ scale image L_{i+1}.We take only that value as key point whose response is higher in *Li* as compared to other points in a window at L_{i+1}.

3.3. Extraction of Descriptors

AKAZE uses Modified local Difference Binary (MLDB) method for extracting descriptors. It uses the initial orientation of key points, average pixel values, average values of first derivative in both horizontal and vertical direction in a grid of size 2×2, 3×3 and so forth are calculated for binary comparison. Thus binary comparison will be of size *4* bits for each grid. Descriptors extraction is done in following *3* steps.

Step 1: First of all we will calculate the initial orientation of the current key point.

Step 2: Now after calculating the initial orientation of the keypoint, we will calculate the Average values of the pixel, first derivative and second derivative. For calculating this values we need to calculate the sub – sampling step size and this size is determined from the pattern size which has been used in the algorithm. We can understand this by taking an example, for example if pattern size is *12* for this method and so sample step sizes are *5, [5(2/3)] = 4[5(1/2)] = 3*. Now assume that coordinate of the key point scaled with octave value be (kx, ky). So here we have *12×12* grid that will be divided into *5×5, 4×4* and *3×3* sub regions.

Now each region is rotated by main orientation and average pixel values and derivatives which are in both horizontal and vertical direction are calculated. So the average values generated for 5, 4 and 3 steps are 12, 27 and 48 respectively. These values are embedded into a 1×87 temporary vector T.

Step 3: Finally, in this step we will compare the values and will create the final descriptor. The temporary vector which we created in second step has three parts: 12 from step size 3, from 13 to 39 for step size 4, and from 40 to 87 for step size 5. We will compare the elements in each part with each other and a 1×486 descriptor is created for each point.

3.4 Descriptor matching algorithm

Let *K* be the number of key point's extracted and *A1, A2…* , *AK* are the descriptors, each of size 1×486. In our algorithm we use Trie data structure to match the descriptors. We first construct the Trie with all the descriptors. To construct a tie consisting of all the available descriptors, a Trie of [486] X [2] where 486 id the depth of the Trie and each descriptors are represented by 0 and 1. Each node in the Trie will contain a bit value 0/1 and address to the next node. The descriptor to be matched, we will traverse an edge from root to leaf node using index as the length of the descriptor and number of mismatch encountered. We will increase the index while traversing. If the bit value of the node of the current index is same with the current bit value of the descriptor, the n mismatch will not increase; otherwise we will increase the mismatch. We set a predefined threshold and then only those descriptors are considered as matched if their mismatch is smaller than the threshold value. Coordinates of the key-points corresponding to the matched descriptors are stored in a matrix M. A row of the matrix contains the co-ordinates of the matched key points.

Algorithm 2: Adding Descriptor to Trie

```
Procedure ADD-DESCRIPTOR-TO-TRIE (bitset, rootNode)
  tempTrieNode ← createNewNode()
  for i ← 0 to bitset.size() do
    copyDescriptor.row (i).datatobitset.[i]
    if tempTrieNode.f [Bitset [i]] = NULL then
      tempTrieNode.f [Bitset [i]] ← createNewNode()
    end if
  tempTrieNode ← tempTrieNode.f [Bitset [i]]
  end for
end Procedure
```

Algorithm 3: Descriptor Matching

```
Procedure Match (index, mismatch, trieNode) mismatch
>cutoff
   return
   if trieNode ← f [bitset [index]] ≠ NULL then
      if (index+1) = = Bitset.size() & mismatch >0 &&
m1 > mismatch then
         m1=mismatch
      end if
      temp = node → f [1-bitset [index]] → pt
      return
   end if
Match (index=1, mismatch, node → f [bitset [index]])
end Procedure
```

3.5 Elimination of False match

After descriptors matching, some false matches remained. To remove those false matches AKAZE uses Random Sample Consensus (RANSAC) proposed [11]. In RANSAC, a set of points is selected randomly from the matched key points and transformation matrix *H* is formed. Each matched key points are transformed by the transformation matrix *H* and the distance between the transformed points and matching points are calculated. The distance is calculated and value is less than a predefined threshold γ then considered as true match, also called inliers. Otherwise it is consider as false match or outlier and removed from M. Here a threshold value of γ=3 is used by this method. False matches are shown in the Fig: 3.a and Fig: 3(b) shows the result after using RANSAC which eliminates the false matches. Effect of RANSAC is explained in the Fig: 3(b). Fig: 2(a) is original image and Fig: 2(b) shows the tempered version. source={si—si=(sxi,syi), i∈1..N} and the corresponding key points in the second set is denoted by Target={ti—ti=(txi,tyi),i∈1...N}. In this algorithm, two radius r1and r2 are taken and we extract circle around each key points. Then the radius is incremented by one starting from r1to r2 and PSNR (Peak Signal To Noise Ratio, (which is the ratio between total number of key points inside a circle to the number of key points outside the circle) for each circle of radius *r* is calculated. The radius *r* which gives the maximum PSNR ratio are taken as final radius and corresponding key points are colored.

Figure 2 (a) Original image (b) Tampered Image

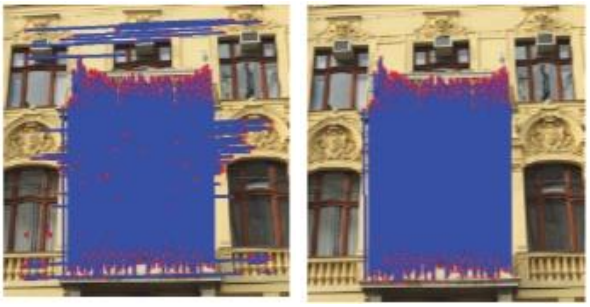

Figure 3 (a) Without RANSAC (b) With RANSAC

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The results have been presented to verify the performance and effectiveness of advanced image forgery detection namely AKAZE taking a common MICC-F220 MICC-F2000 datasets using Open CV, a open-source package for computer vision.

The experiment was carried on a PC with 64 bit Ubuntu 16.04 LTS and IntelR©CoreTMi7-4770 CPU, with a process or speed of 3.40GHz, having RAM of 4GB.

4.1 Datasets available for image forgery detection.

Some of the popular datasets used for the detection of copy move forgery are:

–MICC-F220: 220 images in which 110 original and 110 tampered

–MICC-F2000: 2000 images in which 1300 original and 700 are tampered.

The classification is carried out binary model. A binary can perform classification in which there are two classes for every input pattern that is, either positive (P') or negative (N').The classification performance in the four ways as following

–True positive (TP): Result = P' When ground truth is positive

–False positive (FP): Result = P' When ground truth is negative

–True negative (TN): Result =N' when ground truth is negative

–False negative (FN): Result =N' when ground truth is Positive Using confusion matrix we derived the four cases of the classification (contingency table):

Table 1: Contingency Table

TP	FN	P=TP+FN
FP	TN	N=FP+TN
P'=TP+FP	N'=FN+TN	TOTAL=P+N+P'+N'

Table 2: Average TPR, FPR and time taken for MIC-F220 and MICC-F2000

Data Set	TPR (in %)	FPR (in %)	Time(in sec)
MICC-F220	91.35	10.24	0.873
MICC-F2000	89.75	12.54	1.37

Figure4. a) Original Image b) Resultant Image

Figure 5. a) Original Image b) Resultant Image

Figure 6. a) Original Image b) Resultant Image

Figure 7. a) Original Image b) Resultant Image

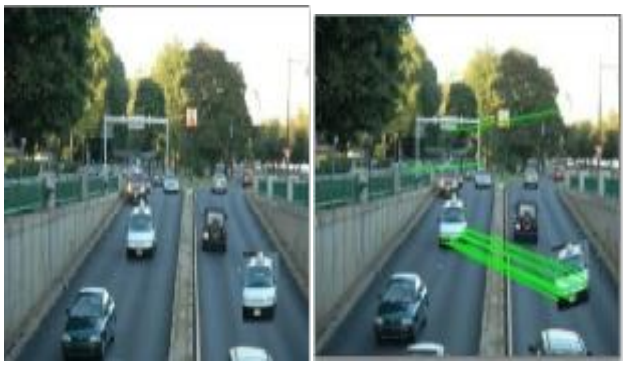

Figure 8. a) Original Image b) Resultant Image

V. CONCLUSION AND FUTURE WORKS

Copy-Move image forgery is causes a loss of integrity and authentication. Existing methods are not addressed with high accuracy and minimum time complexity. The proposed method addressed the above issues. We have used AKAZE method for key point extraction and Trie data structure for matching those descriptors. RANSAC has been used for removing the false matches. Because of using nonlinear scale space, AKAZE can detect the Object Removal with uniform back ground. But the normal brute force matching technique used here has very high time complexity. Since we use Trie data structure in method, it has reduced the computational time sufficiently than the normal brute force matching technique. AKAZ E method was given 90% accuracy in the experimentation on two different data sets.

–For the future work we will try to reduce the computational time with larger datasets.

–For the future work we will try to increase the accuracy and efficiency.

REFERENCES

1. A. C. Popescu and H. Farid,Exposing digital forgeries by detecting duplicated imageregions. 2004.
2. J. Friedrich, B. D. Soukal, and A. J. Lukas,Detection of copy-move forgery indigital images. 2003.
3. S. Kumar, J. V. Desai, and S. Mukherjee,Copy Move Forgery Detection in ContrastVariant Environment using Binary DCT Vectors. 2015.
4. Y. Cao, T. Gao, L. Fan, and Q. Yang, "A robust detection algorithm for copy-move forgery in digital images,"Forensic science international, vol. 214, no. 1-3,pp. 33–43, 2012.
5. E. Mohebbian and M. Hariri, "Increase the efficiency of dct method for detection ofcopy-move forgery in complex and smooth images," inKnowledge-Based Engineer-ing and Innovation (KBEI), 2015 2nd International Conference on, pp. 436–440,IEEE, 2015.
6. D. G. Lowe,Detection of copy-move forgery in digital images. 2004.
7. X. Leng and J. Yang, "Research on improved sift algorithm,"Journal of Chemicaland Pharmaceutical Research, vol. 6, no. 7, pp. 2589–2595, 2014.
8. L. G. Xu Bo, Wang Junwen and D. Yuewei,Detection of copy-move forgery indigital images. 2010.
9. C. S. M. Calonder, V. Lepetit and P. Fua,BRIEF: binary robust independentelementary features. 2010 and P. Fua,BRIEF: binary robust independentelementary features. 2010.
10. Soni, Badal, Pradip K. Das, and Dalton Meitei Thounaojam, "CMFD: a detailed review of block based and key feature based techniques in image copy-move forgery detection" IET Image Processing 12.2 (2017): 167-178.
11. Soni, Badal, Pradip K. Das, and Dalton Meitei Thounaojam, "Blur invariant block based copy-move forgery detection technique using FWHT features" Proceedings of the International Conference on Watermarking and Image Processing. ACM, 2017.
12. Soni, Badal, Pradip K. Das, and Dalton Meitei Thounaojam, "Copy-move tampering detection based on local binary pattern histogram fourier features" Proceedings of the 7th International Conference on Computer and Communication Technology. ACM, 2017.

AUTHORS PROFILE

Dr. Badal Soni was born in Madhya Pradesh, India. He did M.Tech. From Indian Institute of Information Technology Design and Manufacturing Jabalpur, India in did his Ph.D. in Computer Science and Engineering Department, National Institute of Technology Silchar, India. Currently, he is working as an assistant professor in the department of Computer Science and Engineering, National Institute of Technology Silchar. His research interests include Machine Learning, Image Processing, Image forgery detection and Speech Processing. He has published more than 25 papers in repudiated International Journals and Conferences. He is the professional member of IEEE, ACM, and act as a reviewer in many journals papers.

Anji Reddy. V is working as an Associate Professor in department of CSE, Lendi Institute of Engineering and Technology, Andhra Pradesh and pursuing Ph.D. from Computer Science department NIT Silchar. His research area includes deep learning for real time breast cancer cytology images and medical image classification.

Dr. Naresh Babu Muppalaneni working as Assistant professor in the Department of Computer Science and Engineering, National Institute of Technology Silchar, Assam, India. He has completed research projects worth of 2 crore rupees. He has published 3 books, edited 2 volumes and more than 27 papers in different international journals, conference proceedings. He is a life member of CSI, member of ISCA, Senior Member of IEEE. His research interests are Machine Learning, Computational Systems Biology, Bioinformatics, Artificial Intelligence in Biomedical Engineering, applications of intelligent system techniques, Image Processing, and Social Network Analysis..

Candy Lalrempuii is currently pursuing Ph.D in the department of CSE, NIT Silchar. Her research area includes Machine Learning, Artificial Intelligence and Natural Language Processing.