International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4
***Article Received:** 25 January 2023 **Revised:** 12 February 2023 **Accepted:** 30 March 2023*

Cosine Modified K-Means and Neural Network for Classification of Images

Bindu Goyal[1], Vipan Bansal[2*]

[1,2*]Assistant Professor, DAV University, Jalandhar

Abstract: A significant amount of data transfer happens daily through the internet in the form of images, PDFs, and videos. This exchange rate has increased tremendously during the COVID-19 pandemic. However, data transfer consumes a lot of internet bandwidth. It can be reduced significantly if there was a way to determine whether an image is compressed or non-compressed. While much research has been done on image compression in modern photography, detecting whether an input image is compressed or uncompressed has not been studied. This research aims to develop a new algorithm to classify images as compressed or uncompressed. The first step is to propose a new clustering method that uses two random centroids based on randomly selected pixels from the image. The Euclidean distance of each pixel from the randomly selected centroids is used to perform clustering. If clustering fails, then the cosine similarity method is used to perform clustering. This method is called the Cosine modified K-Means method. The SURF feature detection method is used to find the features of each image. Based on these extracted features, a neural network is trained. To test the algorithm, a random image is selected and passed through the network. The algorithm can classify the image as compressed or uncompressed. Precision, recall, classification accuracy, and error rate are calculated to evaluate the performance of the method.

Keywords: Compressed/uncompressed images, Cosine Similarity, Euclidean distance, *K-means* Algorithm, Neural Network, Speeded Up Robust Features (SURF).

1 Introduction

A lot data transfer in the form of sending and receiving of images, PDFs, and video takes place via internet daily and this exchange rate had been boosted up in the COVID-19 pandemic tremendously[1]. The images that had been captured provides vital and detailed knowledge. With certain processing on the image, it loses its information e.g. like on applying any compression technique, along with reduction in size, some of its information is also lost. Upon visual inspection of the image no one can judge whether the image has undergone some kind of changes or whether the given image is compressed or non-compressed. For comparison, there is need of a ground truth image with which the given image can be compared and from the properties of both the images one may find out if the given input image is compressed or not. But in the absence of ground truth image it is difficult to classify the image as compressed or n on-compressed image. The objective of this paper is to design an algorithm that can classify the image without its ground truth image. Nowadays, Artificial Neural Networks (ANN) has been emerging better in contrast to other statistical tools for classification [2]. This paper makes use of three algorithms i.e. ANN, Speeded Up Robust Features (SURF) method for feature extraction and *Cosine modified K-means* to classify the image. Section 2 of the paper covers the preliminaries of image and image compression. In section 3, there is brief on *K-means* algorithm and SURF methodology. In section 4, the step by step proposed methodology is described with algorithm. Section 5 presents the results and discussions and the paper is concluded in section 6.

2. Preliminaries of Image and Image Compression

Generally, image processing methods are implemented and processed in some software either to enhance visual appearance of image for quantitative extraction of features for object recognition. In this section, concepts on the image representation, types of images, and image compression has been put forward.

2.1 Image representation

Images consist of number of dots/ points as an array of numbers called picture element or pixels. An image point is given as its spatial coordinates (x,y) where x and y are horizontal and vertical axes of an image. These values shows the color of intensity at that particular location and set of all values are shown within one matrix [3].

A grayscale image is given as a two layers of values 0 or 1. An RGB image is given as three matrix image for Red, Green and Blue intensity color values as shown in figure 1:

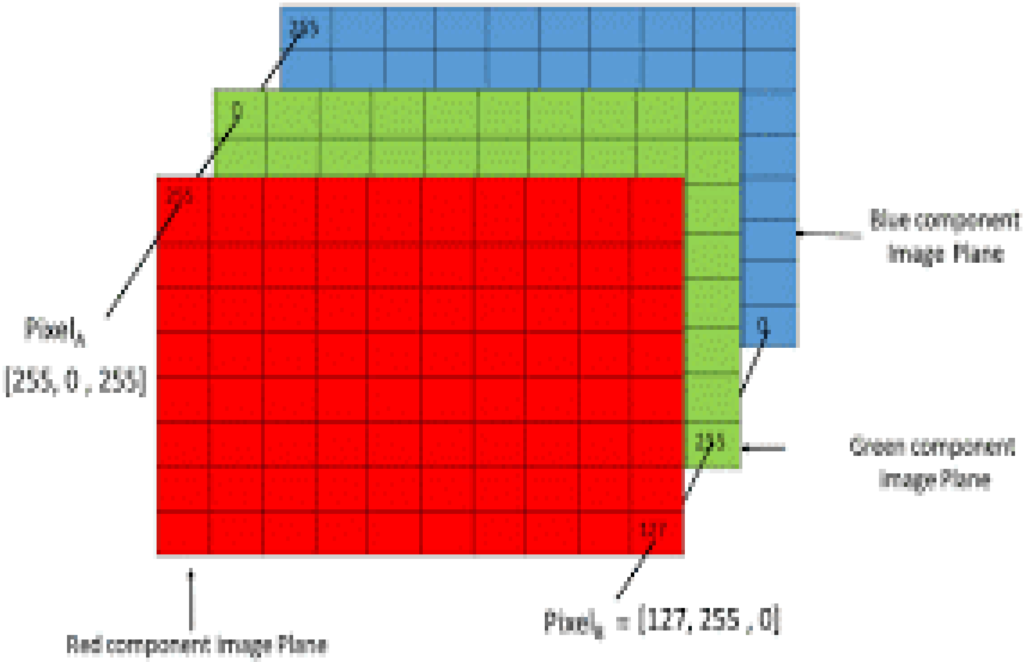

255
0
255
Pixel$_A$
[255, 0 , 255]
Blue component Image Plane
0
255
Green component Image Plane
127
Pixel$_B$ = [127, 255 , 0]
Red component Image Plane

Figure 1: RGB Colour image.

The values of RGB intensities lies between 0-255 each and combination of RGB values represents different color in color space. In the figure above (255, 0, 255) represents one pixel value.

2.1.1 Color Space

Due to additive properties of primary colours red, green and blue, a wide range of colors are generated. The colors one see in digital images, cameras, televisions, and computers are due to RGB color space. This color range depends upon the bit resolution. Each bit represents values between 0 to 255. White color is represented as maximum intensity of all three color channels and black with lowest intensity and shades of grey with equal intensities of all three color channels. In aggregate this is 24 bit RGB color model or true color model and produces 16,777,216 different shades.

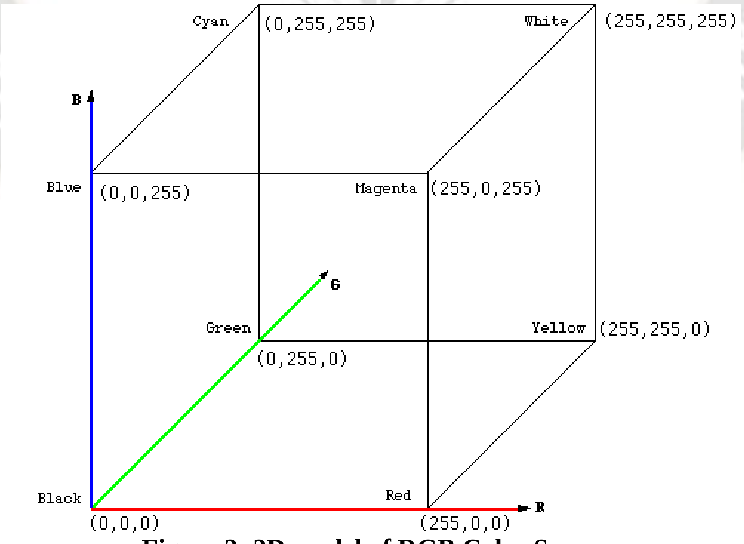


Figure 2: 3D model of RGB Color Space

It can be represented by the figure 2. The three axes represent different Red, Green and Blue values in the range 0 to 255 respectively. Each combination gives a value. It gives three laws of colorimetry as [4].

Law 1: Each combination of primary colours gives unique value and all colours can be formed out of it.
Law 2: Values of colours remain same when multiplied and divided by a scalar value.

Law 3: Total luminance of point equals sum of luminance of each colour value.

2.1.2 Spacial Resolution

To create a digital image, the light received from finite number of sensors is stored in digital form. The number of sensors determines image size and must be able to give image detail and also be efficient to utilize the memory of device and processing time. The figure 3 shows images with different image size. Figure 3(a) is 15x15 image and is just showing basic structure of image. Figure 3(b) is 30x30 pixel image showing better image details and identifiable background details but still flower is difficult to identify. Figure 3(c) gives 60x60 pixel image showing good image details and background objects but is still suffering from pixelization.

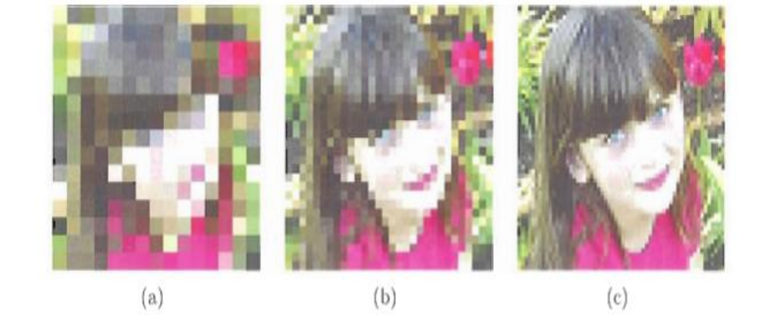


Figure 3: Images with different image size.

The method of using appropriate spatial resolution level is done via sampling criterion. According to the Nyquist sampling theorem, the sampling frequency must be at least twice the highest frequency of the sampled signal to reconstruct a signal from its samples[5].

2.1.3 Digital Image types and their uses:

Mainly, there are 5 types of image types. These are as below:
1. TIFF (.tif)
TIFF stands for Tagged Image File Format. These are large file size images. These images contain a lot of image information so are generally, uncompressed. These can be grayscale, RGB, CMYK. These are used for softwares like Photoshop, and page layout softwares.
2. JPEG (.jpg)
JPEG stands for Joint Photographic Experts Group. These images contain a lot of information in small size and thus are generally, compressed. Digital cameras use JPEG format. JPEG images loses some of image detail so comes under "lossy" compression. These are generally used for web photos as it is useful to load these images easily. These are not used for line drawings as it gives jagged lines appearance due to compression.
3. GIF (.gif)
GIF stands for Graphic Interchange Format. These images are lossless compressed files and are bigger in size than JPEG. GIFs have limited color range so cannot be used for printing and photography. These are suitable for web images and animations.
4. PNG (.png)
PNG stands for Portable Network Graphics. These images have better compression and color range than PNG and were created to replace GIF. It can be used for web images, text art and line art. These cannot be used for print images and photography.
5. Raw images
Raw images have information from a digital camera. These are unprocessed images and cannot be edited or printed. These are generally uncompressed and have large information and large size.

2.2 Image Compression

One of the important segment of image processing is image compression and till date a lot of work has already been done in this segment. Image compression is a data compression technique which encodes the input image in such a way, it reduces the size and redundancy of the image. Generally, in compression along with removing redundancy it also removes some of the vital information and non-repeating information from the image. So, the goal of image compression is to achieve lower the image size without deletion of important information. Considering an example, sometimes on the internet there is some high definition image but upon downloading it just takes few KBs of space on hard drive. This is compressed image. A basic flow chart of image compression is shown in figure 4:

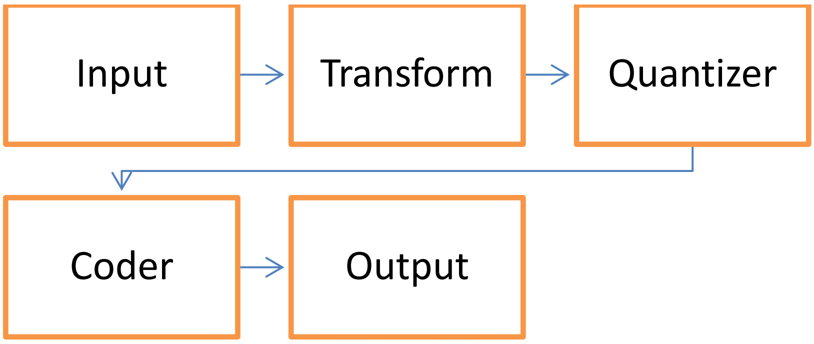


Figure 4: Basic image compression flowchart.

An image is represented as a set of pixels which is represented by number of bits and which in turn determines the intensity of the image. Black and white images are called grayscale images and colour images are called RGB images (Red-Green-Blue Color values). There are three basic redundancies in camera pictures [6], [7]:

a) Psycho visual redundancies which covers the individual's sensitivity to see image details and removing less important information.

b) Inter-pixel redundancies which covers statistical correlations between pixels especially between adjoining ones.

c) Coding redundancies involves coding each pixel of uncompressed image by some set length means like Huffman coding and arithmetic coding to create compression.

2.2.1 Image Compression Standards

With the increasing digital era the requirement for the image compression is also increasing day by day. Broadly, compression can be classified as Lossless and Lossy compression techniques with their individual methods. In this sub-section, the broad categories and their concerned methods will be discussed.

2.2.2 Image compression Formats

There are two broad categories of image compression namely:

i) Lossy image compression
ii) Non-Lossy image compression

Lossy Compression

As the name suggests this technique compresses the data and some of the data (original file) is lost and it is an irreversible process. The more the compression, more the information is lost. This method is efficient in size reduction but there is data loss as well. Figure 5 shows the image output on 50% and 80% compression. Size of the corresponding images is also given here.

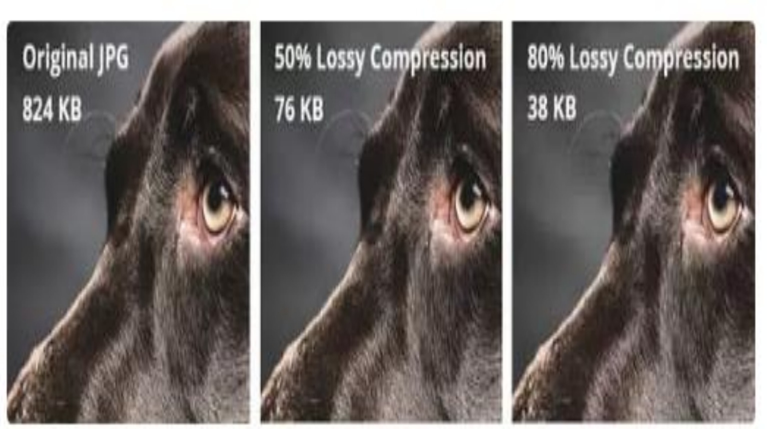


Figure 5[a], [b], [c]: Lossy image compression with compression rate and image size.

Upon visual inspection of the images one cannot find the image differences but upon looking carefully in the image 5[c], there are certain compression artifacts in the darker regions. In conclusion, one can say that if image details are not an issue then size can be considerably reduced with this technique and there are numerous software(s) to do it. But it suffers from the issue of quality degradation, and one cannot get back original image. JPEG images are generally lossy images [6], [7],[8],[9].

Non-lossy image compression

In lossless image compression original information available in the image is retained after compression. It has important applications like text data, computer data and for image and video information. It compresses input only up to some standard i.e. it has small compression ratio. Its examples are Gif and Tiff file formats. Important algorithms for lossless image compression are Huffman Encoding and LZW.

Over the time many algorithms have already been researched till date for image compression. Many methodologies have been implemented and proved better than available technique. No researcher till date focused upon finding a method to check if the input image is compressed or non-compressed. Keeping this drawback in mind, this research has been done and its methodology is presented in the section 4.

3. Algorithms used in Proposed Approach

This section covers the underlying algorithms namely, K-means algorithm and Speeded Up Robust Features (SURF) which serve as a pre-requisite for the proposed methodology.

3.1 K-means algorithm

Due to increasing number of data repositories and high growth the need of machine leaning is being increasing day by day. Broadly, machine leaning can be divided into two parts (i) supervised learning (SL) and (ii) unsupervised learning (USL). For supervised learning (SL) there is a function used to map a given input to a corresponding output from a given set of input –output pairs. Unsupervised Learning (USL) technique does not require any user intervention. The model itself recognizes new patterns and data that were previously undetected. With more recent studies there had been invention of semi SL which lies intermediate to SL and USL. This research paper focuses only upon the K-means algorithm which is a part of USL.

Basically clustering algorithms provides clusters on the input data which have similar attributes bounded by certain rules. So, it provides partition on the input dataset with a defined clustering criteria without any prior knowledge. Clustering has been a very important branch in many applications like computer vision, image processing, bioinformatics, and pattern recognition. Due to its increasing use it is being used in union with image segmentation.

The basic *K-means* algorithm [10], [11] is based upon the concept of decomposition. First step is to consider one value of K and divide the objects in K clusters, which creates high similarity index in the cluster and very low similarity between different clusters. It then creates minimum distance between the cluster values and cluster centre (mean of cluster values). Similarity calculation is based upon the mean cluster object values. The similarity measurement is given by reciprocal of Euclidean distance. The figure 6 shows step by step methodology of K means algorithm:

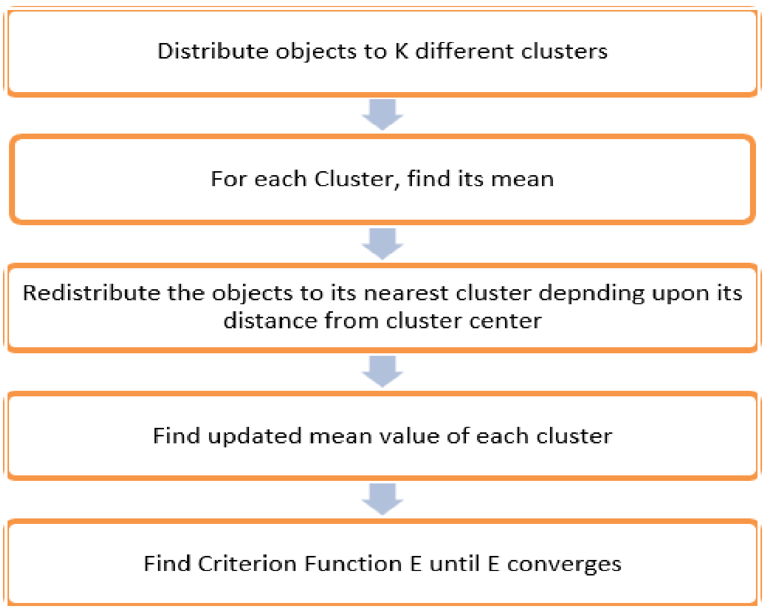

Distribute objects to K different clusters
For each Cluster, find its mean
Redistribute the objects to its nearest cluster depnding upon its distance from cluster center
Find updated mean value of each cluster
Find Criterion Function E until E converges

Figure 6: *K-means* algorithm

This algorithm uses square error criterion as:

$$E = \sum_{i=1}^{i=k} \sum_{p \in C_i} |p - m_i|^2$$

Where E is total square error, p is data object, and m_i is mean value of cluster C_i.

In the recent studies, the K-means has been used in various implementations like Hozumi et al. [12] proposed UMAP-assisted K-means clustering methodology for COVID-19 isolates, Wan et al. [13] proposed fault diagnosis method using Ant Colony Optimisation K-means algorithm, Rezaei and Rezaei [14] proposed a multi objective optimisation variant of K-means algorithm, Gilvaei et al. [15] proposed to use hybrid clustering technique using k-means algorithm

Article Received: *25 January 2023* ***Revised:*** *12 February 2023* ***Accepted:*** *30 March 2023*

and PSO for stochastic programming problem, and in many more applications.

In our approach we have extended the *K-means* algorithm called *Cosine modified K-means* algorithm which is presented in Section 4.

3.2 SURF

The SURF algorithm [16] [17] is an efficient and robust algorithm for similarity invariant , local representation of images. It is an enhanced version of the SIFT descriptor [14]. It extracts the interest areas of an input image which are defined as salient features from a scale-invariant depiction. This algorithm is based on two consecutive functions namely, feature detection and description. The main element of this algorithm is the structure named integral images, which allows us to significantly reduce the number of operations. The SURF algorithm uses a blob detector, the Hessian matrix, to find points of interest [18]. The determinant of this matrix is used as a degree of local change around the point and points are chosen in such a way that the determinant is maximal. To achieve rotational invariance, the orientation of the point of interest is calculated. The Haar wavelet responses in both x- and y-directions within a circular neighborhood around the point of interest are computed. To describe the region around a point, a square region is extracted, centered on the interest point and oriented along the orientation as selected above.

4. The Neural Network based proposed Methodology

This section concentrates upon the methodology of the classification technique proposed. The algorithm is based upon the training of the neural network. Once the neural network is trained any image can be passed from the trained network the output can be obtained. Neural Networks have been widely used in number of fields and applications during recent years. Biswas et al. [19] used neural network for segmenting the retinal blood vessels in human body, Zoughi et al. [20] proposed to use deep neural networks for speech recognition of both gender and phoneme information, Irmak [21]used convolutional neural network for making multi-classification of brain tumors at the early diagnosis stage, Davanipour [22] proposed to use fuzzy wavelet neural network (FWNN) for self-tuning of PID controller. Research shows that neural networks can be used and employed in various fields of research. In this research paper, we have also used neural network whose framework of the proposed method has been given in figure 7.

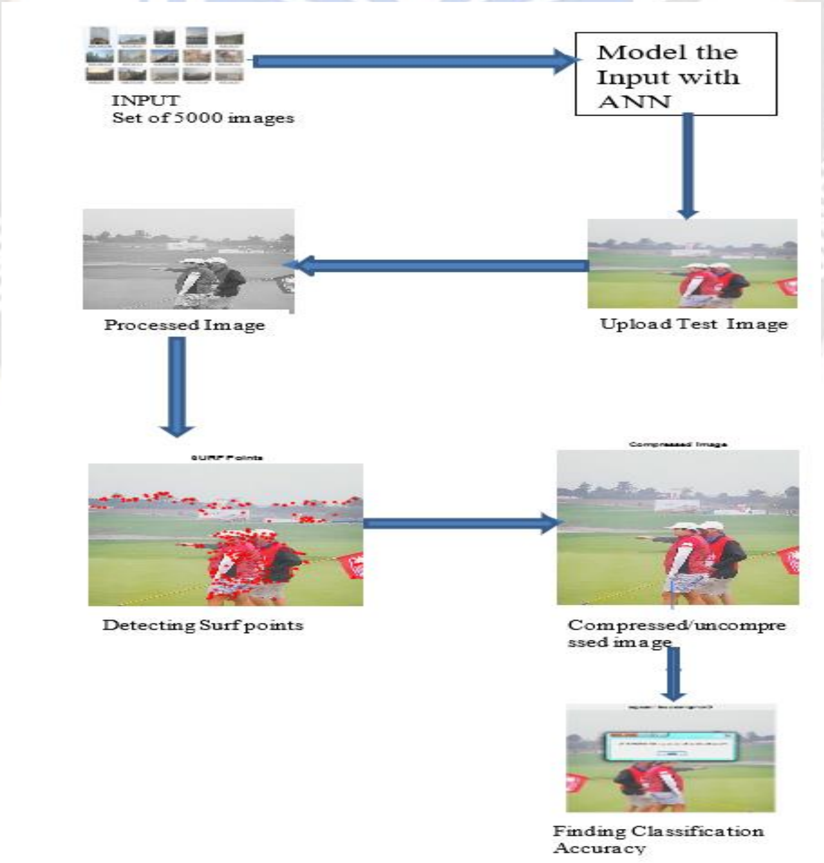


Figure 7: The Proposed Methodology

This algorithm proposes *Cosine modified K-means* classification algorithm. The method makes use of the random function to find the centroid C1 and C2. Euclidean Distance [10], [11], [23] has been used to find the distance of each pixel of image from the C1 and C2. *Cosine modified K-means* algorithm uses cosine similarity index[24]–[28] to classify the image as compressed or non-compressed, if the distinction cannot be done by Cluster 1 and Cluster2. The Speeded Up Robust Features (SURF) method [5], [29]–[31] has been used for extraction of features from the compressed and non-compressed folders and further, to train the neural network based upon the features extracted. Finally, test images have been taken to pass through this trained network.

4.1 *Cosine modified K-means Algorithm*

In the *Cosine modified K-means* clustering method the original k-means algorithm has been modified to improve the accuracy and efficiency. The *Cosine modified K-means algorithm* is as follows:

Algorithm 1 Procedure for *Cosine modified K-means*
Input: {n1, n2, n3…. nn} data items. (n is no. of images here. In this research paper n=10,000)
Output: Images in Compressed and Non-compressed folders (No. of clusters)

Steps:

```
1: For i= 1 to n
2. Compute image-pixel (ip) = 100X100 matrix randomly.
3. Find Mean $M_s = \sum_{j=1}^{ip} \frac{ip(i)}{Count\ of\ ip}$
4. Select random centroids C1 and C2.
5. Set itr =1;
6. while (itr<10)
    Set count1 = count2=1;
Set Group1=Group2=Null;    //Group1 will contain pixel near C1 and Group2 will contain pixels near C2//
For i = 1:ip
Compute d1 as Euclidean distance of each i from C1.
Compute d2 as Euclidean distance of each i from C2.
If d1<d2
Group1 = location of ip
count1 += 1;
else
Group2 = location of ip
count2 += 1;
end
end
7. Record the position of C1 and C2 for each iteration.
8. itr +=1;
9.  Set New Centroid C1 = mean(Group1)
10. Set New Centroid C2 = mean(Group2)
11. If values of C1 and C2 starts repeating

Break;
Endif
12. End while Loop
13. If C1< Mean($M_s$) and elements in Group 1 >= elements in Group 2
Compute PDF1 = mean of normal probability distribution of Group 1
PDF2 = mean of normal probability distribution of Group 2

If PDF1> PDF2
Then write image (i) in compressed folder
Else
IF C1> Mean and Group 1 < Group 2
Then write image(i) in Non-compressed folder
Else Record min value of Group 1 and Group 2 values

If CosineSimilarity(Group1, Group2) >0.95
Then write image (i) in compressed folder
Else write image (i) in Non-compressed folder
End
14. End
15. End
```

Cosine similarity is the measure of similarity between two or more vectors and its value is between the range of 0 and 1. Cosine similarity function can be defined as:

$$cosinesimilarity = \frac{X.Y}{\|X\|\|Y\|} = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2}\ \sqrt{\sum_{i=1}^{n} Y_i^2}}$$

In the above proposed algorithm, the cosine similarity has been calculated only on the pixels of group 1 and group2 in case the probability distribution fails to segregate the pixels i.e. if PDF1 and PDF2 comes with same values. This method has already been used for applications like document similarity and pose making. In this case, we have implemented it in conjunction with K-means algorithm to work for the pixels when their probability distribution falls same. This algorithm has been named as *Cosine modified K-means algorithm.*

4.2 SURF Algorithm

The third part of the algorithm focuses on feature detection and extraction of the images. Once the folders for the compressed and on-compressed images has been created in step 4.1.2, we then extract the features of the images based upon their folders.

In MATLABR2016a the function detectSURFFeatures is used to compute the SURF features. In our methodology the SURF method has been implemented as below:

Algorithm 2 Procedure for *Feature Extraction*
Input: Compressed and Non-Compressed Folders with images
Output: Feature detection of images of different folders
Steps:

```
1.  Select any one folder.
2.  // For Compressed images Folder
For I = 1 to N (No. of images)
Preprocess the image if I is RGB
Detect feature points using detectSURFFeatures
```

Find compressed images features using extractFeatures

3. // For Non-Compressed images Folder
For I = 1 to N (No. of images)
Preprocess the image if I is RGB
Detect feature points using detectSURFFeatures
Find non-compressed images features using extractFeatures

4.3 The Neural Network Model

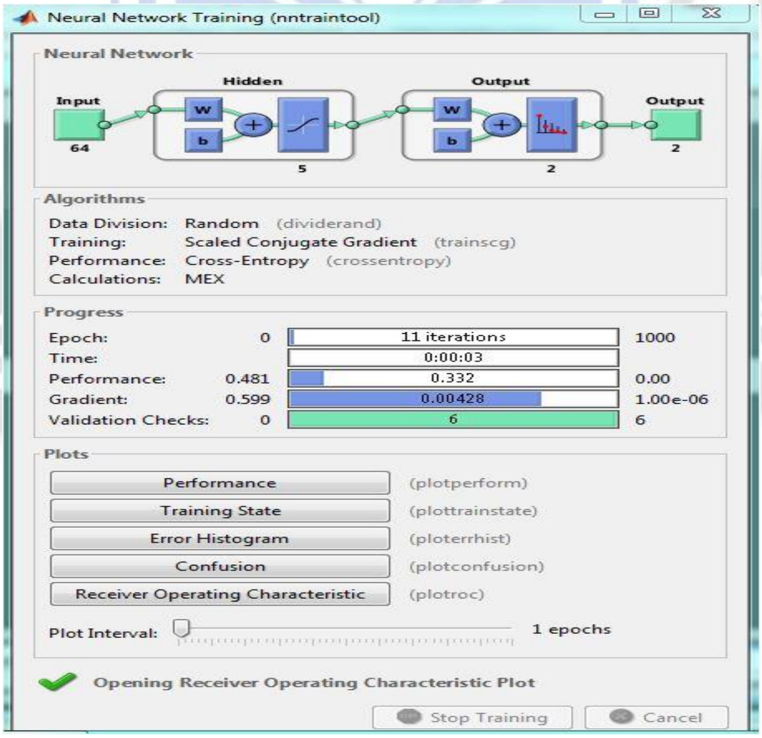
The most important part of this algorithm is neural network model which uses n layers with 2 inputs and 2 outputs. This model uses the set of images and then classifies them in compressed and non-compressed folders, for training, validating, and testing. These components are put together in two groups as training and validating components.in our implementation, we have trained neural network with 5 hidden layers, in which 70%, 15% and 15% of the images will be used for training, validation and testing purpose respectively. We have used MATLAB R2016a for developing, training, validating and testing purpose[32]. Figure 8 represents the Training architecture of neural network. Out of four, the training and validating components are as follows:

4.3.1 Training Components

Training components makes use of the following matrices:
a) The Input Matrix: It includes data for training stage in proposed neural network. This matrix is nX64 logical matrix.
b) The Target Matrix: This matrix includes the decision for each image stored in Input Matrix. In our methodology, this is nX1 logical matrix.
c) The Fitness network: A network with n neurons in the hidden layer of neural network which uses data from above two matrices for training, validating and testing purpose. In our case we have 5 hidden layers, in which 70%, 15% and 15% of the images will be used for training, validation and testing purpose respectively.

4.3.2 Validating Components

Validating components makes use of the following matrices:
a) The Sample Matrix: It contains sample data from the Input matrix which is used as input data at validation stage. In our implementation, it is an nX64 matrix.
b) The output matrix: It contains output data for the sample matrix data. NN predicts the data from the sample matrix and saves it in Output Matrix. In our implementation, it is an nX1 logical matrix.



Data Division: Random (dividerand)
Training: Scaled Conjugate Gradient (trainscg)
Performance: Cross-Entropy (crossentropy)
Calculations: MEX

Epoch:	0	11 iterations	1000
Time:		0:00:03	
Performance:	0.481	0.332	0.00
Gradient:	0.599	0.00428	1.00e-06
Validation Checks:	0	6	6

Performance (plotperform)
Training State (plottrainstate)
Error Histogram (ploterrhist)
Confusion (plotconfusion)
Receiver Operating Characteristic (plotroc)

Plot Interval: 1 epochs

✓ Opening Receiver Operating Characteristic Plot

Figure 8: Training architecture of neural network

Steps for setting data for Neural Network and training Neural Network
1: Initiate Training_data as an empty matrix.
2. // for images in compressed folder
For i = 1 to n
Compute mean of each image and store the results in variable array CurrentFeature
Training_data will be appended with the results row-wise on every iteration.

3. Set Group_Boundary as size of Training_data

***Article Received:** 25 January 2023 **Revised:** 12 February 2023 **Accepted:** 30 March 2023*

```
4. // for images in Non-compressed folder
For i = 1 to n
Compute mean of each image and store the results in variable array CurrentFeature
Training_data will be appended with the results row-wise on every iteration.
5.  Initialize Target=[], count=1;
6. For i=1 to size of Training_data
If i<= Group_Boundary
        Increment Target[1]
Else
Increment Target[2]
End
End
```

7. Create feed forward pattern based neural network with 5 hidden layers. We took default values for the training function and performance method which are trainscg and Cross-entropy respectively.

8. Divide the input data for Training, validation, and test purpose in the ratio 70%, 15%, and 15% respectively.

9. Save and Load the trainedNet.

The training Net has been shown in the figure 8. The figure shows that the data division used in the neural network is Random, Training function used is trainscg, and the performance method used is Cross Entropy. The network took 11 Epochs to train the Neural Network and total 6 validation checks took place on the input data.

4.4 Testing

The final phase of the proposed methodology is to test any input image to be classified as compressed or non-compressed. As discussed earlier this classification is without any ground truth image of the test input image and is based upon feature detection of the images and segregate them as the Compressed or Non-compressed images. The steps to be followed in testing phase are given below:

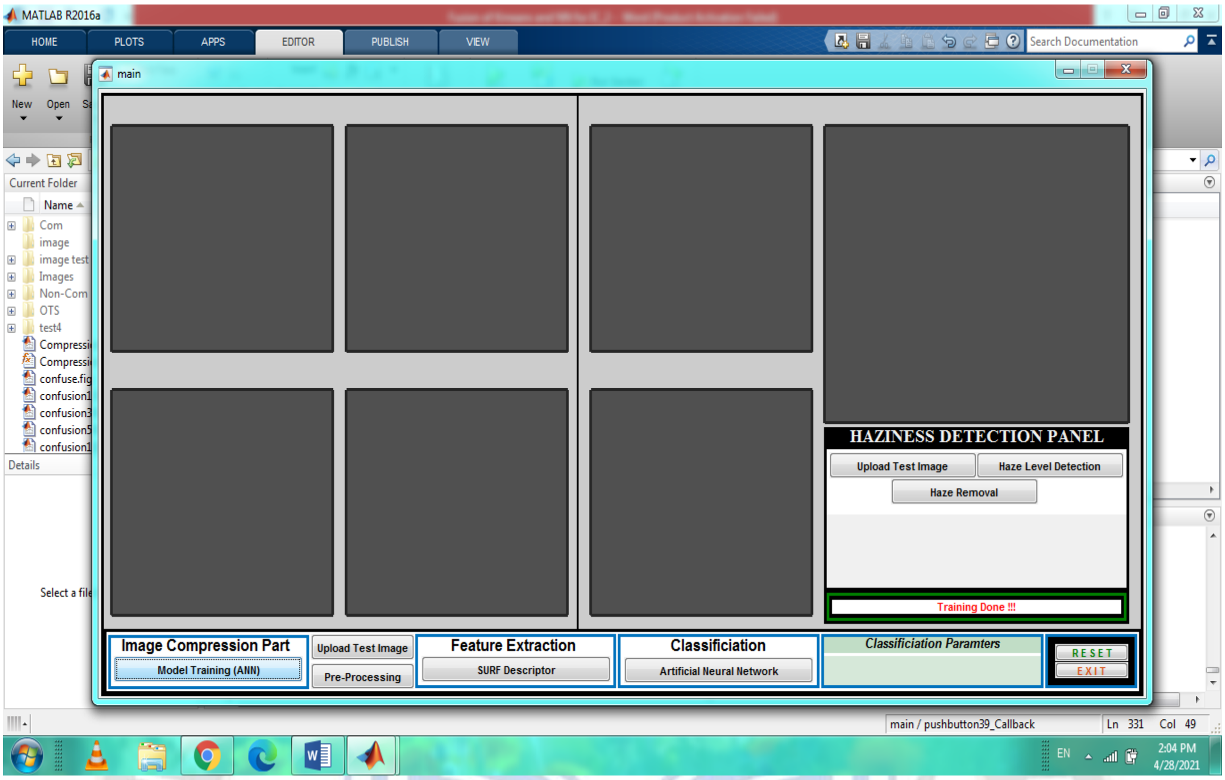
Steps for Testing of data

```
1.  Upload a random test image TEST1.
2.  If Test1 is RGB
Then perform Preprocessing.
End
3.  Perform Feature Detection and Feature extraction using SURF methodology given in 4.1.2. (These are represented as RED dots in the TEST1 image.)
4.  Save SURF_Features.
5.  Load trainedNet (from step 4.1.3)
6.  Simulate the results using sim command in MATLABR2016a as $\mathrm{sim}(\mathrm{Net}, \mathit{Features})$ causing Simulink to simulate the trained net, $Net$, using the parameter values.$Features$.
7.  Find the category wise matching features of the TEST1 with the simulator Results.
8.  If the matching Results matches more for Compressed images
then
Output “Image is Compressed”.
Else
Output “Image is Non-Compressed”.
End
9.  Calculate the number of matching extracted features as Correct Count with respect to Image category.
10. Calculate Accuracy as ratio of (Correct Count/Category)*100;
11. Finish
```

4.5. MATLAB GUIDE

The tool that has been used for implementing the above steps has been chosen as MATLABR2016a. GUIDE is GUI based interface where the data processing is easy using click and show method. Easy commands are there to represent images on the axes and it’s easy to compare the results of the different images. The screenshot of the GUIDE tool used in our methodology is shown in figure 9 where we can see there are easy click functions for Training ANN. Time taken by TRAINING ANN depends upon number of images taken for training purpose. We took 10000 images to train the network and it took about 4 hours to train the network. Once the training is done we need not to perform it every time. For testing purpose click on UPLOAD TEST IMAGE, the selected image will be shown on first rectangular box and the description will be given on the image top. Preprocessed image will come just below the TEST IMAGE. For Feature Extraction click on SURF Descriptor and the output image will be displayed on Top second rectangular box with its description and appearance of red dots to show the extracted features. At the end upon clicking the Classification tab the Final image will be displayed on the bottom second rectangular box with its description as Compressed or non-compressed image. A sample output screen has been shown in figure 10.

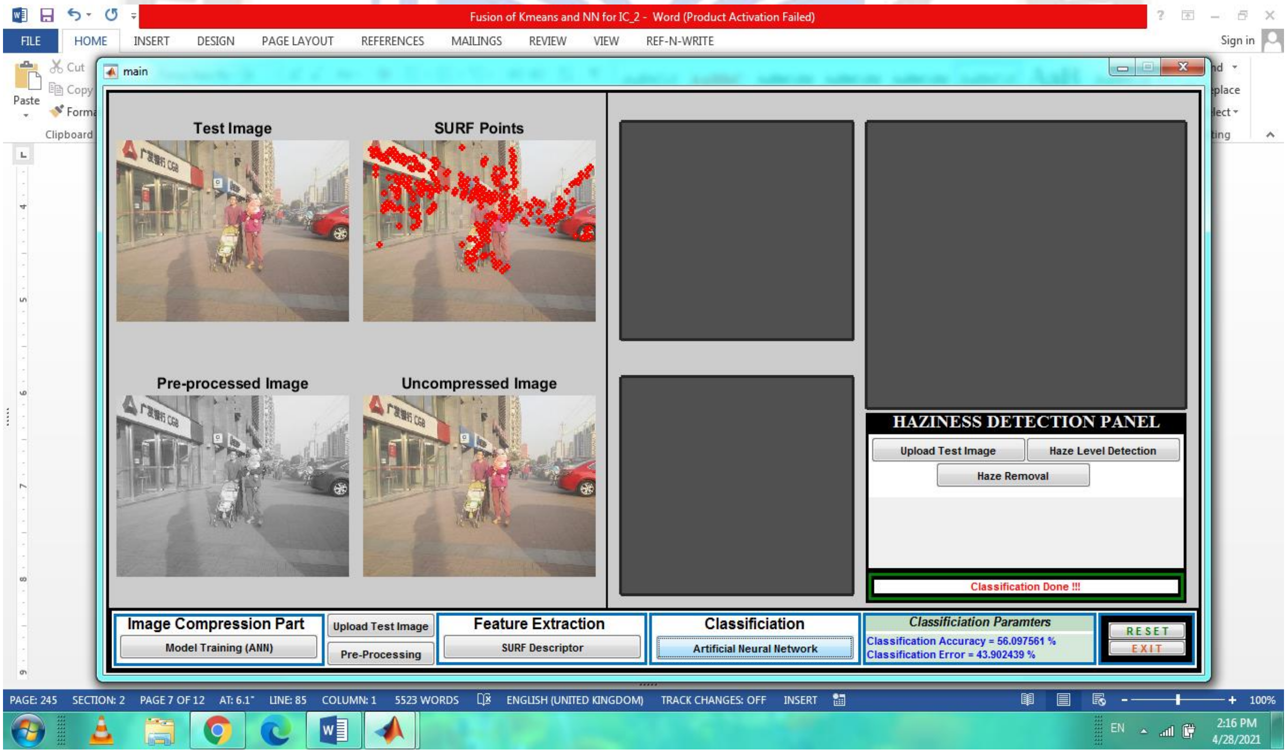
***Article Received:** 25 January 2023 **Revised:** 12 February 2023 **Accepted:** 30 March 2023*



Training Done !!!

main / pushbutton39_Callback Ln 331 Col 49

Figure 9: MATLABR2016a GUIDE result.



Classification Done !!!

Classification Accuracy = 56.097561 %
Classification Error = 43.902439 %

Figure 10: Sample output screen.

The detailed flow chart for the entire process has been given in figure 11 shown below.

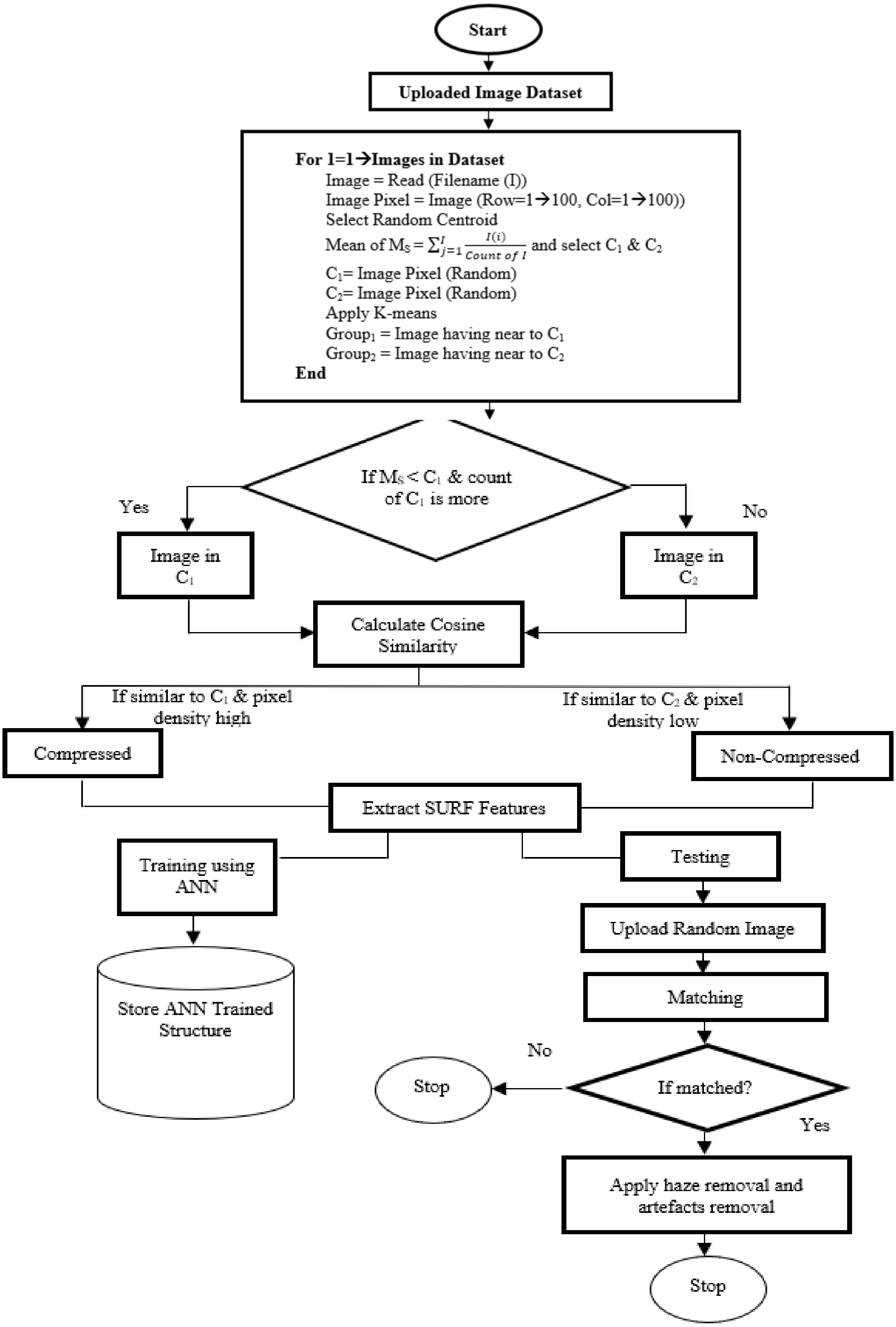

Start
Uploaded Image Dataset
For 1=1→Images in Dataset
Image = Read (Filename (I))
Image Pixel = Image (Row=1→100, Col=1→100))
Select Random Centroid
Mean of $M_S = \sum_{j=1}^{I} \frac{I(i)}{Count\ of\ I}$ and select C_1 & C_2
C_1= Image Pixel (Random)
C_2= Image Pixel (Random)
Apply K-means
$Group_1$ = Image having near to C_1
$Group_2$ = Image having near to C_2
End
If $M_S < C_1$ & count of C_1 is more
Yes
No
Image in C_1
Image in C_2
Calculate Cosine Similarity
If similar to C_1 & pixel density high
If similar to C_2 & pixel density low
Compressed
Non-Compressed
Extract SURF Features
Training using ANN
Testing
Store ANN Trained Structure
Upload Random Image
Matching
No
Stop
If matched?
Yes
Apply haze removal and artefacts removal
Stop

Figure 12: The Training Confusion Matrix

Article Received: *25 January 2023* ***Revised:*** *12 February 2023* ***Accepted:*** *30 March 2023*

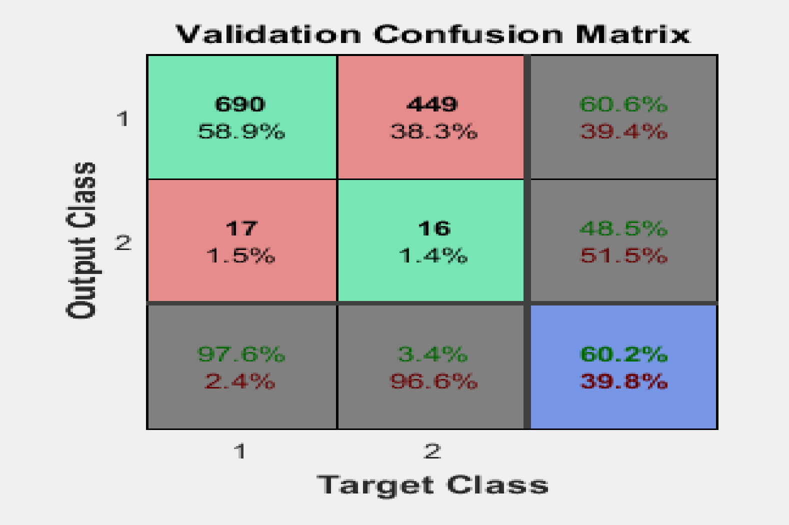


Figure 13: The Validation Confusion Matrix

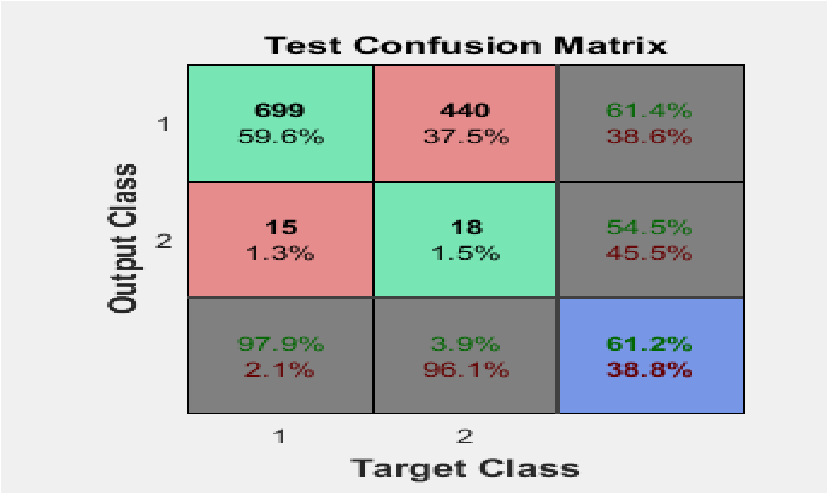


Figure 14: The Test Confusion Matrix

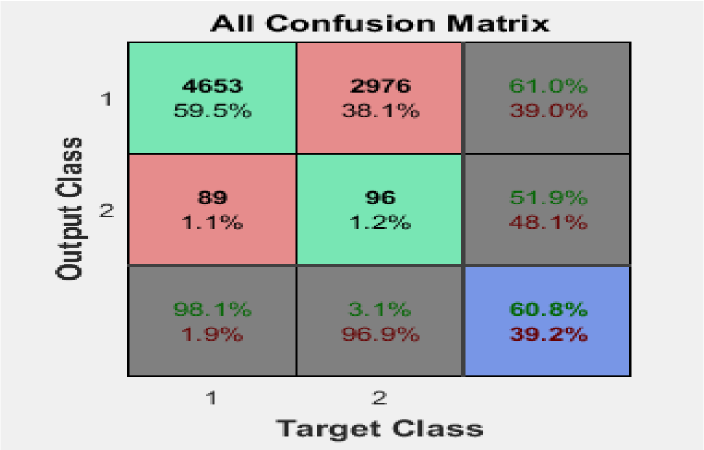


Figure 15: The All Confusion Matrix

***Article Received:** 25 January 2023 **Revised:** 12 February 2023 **Accepted:** 30 March 2023*

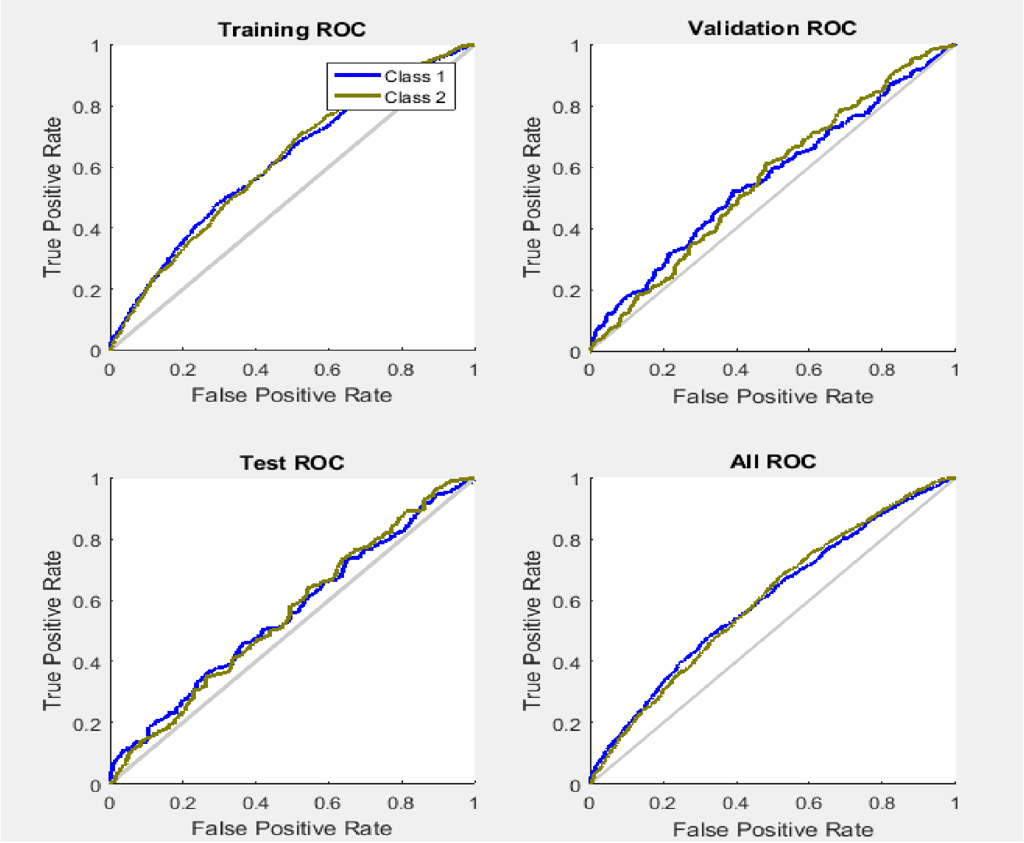


Figure 16: The Receiver Operating Characteristic Curve

5. Simulation Results and Discussion

In this proposed method, *Cosine modified K-means* and SURF technique have been used for identifying an image as compressed or non-compressed. The *Cosine modified K-means* algorithm has been developed to have better accuracy of classification and SURF method has been chosen due to its better efficiency than SIFT algorithm[16]. *K-means* represents the group of machine learning classifier which has been used in number of applications like tumor classification [33], disease segmentation, satellite images[34][27] and many more. In our proposed method, we have used 10,000 images dataset constituting both the compressed and non-compressed images. The standard RESIDE dataset has been used for training purpose which can be downloaded from the link sites.google.com/site/boyilics/website-builder/reside. Then, a neural network has been trained on the basis of input images that performs classification with 5 hidden layers. The confusion matrices for each stage i.e. training stage, validation stage, and testing stage has been shown in Figure 12-14. We have also retrieved the results for all confusion matrix which is combination of all three mentioned stages and has been shown in Figure15. Here, the green squares shows the number of correct responses and the red squares shows the incorrect responses. The gray colored boxes shows the accurate and inaccurate percentages for output class and target class respectively. The blue box contains the overall accuracy and inaccuracy resultant from the gray colored boxes.

Considering the values of the red and blue boxes one can say that outputs are correct which means the neural network has been trained well and it performs correctly. Now, looking at the figure 16 one may find Receiver operating characteristic curves (ROC) for the three phases and one combined for all phases. Basically, ROC is the plot of sensitivity (true positive rate) VS specificity (false positive rate).

From the confusion matrix we can evaluate the various performance metrics as:

a) Accuracy: it is defined as ratio of correct forecasts to total forecasts.

$$ACCURACY\ PERCENT = \left(\frac{TP + TN}{TOTAL\ FORCASTS}\right) * 100$$

b) Error rate: (1-ACCURACY)

c) Precision: it represents the accuracy of positive class.

***Article Received:** 25 January 2023 **Revised:** 12 February 2023 **Accepted:** 30 March 2023*

$$Precision = \frac{TP}{TP + FP}$$

d) TPR/ Sensitivity/ Recall: it represents the computation of positive classes.

$$Recall = \frac{TP}{TP + FN}$$

e) TNR/ Specificity: it represents the computation of negative classes.

$$Specificity = \frac{TN}{TN + FP}$$

f) F1 Score: it is represented by average of precision and recall.

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Abbreviations: TP means True Positives, FP means False Positives, TN means True Negatives, FN means False Negatives, TPR means True Positive Rate and FPR means False Positive Rate.

From the above figure the values of these matrices can be derives as:

$$ACCURACY\ PERCENT = \left(\frac{TP + TN}{TOTAL\ FORCASTS}\right) * 100 = \ 60.78\%$$

Error rate= (1-ACCURACY) *100 = 39.22%

$$Precision = \left(\frac{TP}{TP + FP}\right) * 100 = 3.1\%$$

$$Recall = \left(\frac{TP}{TP + FN}\right) * 100 = 51.89\ \%$$

$$Specificity = \left(\frac{TN}{TN+FP}\right) * 100 = 61\%$$

$$F1\ Score = \left(\frac{2 * Precision * Recall}{Precision + Recall}\right) * 100 = 5.85\%$$

For the purpose of results different test images have been considered along with their results and classification accuracy shown in figure 17-24.

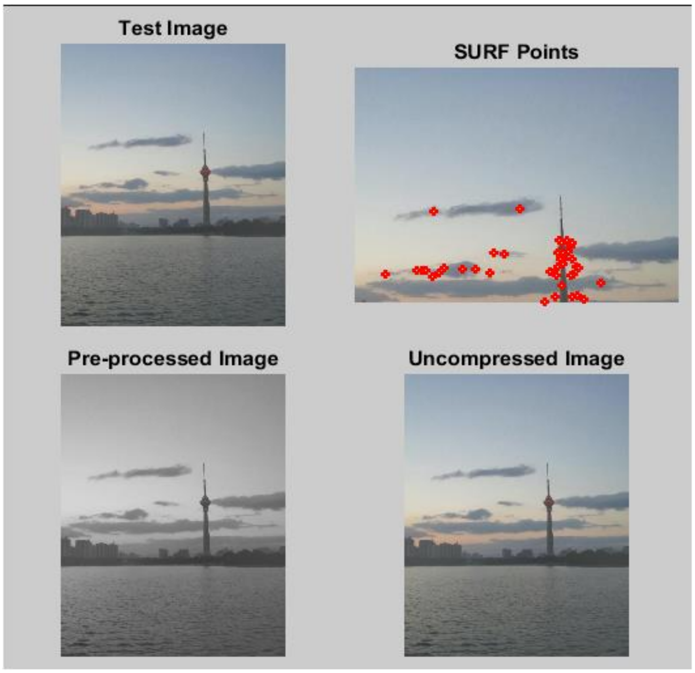


Figure 17: Output results for Test image 1 and with Classification Accuracy = 77.18%

Article Received: *25 January 2023* ***Revised:*** *12 February 2023* ***Accepted:*** *30 March 2023*

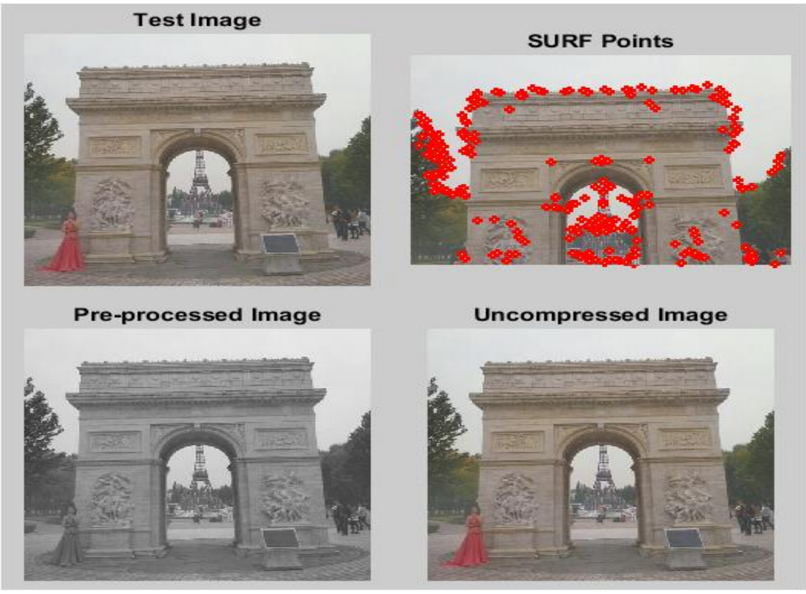


Figure 18: Output results for Test image 2 and with Classification Accuracy = 54.77%

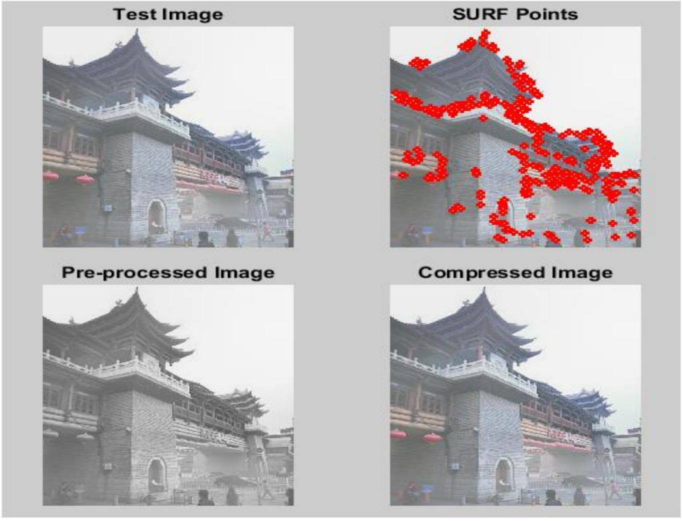


Figure 18: Output results for Test image 3 and with Classification Accuracy = 57.48%

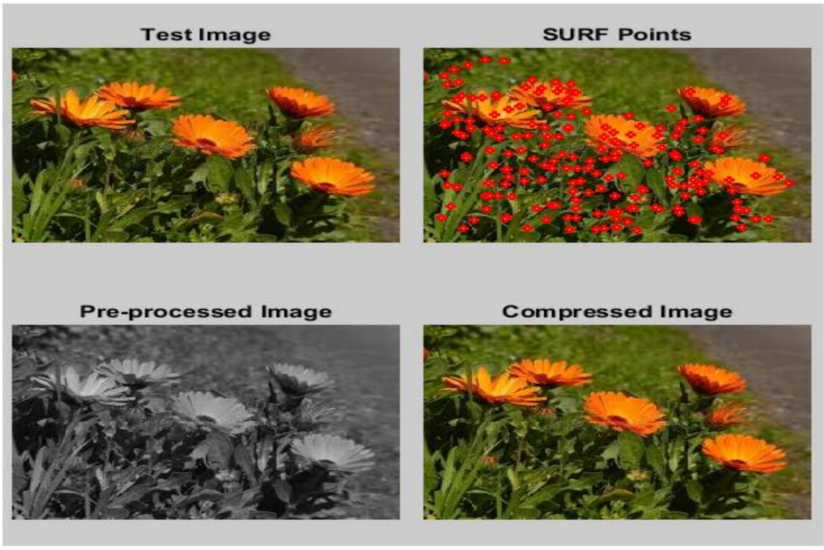


Figure 19: Output results for Test image 4 and with Classification Accuracy = 70.15%

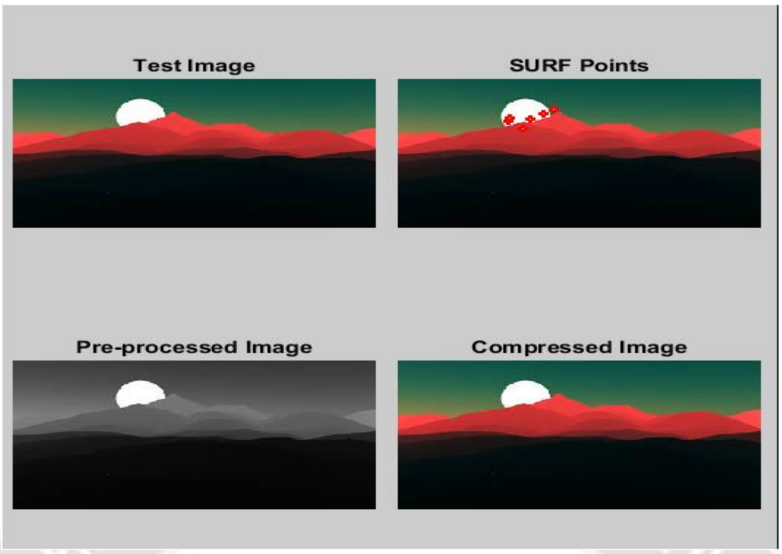


Figure 20: Output results for Test image 5 and with Classification Accuracy = 66.66%

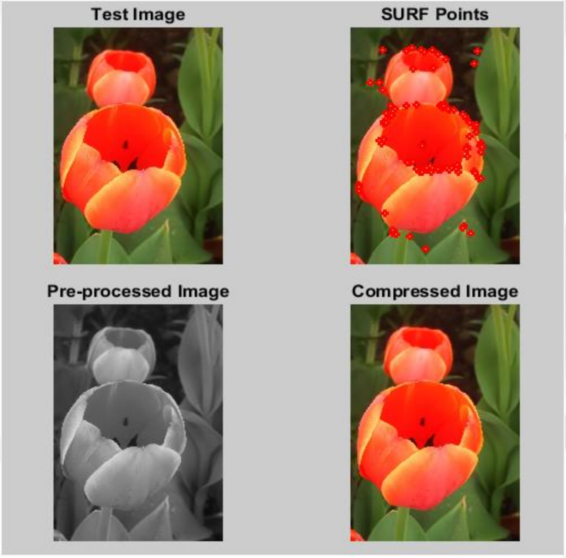


Figure 21: Output results for Test image 6 and with Classification Accuracy = 57.26%

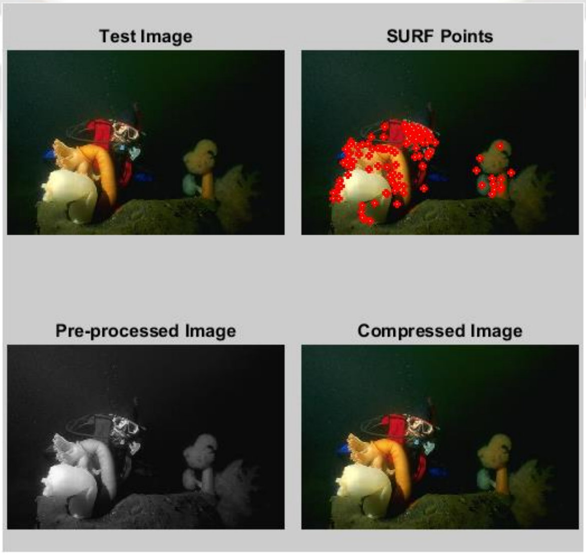


Figure 22: Output results for Test image 7 and with Classification Accuracy = 66.66%

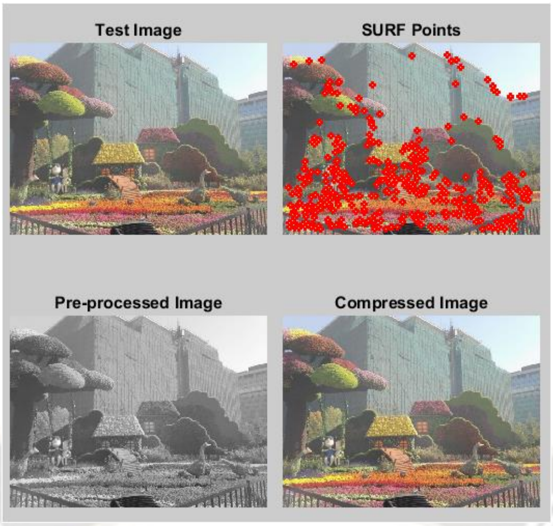


Figure 23: Output results for Test image 8 and with Classification Accuracy = 89.75%

6. Conclusion

This paper concentrates upon a new algorithm for segregating the input image as compressed and non-compressed images. This research paper proposed a clustering algorithm naming; Cosine modified K-Means method. Image classification has been based upon proposed Cosine modified K-Means, the SURF feature detection method and then, training of a neural network. For testing purposes, random image is selected and passed through the network. Outputs show that the images have been classified as Compressed and Non-compressed effectively. Any number of images can be trained in the neural network and in this research paper we have taken 10,000 images for training purpose.

References

[1] D. Mahmassani, H. Tamim, M. Makki, and E. Hitti, “The impact of COVID-19 lockdown measures on ED visits in Lebanon,” *Am. J. Emerg. Med.*, vol. 7, pp. 59–65, 2020.

[2] N. A. Mahmon and N. Ya’Acob, “A review on classification of satellite image using Artificial Neural Network (ANN),” *Proc. - 2014 5th IEEE Control Syst. Grad. Res. Colloquium, ICSGRC 2014*, pp. 153–157, 2014.

[3] T. Alia, I. Mohamad, D. Tabaa, and L. Alrhia, “An Analytical Study on Comparison of Different Image Compression Formats,” *Tishreen Univ. J. Res. Sci. Stud. - Basic Sci. Ser.*, vol. 36, no. 6, pp. 56–80, 2014.

[4] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang, “Color image segmentation: Advances and prospects,” *Pattern Recognit.*, vol. 34, no. 12, pp. 2259–2281, 2001.

[5] G. Kumar and P. K. Bhatia, “A detailed review of feature extraction in image processing systems,” *Int. Conf. Adv. Comput. Commun. Technol. ACCT*, pp. 5–12, 2014.

[6] J. Singh, “Image Compression - An Overview,” *Int. J. Eng. Comput. Sci.*, vol. 5, no. 17535, pp. 17535–17539, 2016.

[7] A. Kaur and J. Singh, “Review on Image Compression Techniques and Advantages of Image Compression,” *Int. J. Adv. Res. Sci. Eng.*, vol. 5, no. 08, pp. 216–221, 2016.

[8] H. Wang and B. Yang, “The Research and Improvement on Dark Channel Prior Image Dehazing Algorithm,” pp. 1–20, 2012.

[9] A. Kaur, J. S. Sidhu, and J. S. Bhullar, “Artifacts reduction based on separate modes in compressed images,” *J. Intell. Fuzzy Syst.*, vol. 35, no. 2, pp. 1645–1656, 2018.

[10] M. D. J. Bora and D. A. K. Gupta, “Effect of Different Distance Measures on the Performance of K-Means Algorithm: An Experimental Study in Matlab,” vol. 5, no. 2, pp. 2501–2506, 2014.

[11] M. Mohibullah, M. Z. Hossain, and M. Hasan, “Comparison of Euclidean Distance Function and Manhattan Distance Function Using K-Mediods,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 13, no. 10, pp. 61–71, 2015.

[12] Y. Hozumi, R. Wang, C. Yin, and G. W. Wei, “UMAP-assisted K-means clustering of large-scale SARS-CoV-2 mutation datasets,” *Comput. Biol. Med.*, vol. 131, p. 104264, Apr. 2021.

[13] L. Wan, G. Zhang, H. Li, and C. Li, “A Novel Bearing Fault Diagnosis Method Using Spark-Based Parallel ACO-K-Means Clustering Algorithm,” *IEEE Access*, vol. 9, pp. 28753–28768, 2021.

[14] K. Rezaei and H. Rezaei, “HFSMOOK-Means: An Improved K-Means Algorithm Using Hesitant Fuzzy Sets and Multi-objective Optimization,” *Arab. J. Sci. Eng.*, vol. 45, no. 8, pp. 6241–6257, Aug. 2020.

[15] M. Nasouri Gilvaei and A. Baghramian, “A Two-

Stage Stochastic Framework for an Electricity Retailer Considering Demand Response and Uncertainties Using a Hybrid Clustering Technique," *Iran. J. Sci. Technol. - Trans. Electr. Eng.*, vol. 43, no. 1, pp. 541–558, Jul. 2019.

[16] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features."

[17] F. Baig *et al.*, "Boosting the Performance of the BoVW Model Using SURF–CoHOG-Based Sparse Features with Relevance Feedback for CBIR," *Iran. J. Sci. Technol. - Trans. Electr. Eng.*, vol. 44, no. 1, pp. 99–118, Mar. 2020.

[18] M. Gabryel and R. Damaševičius, "The image classification with different types of image features," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10245 LNAI, pp. 497–506, 2017.

[19] R. Biswas, A. Vasan, and S. S. Roy, "Dilated Deep Neural Network for Segmentation of Retinal Blood Vessels in Fundus Images," *Iran. J. Sci. Technol. - Trans. Electr. Eng.*, vol. 44, no. 1, pp. 505–518, Mar. 2020.

[20] T. Zoughi and M. M. Homayounpour, "A Gender-Aware Deep Neural Network Structure for Speech Recognition," *Iran. J. Sci. Technol. - Trans. Electr. Eng.*, vol. 43, no. 3, pp. 635–644, Sep. 2019.

[21] E. Irmak, "Multi-Classification of Brain Tumor MRI Images Using Deep Convolutional Neural Network with Fully Optimized Framework," *Iran. J. Sci. Technol. Trans. Electr. Eng.*, pp. 1–22, Apr. 2021.

[22] M. Davanipour, H. Javanmardi, and N. Goodarzi, "Chaotic Self-Tuning PID Controller Based on Fuzzy Wavelet Neural Network Model," *Iran. J. Sci. Technol. - Trans. Electr. Eng.*, vol. 42, no. 3, pp. 357–366, Sep. 2018.

[23] N. Bouhmala, "How good is the euclidean distance metric for the clustering problem," *Proc. - 2016 5th IIAI Int. Congr. Adv. Appl. Informatics, IIAI-AAI 2016*, no. September, pp. 312–315, 2016.

[24] W. Usino, A. S. Prabuwono, K. H. S. Allehaibi, A. Bramantoro, A. Hasniaty, and W. Amaldi, "Document similarity detection using K-Means and cosine distance," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 2, pp. 165–170, 2019.

[25] M. Thangarasu and H. H. Inbarani, "Analysis of K-Means with Multi View Point Similarity and Cosine Similarity Measures for Clustering the Document," *Int. J. Appl. Eng. Res.*, vol. 10, no. 9, pp. 6672–6675, 2015.

[26] L. Sahu and B. R. Mohan, "An improved K-means algorithm using modified cosine distance measure for document clustering using Mahout with Hadoop," *9th Int. Conf. Ind. Inf. Syst. ICIIS 2014*, 2015.

[27] M. Jigar, S. Brijesh, and S. Satish, "A New K-mean Color Image Segmentation with Cosine Distance for Satellite Images," *Int. J. Eng. Adv. Technol.*, no. 5, pp. 2249–8958, 2012.

[28] S. Al-Anazi, H. Almahmoud, and I. Al-Turaiki, "Finding Similar Documents Using Different Clustering Techniques," *Procedia Comput. Sci.*, vol. 82, no. March, pp. 28–34, 2016.

[29] N. Thakur, S. Raju, and A. Gupta, "Feature extraction: an application to object and image recognition," *Int. J. Latest Trends Eng. Technol.*, no. 1, pp. 178–184, 2017.

[30] E. Oyallon and J. Rabin, "An Analysis of the SURF Method," *Image Process. Line*, vol. 5, no. 2004, pp. 176–218, 2015.

[31] Z. Zhu, G. Zhang, and H. Li, "SURF feature extraction algorithm based on visual saliency improvement," no. 3, pp. 13–17, 2018.

[32] N. M. Sheykhkanloo, "Employing Neural Networks for the detection of SQL injection attack," *ACM Int. Conf. Proceeding Ser.*, vol. 2014-Septe, no. September 2014, pp. 318–323, 2014.

[33] N. Arunkumar *et al.*, "K-Means clustering and neural network for object detecting and identifying abnormality of brain tumor," *Soft Comput.*, vol. 23, no. 19, pp. 9083–9096, 2019.

[34] D. Stathakis and A. Vasilakos, "Satellite image classification using granular neural networks," *Int. J. Remote Sens.*, vol. 27, no. 18, pp. 3991–4003, 2006.