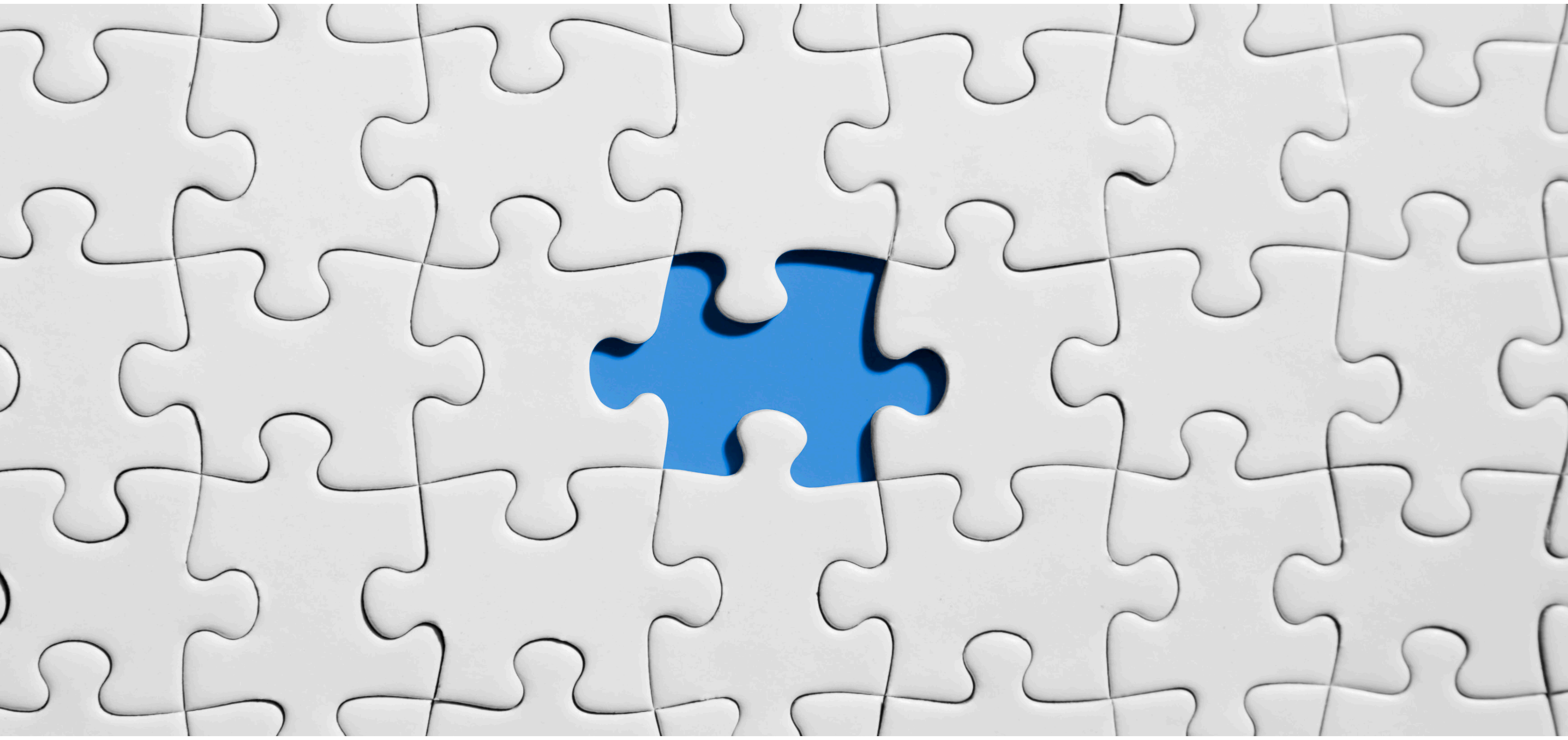
HUMAN EMOTION RECOGNITION USING MACHINE LEARNING



Rajasekhar Nannapaneni
Sr Principal Engineer, Solutions Architect
Dell EMC
Rajasekhar.nannapaneni@dell.com

Knowledge Sharing Article

Table of Contents

List of Figures

Glossary

COD	Cascade Object Detector
HOG	Histogram of Gradients
KNN	K-Nearest Neighbor
MSE	Mean Square Error
NN	Neural Networks
PCA	Principle Component Analysis
PSNR	Peak signal to noise ratio
S/N	Signal to noise ratio

Abstract

Human machine interaction plays a significant role in human-machine partnerships. These interactions can be made possible only through effective communication, be it verbal or non-verbal.

Emotion recognition, one of the crucial non-verbal means by which this communication occurs, helps identify the mood and state of the person. Human machine collaboration becomes more natural if communication happens through the non-verbal means, such as emotions.

Machines can offer us more help if they are able to perceive and recognize human emotions. Of the several approaches to emotion recognition, facial expression and speech-based methods are prominent. According to some psychologists, communication occurring through facial expressions account for about 55% of communication.

In this article a novel machine learning algorithm is developed that uses human facial expressions for emotion recognition and a comparative study is made with existing algorithms.

A1.1 Introduction

At the dawn of human machine partnerships, communication plays a vital role in taking these interactions further. Humans obviously prefer natural ways of communication with machines using the languages that we developed over the years. Besides language that we speak, the indirect natural communication between humans are emotions which form a logical means of messaging.

It would be beneficial if the machines are able to understand human emotions, enabling communication to take another step forward. There are several ways in which this emotion recognition can happen and primarily the focus is on speech and facial-based emotion recognition. There are also many techniques to achieve these facial and speech-based emotion recognitions including deep learning and classical machine learning algorithms.

The aim of this article is to leverage non-deep learning-based or, in other words, classical machine learning techniques so that it requires less compute and complexity. Simpler techniques such as COD, HOG and KNN Classifier are used to perform facial expression-based emotion recognition.

Part A of this article gives an overview of facial expression and speech-based emotion recognition techniques. It details the techniques used to perform facial expression-based emotion recognition such as HOG and KNN classifier.

Part B provides design and implementation of facial expressions-based emotion recognition for a benchmark dataset known as JAFFE dataset. The test results were captured, and the performance of the algorithms documented.

The implementation of the two techniques are done in MATLAB and the code is mentioned in the Appendix.

A1.2 Facial expression in emotion recognition

The important role that human machine interaction plays is possible only through effective communication. Both verbal and non-verbal ways serve communication and emotion recognition is one of the crucial non-verbal means by which this communication happens. Emotion recognition can be captured through different mechanisms such as speech, facial expression, body gesture, etc. According to some psychologists, communication happening through facial expressions account for about 55% of communication [1].

Facial expressions communicate a lot without any voice. Facial expressions are caused by movement of facial muscles in various positions resulting in various emotions and mood. Facial expressions are key to conveying feelings, attitude, intentions, etc. and these are pivotal in recognizing the emotions. Emotion recognition through facial expression is becoming popular due its various applications like robotics where interaction between machines and humans is important. Facial expression analysis for emotion recognition is also used in other applications such as security measures via biometrics and in surveillance. [2]

A1.3 Parameters and challenges in facial expression detection

Facial expression detection is a complex task as it involves identifying different shapes, poses, variations, etc. of the face. Eyes, mouth and eyebrows are some of the key features that are detected and analyzed to identify the emotion. Other crucial parameters in facial expressions are nose wrinkle, [5] lip tightener, inner brow raiser, upper lid raiser, outer brow raiser, mouth stretcher, lip corner depressor, lip parts, etc. help in determining the emotion. Thus, the nasio-labial, brows, eyes, forehead, cheeks and mouth are regions of interest where the movement of underlying muscles causes the various emotions.

Challenges to detect facial expressions include occlusion and illumination effects due to noise on the images that restrict the clarity to detect the facial expressions. In addition to noise, the presence of a beard, glasses, makeup, and hairstyles impact the facial expression detection causing inaccurate emotion recognition. Age of the person, lightening conditions, birth marks, ethnicity and backgrounds often add complexity to existing challenges.

A1.4 Analysis of techniques for facial expression recognition

Facial recognition was initially done by Ekman and Friesen who categorized human emotions into six categories; happiness, fear, sadness, disgust, surprise and anger. [3] Movement of facial muscles allows the variations in cheeks, nasio-labial, eyes, brows and mouth. These movements result in representation of different emotions via facial expressions. Figure 1.1 details various measurements of the key facial features which help to determine the emotion. [4]

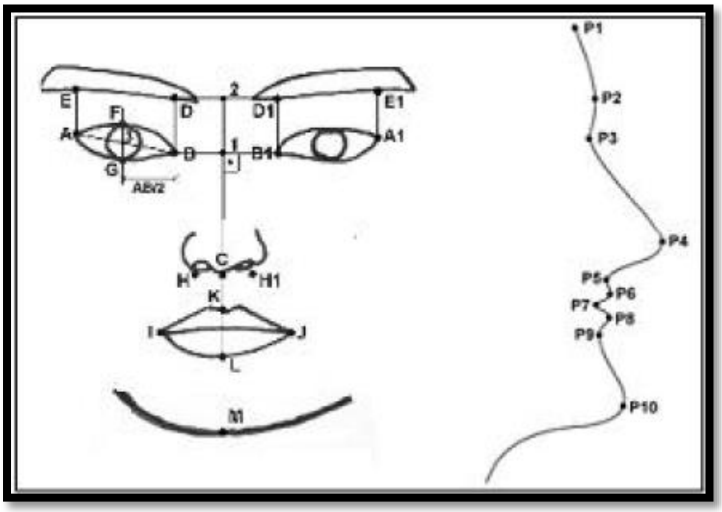


Figure 1.1: Facial feature relationship

Movement of facial muscles due to emotions result in variations of facial feature changes. Figure 1.2 lists various facial feature changes that represents different emotions. Detection of facial feature changes results in identifying the corresponding emotion from the images. [8]

Feature change	Happy	Sad	Anger	Fear
Nose flare	—	—	X	—
Brow distance	—	X	X	X
Brow upward curve	—	X	X	X
Brow curvature	—	X	—	X
Brow height	—	X	X	X
Upper eyelid height (visible sclera)	X	—	X	—
Lower lid position	X	X	X	X
Outer edges of mouth	X	X	—	—
Width of mouth	X	X	—	—
Low lip position	X	X	X	X

Figure 1.2: Mapping of facial features to emotions

A1.5 Facial expressions vs. speech methods for emotion recognition

Emotion recognition helps in identifying the mood and state of the person. Human machine partnership would gain momentum and becomes natural if collaboration occurs through non-verbal communication such as emotions. Machines can offer more help to humans according to human needs if they are able to perceive and recognize human emotions.

Of the several approaches to emotion recognition, facial expression and speech-based methods are prominent. Facial expression method is based on the movement of human facial muscles whereas speech-based method recognizes emotions through variations in tone and energy in the human voice.

Experiments have been conducted separately on the same people to recognize emotions from a given set of audio and visual recordings. [7] Compared to visual facial expression-based approach, the audio speech-based methods such as automatic speech recognition techniques were not efficient in classifying sadness which was misclassified as neutral and vice versa. Similarly, happiness is misclassified to anger and vice versa due to presence of long note duration for both anger and happiness. Both sadness and neutral emotions usually carry brief note conveying silence causing misclassification.

A1.6 Facial expression-based emotion recognition

Emotion recognition plays a vital role in human machine interaction and makes this communication more natural compared to verbal-based techniques. Facial expression-based identification results in efficient emotion recognition. The underlying muscle movements in the human face caused by emotions express variations in facial features which when captured by facial expression identification methods, can decode the emotion expressed by the humans.

Though there are other ways to identify human emotions – particularly through speech-based methods – facial expression-based approaches seem to perform better. Speech-based approaches misclassify certain emotions such as neutral, sadness and happiness, angry which degrades the performance of those approaches. Thus, one can consider that emotion recognition through facial expression-based approaches are superior to other methods.

A1.7 Machine learning algorithms used in this article

Of the various machine learning algorithms that exist, two are used in this article that could extract the relevant features and classify the emotion of the humans; HOG and KNN classifier techniques.

Though deep learning approach is another viable alternative, non-neural network-based techniques like HOG and KNN require less compute and yet provide reasonable efficiencies in determining the human emotions.

The given human images are initially preprocessed through cascade object detector (COD) which detects or segregates the face of the human from the image. This extracted face is further cropped to obtain the human mouth of the image and the mouth images are used to identify the emotions.

The extracted mouth images are sent to HOG to obtain feature descriptions and these features are then classified using KNN classifier into various emotions. In this section, the HOG feature descriptor and KNN classifier are discussed.

A1.7.1 Histogram of Gradients

Histogram of gradients (HOG) is a well-known approach in the field of computer vision that can provide feature descriptions for any given image.

A sliding window is moved over a given image and the orientation of each of the pixels is captured within the sliding window. These orientations are also called gradients and all the gradients in a sliding window form a gradient vector. These gradient vectors are used to form a histogram which reduces the matrix dimensions of the gradient vector significantly.

The magnitude of these gradients that are stored in the form of histogram are then normalized to make them robust to changes in illumination of the image. Figure 1.3 gives a quick glimpse of what HOG does.

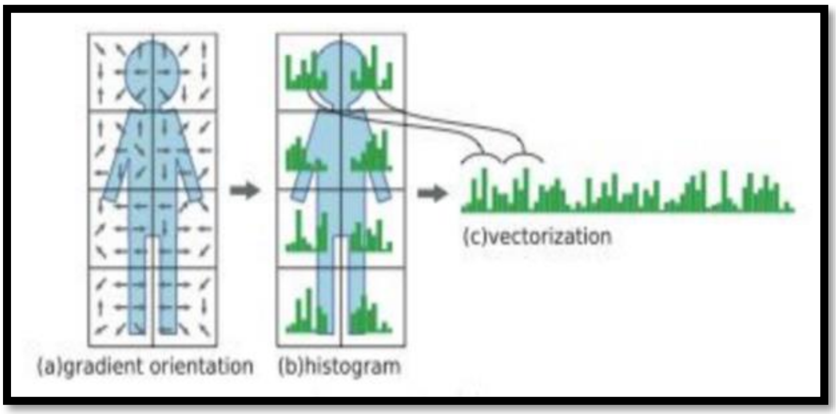


Figure 1.3: Histogram of gradients calculation

A1.7.2 K-Nearest Neighborhood

K-nearest neighborhood (KNN) classification technique is one of the simplest techniques where the constant “k” is pre-defined based on the dataset. This algorithm considers the similarity of the data points primarily using the distance measures within the dataset which is dependent on the number of neighbors “k” to be considered for a specific data point to be classified. Figure 1.4 gives an overview of KNN classification.

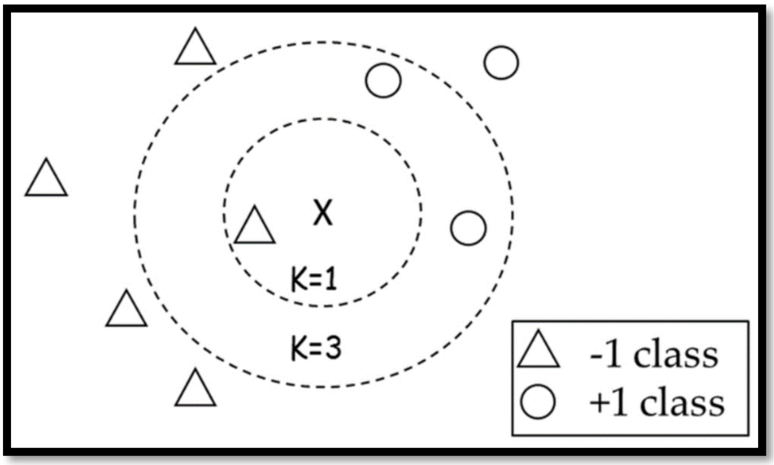


Figure 1.4: KNN Classification

There are several similarity-based approaches and the most prominently used are the Euclidean, Minkowski and Manhattan distances.

$$Distances : Euclidean = \sqrt{\Sigma(x-y)^2} \; ; Minkowski = [\Sigma\,[|x-y|]^n]^{\frac{1}{n}} \; ; \quad Manhattan = \Sigma\,|x-y|$$

B2.1 Facial expressions database & pre-processing techniques

The facial expression database used in this section has 40 images which is extracted from benchmark JAFFE (The Japanese Female Facial Expression) database which originally has 213 images of 10 female models and seven different expressions. [9].

Of the 40 images, 20 are used for training and other 20 are used for testing. The training images belong to five different female models containing four different emotions (Happiness, Surprise, Sad and Anger). The testing images belong to eight different female models containing four different emotions (Happiness, Surprise, Sad and Anger).

For the given images of the dataset, the preprocessing techniques are applied to first extract the human face from the image and then the mouth/eyes/eyebrows are extracted from the face image. Once the facial features are extracted, its corresponding image features are extracted using a feature extraction technique to train with a neural network or a classifier.

B2.2 Analysis and comparison of emotion recognition techniques

Emotion recognition from facial expressions is a popular and efficient way to determine human emotions. Underlying muscle movements in the human face caused by emotions express variations in facial features which when captured by facial expression identification methods can decode the emotion expressed by the humans.

Figure 2.1 shows the training dataset consisting for 20 images of five different female models containing four different emotions.

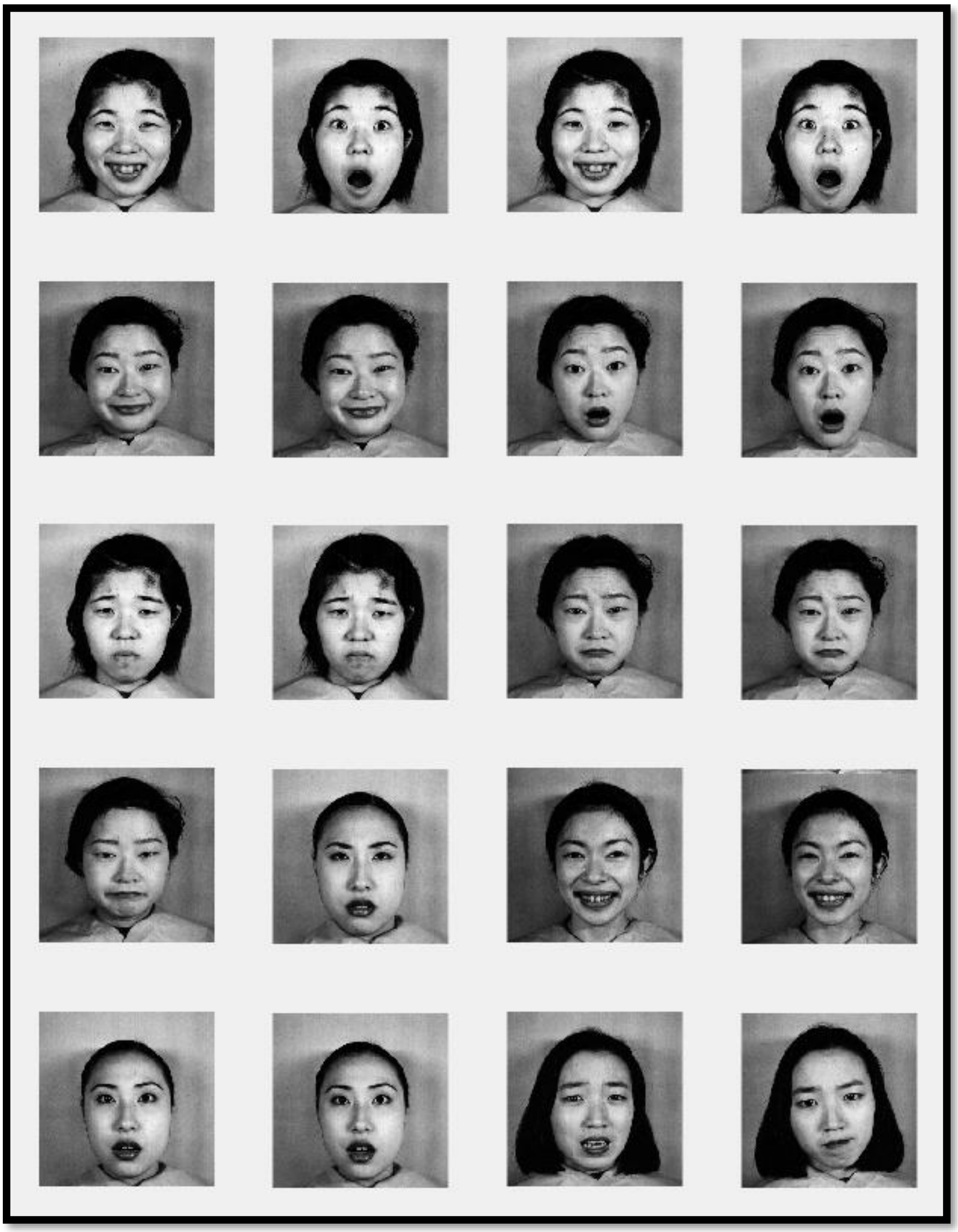


Figure 2.1: Training dataset for emotion recognition

Figure 2.2 shows the testing dataset for emotion recognition which consists of 20 images of eight different female models containing four different emotions.

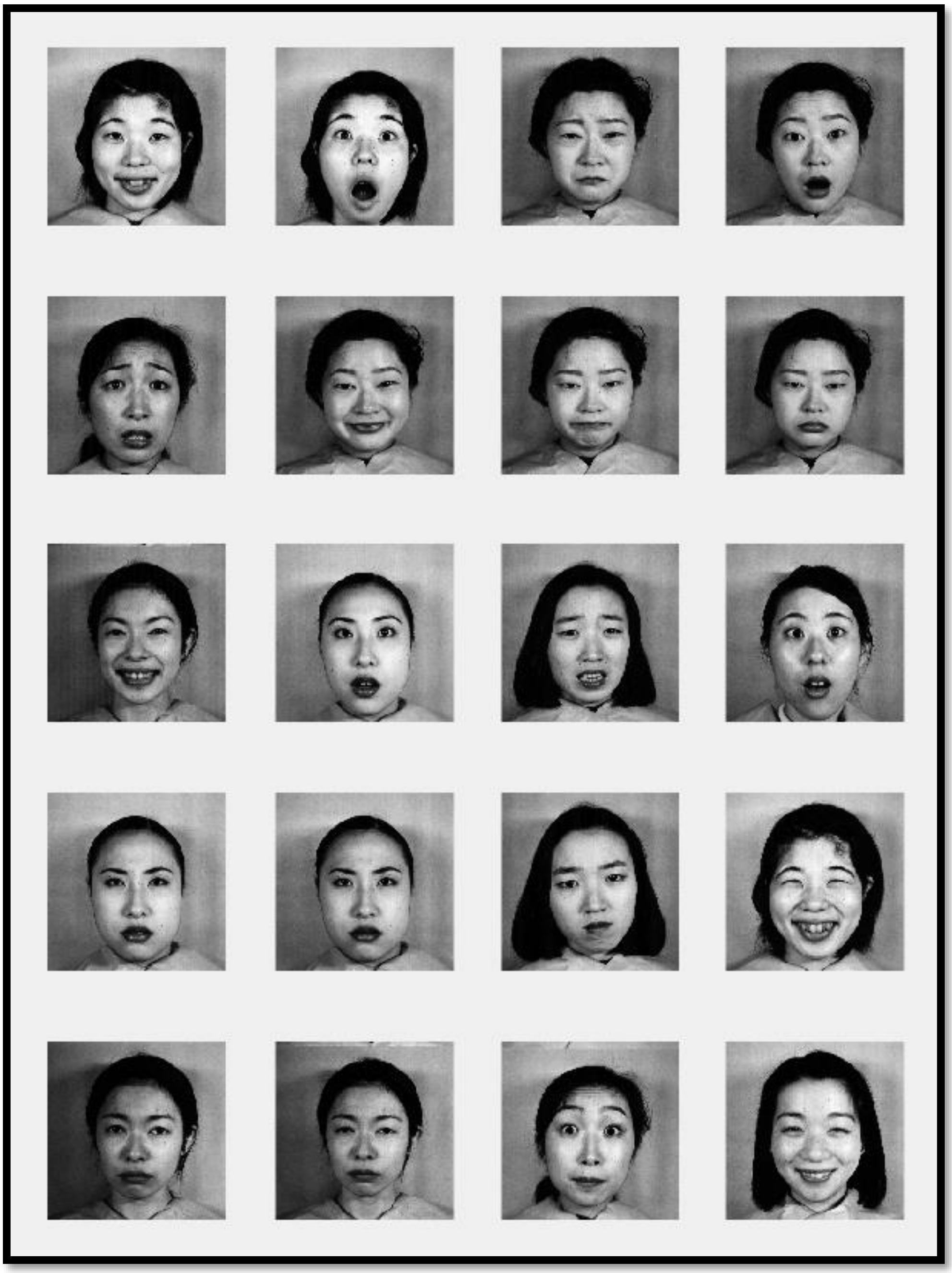


Figure 2.2: Testing dataset for emotion recognition

Various methods and algorithms can be used to determine emotions from facial expressions. One can broadly categorize the methodologies into neural network-based and classical methods-based. Neural networks tend to require more hidden layers which would require intense training through high computing power that becomes suitable to train huge training datasets one time and then later classify emotions for a variety of datasets.

For application-specific or dataset-specific emotion recognition systems it has been observed that classical methods not only require less computational power but are more efficient and reliable.

Usually in classical methods, at a high level there would be feature extraction techniques that extract the features and also a classifier which is trained with these features. Prior to extracting features, usually there is a preprocessing technique involved which will extract human facial features such as eyes, eyebrows, nose, mouth, etc. so that one or a combination of these facial features are used for further processing using feature extraction.

Feature extraction techniques popular in extracting features from facial features are fisher weight maps, principle component analysis (PCA), histogram of gradients (HOG), Wavelets, linear discriminant analysis (LDA), stochastic neighbor embedding, and local features bidirectional.

The feature extraction techniques mentioned are used in a combination with classifiers such as Naïve Bayes, support vector machines (SVM), K-nearest neighborhood (KNN), etc.

Table 2.1 compares emotion recognition techniques and their corresponding accuracies in recognizing emotions for the given dataset. The references contain details for each of these methods and cover the pitfalls of these techniques. The novel method used in this section delivers emotion recognition accuracy of 95% over the other specified methods.

Reference	Method	Accuracy
Shinohara [10]	HLAC + Fisher weight maps	69.4
Lyons [11]	Wavelet + PCA + LDA	75
Huang, M.W. [12]	GPLVM + SVM	65.24
Mingwei Huang [13]	SNE + SVM	73
Bin Hua [14]	bidirectional 2DPCA	92.52
My Method	COD + HOG + KNN Classifier	95

Table 2.1: Comparison of emotion recognition techniques

B2.3 Facial emotion recognition algorithm

The emotion recognition algorithm that was modeled in this section is non-neural network approach-based or, in other words, a classical approach-based algorithm which is aimed to be efficient for application-specific or dataset-specific requirements.

In this algorithm, the 20-image training dataset as shown in Figure 2.1 is used for training the classifier. Preprocessing is done on these training images and the human face is extracted or identified first.

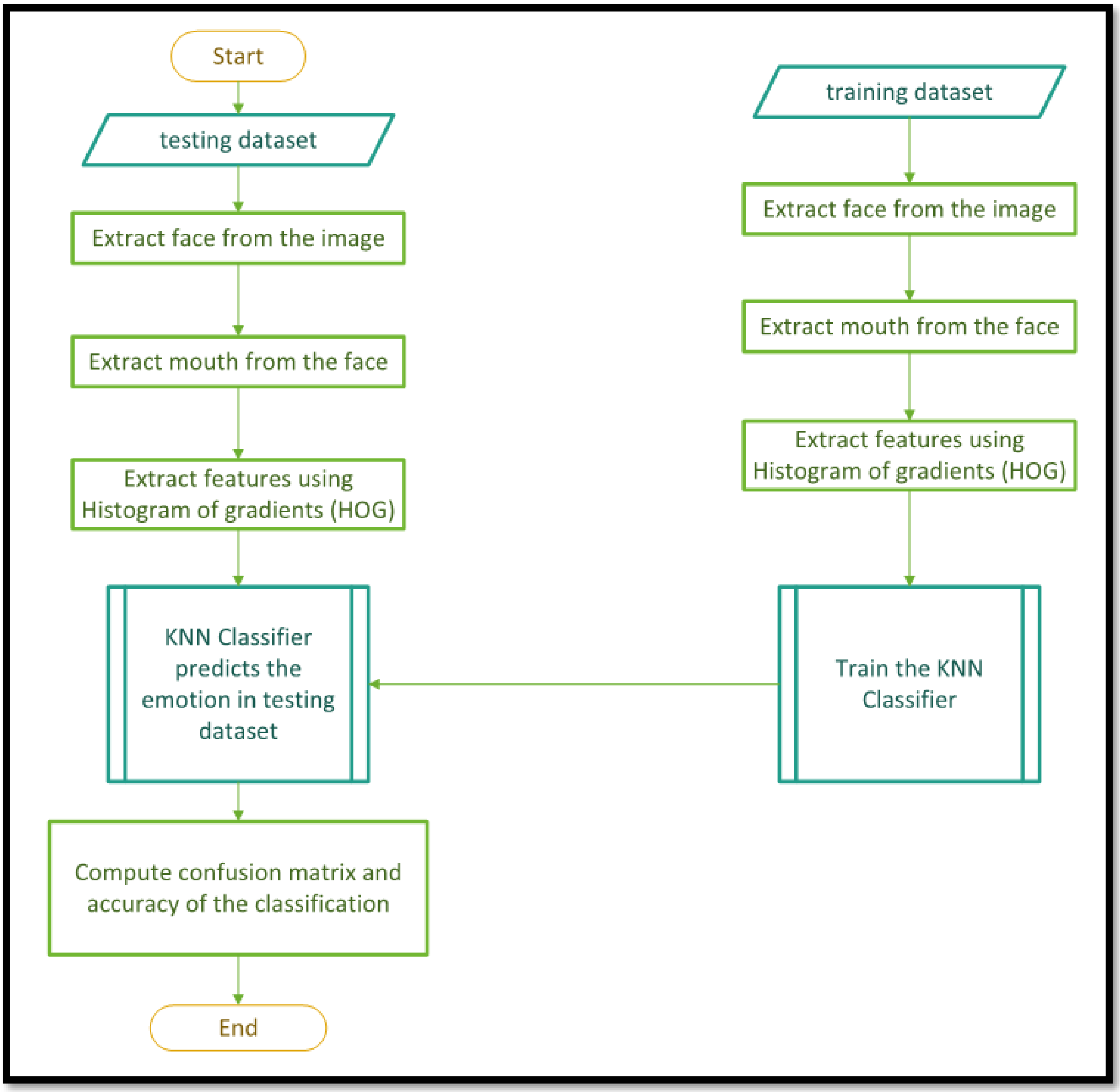


Figure 2.3: Flowchart of emotion detection algorithm

Figure 2.3 details the flowchart of the emotion recognition algorithm based on facial expressions. Once the human face is identified, one of the facial features (in this case, mouth) is extracted or identified. Since facial muscle movements reflect emotions, facial feature extraction is crucial in identifying the emotions.

Features of the identified mouth are extracted using HOG and these features, along with the emotion labels, are used to train the KNN algorithm. Hence, this is a supervised algorithm where the emotion labels are explicitly mentioned during the training. This procedure of training is repeated for all the images in the training image dataset. Figure 2.4 shows a high-level process of facial expression recognition.

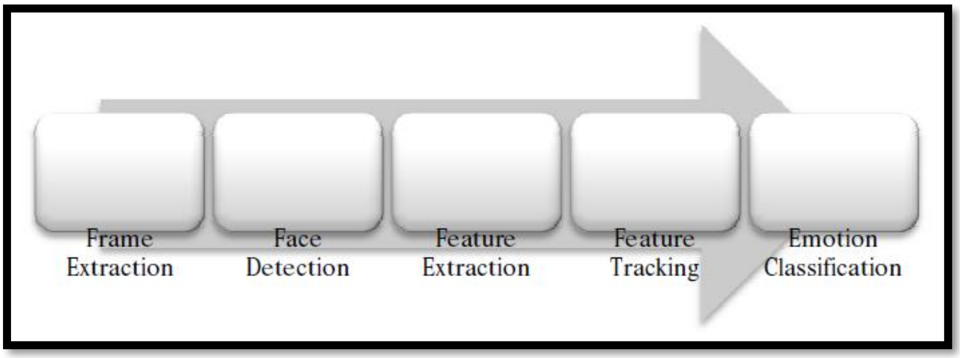


Figure 2.4: Facial expression recognition process

The trained KNN classifier is then used to test the feature vectors obtained from the HOG and prior preprocessing of testing images. The distance technique use in KNN classifier is Euclidean distance. The output of the KNN classifier is predicted emotion of the given test image which is then compared with expected test results to compute the accuracy of the algorithm in determining the emotions.

This algorithm uses facial expressions occurring in the mouth area of the face and identifies the emotions. There are other facial features which could play significant roles such as eyes and eyebrows. However, for this specific database the mouth expressions gave large variation in emotions. The use of HOG complements mouth expressions by forming feature vectors with large variance. This algorithm is well suited for the JAFFE dataset as it gives about 95% accuracy and takes under 15 seconds computation time.

B2.4 MATLAB model for emotion recognition

MATLAB implementation code of emotion recognition through facial expression is mentioned in the Appendix. The 20 training images are sent through cascade object detector (COD) which detects the face. The images are then cropped to just face level and then the facial feature “mouth” is extracted. These mouth-extracted images are then sent to HOG. Figure 2.5 shows the mouth extracts and their corresponding HOG vector orientations.

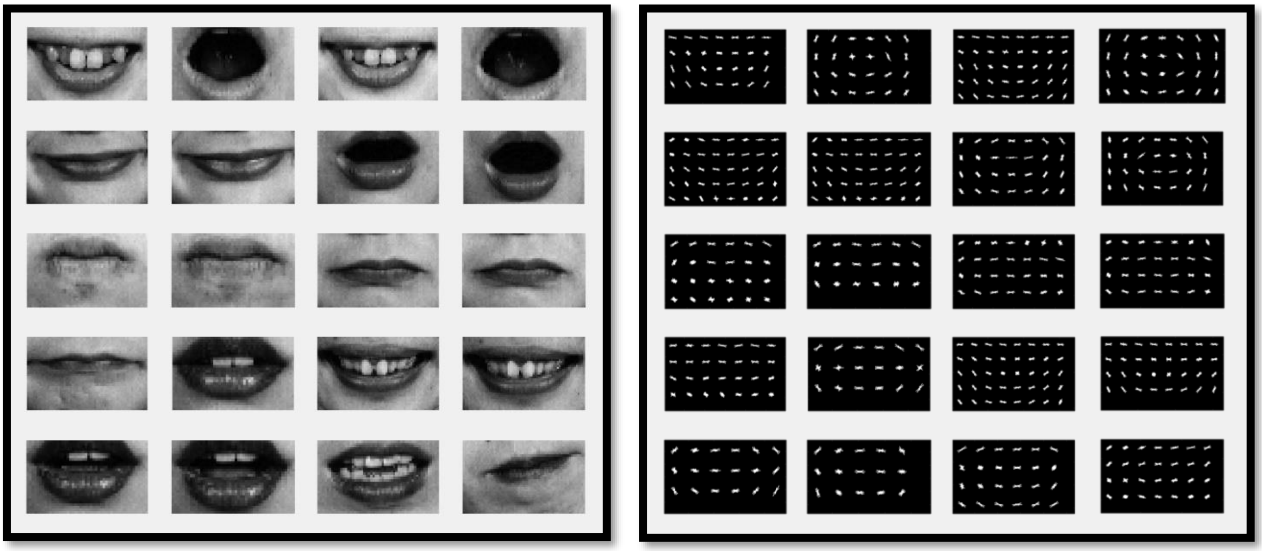


Figure 2.5: Mouth extract and HOG output for training images

HOG produces a higher dimensional feature vector which is trimmed to 120 dimensions to reduce computational complexity. These trimmed vectors are used to train the KNN classifier with appropriate labels.

Of the 20 training images, there are a mix of four different emotions (Happy, Surprise, Sad and Angry) from five different female models. Since there are a variety of different mouth expression for the same emotion, mean of feature vectors corresponding to each emotion is taken so that the mean vectors per each emotion can act as a reference to determine the mean square error and other accuracy measures.

In the testing phase, 20 testing images (five per each emotion) are passed through the trained KNN classifier model to determine the emotion associated with each image. These testing images belong to eight different female models and the images are different from training images. Figure 2.6 shows the mouth extracts of the test images and their corresponding HOG vector orientations.

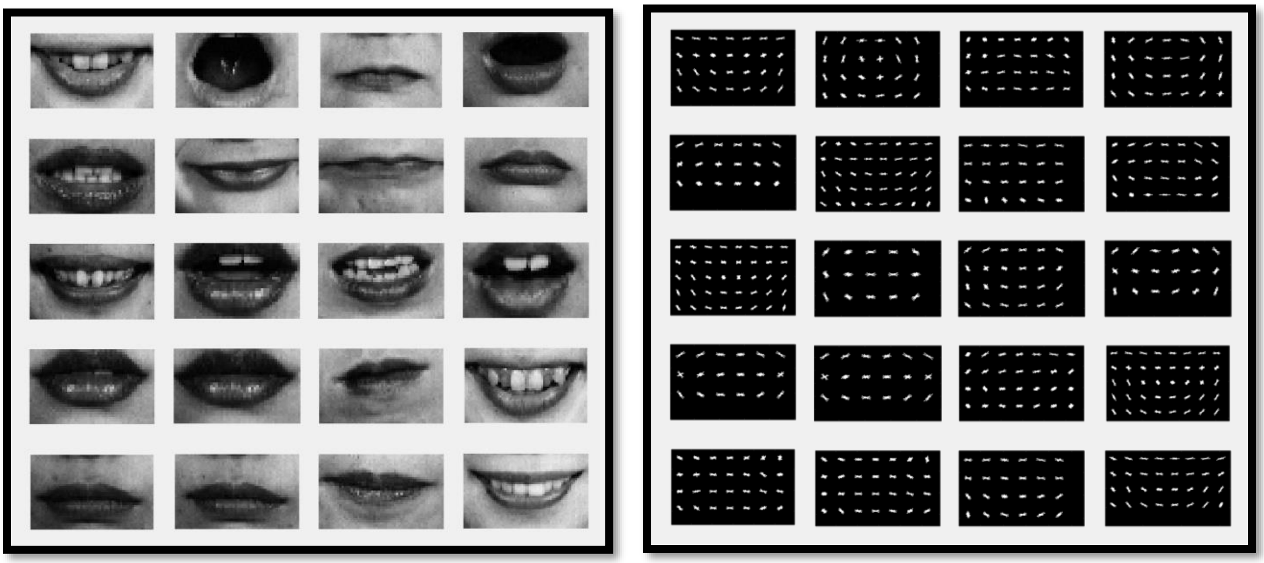


Figure 2.6: Mouth extract and HOG output for testing images

B2.5 Results of emotion recognition algorithm

This section mentions the results corresponding to the 20 testing images. The mean time of execution is 15 seconds and the overall accuracy of the classifier is 95%.

Confusion Matrix

Happy ->	**5**	**0**	**0**	**0**
Surprise ->	**0**	**4**	**0**	**1**
Sad ->	**0**	**0**	**5**	**0**
Anger ->	**0**	**0**	**0**	**5**

Mean Time of execution: 15 seconds

Accuracy: 95 %

Only one image which is supposed to be "Surprise" emotion is misclassified as anger as the mouth features, unlike all other surprise images, was not wide open and instead the mouth appeared closed. This misclassification is corresponding to the 19^{th} test image and Figure 2.7 illustrates the peak signal to noise ratio (PSNR) which specifically is very low for 19^{th} image on horizontal axis. Similarly, Figure 2.8 illustrates the mean square error (MSE) which again has a high error representation of 19^{th} test image which was misclassified.

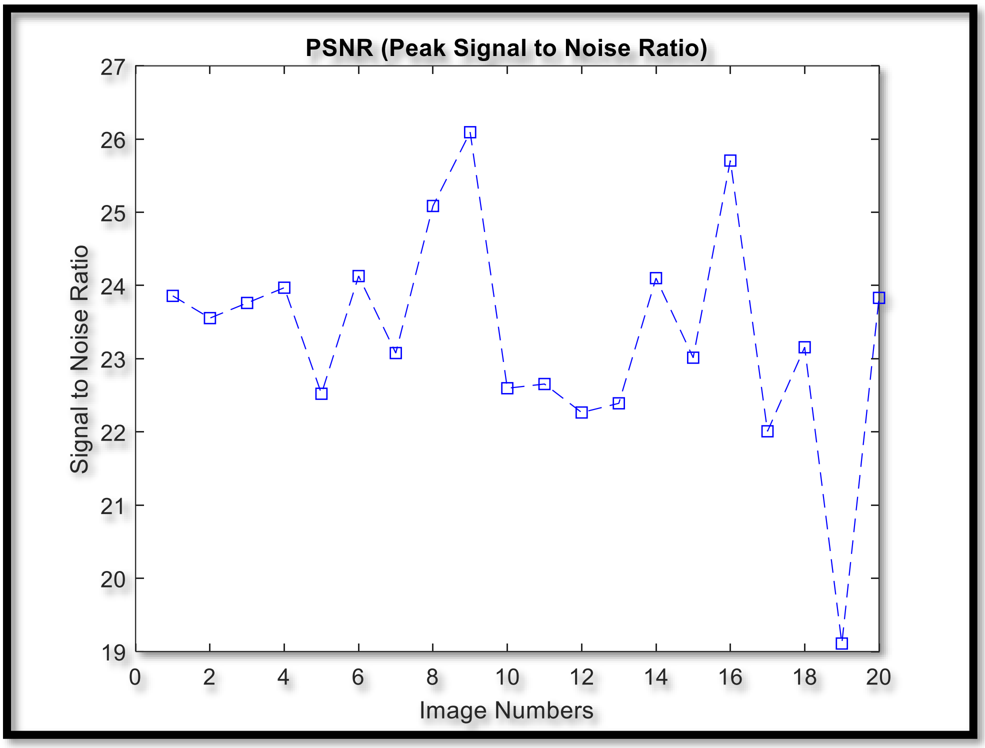


Figure 2.7: Peak signal to noise ratio

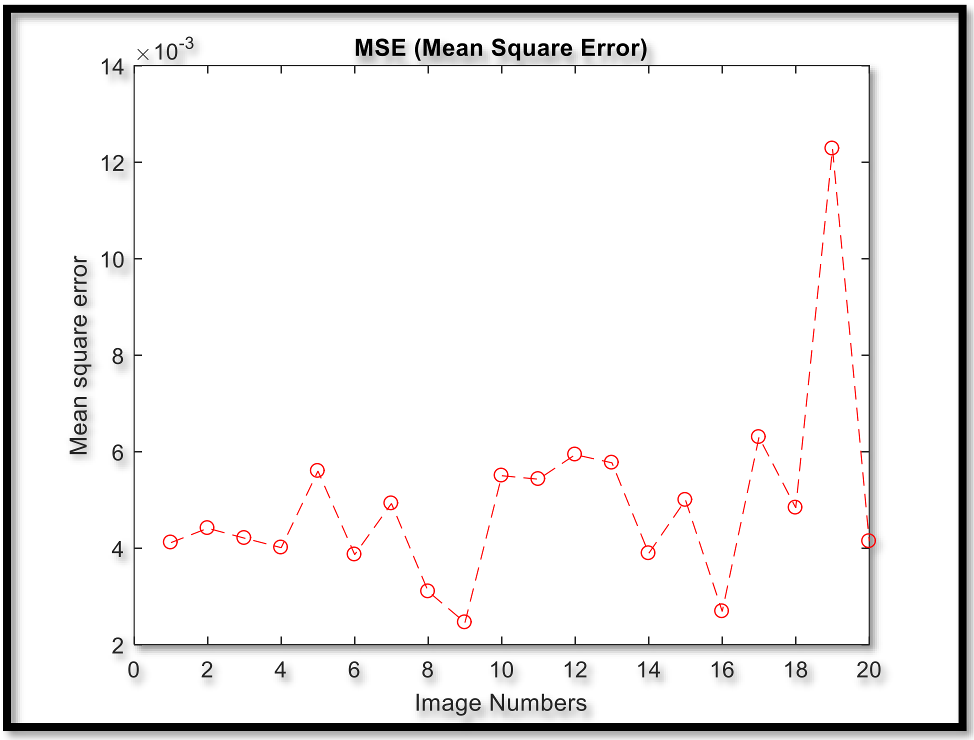


Figure 2.8: Mean square error

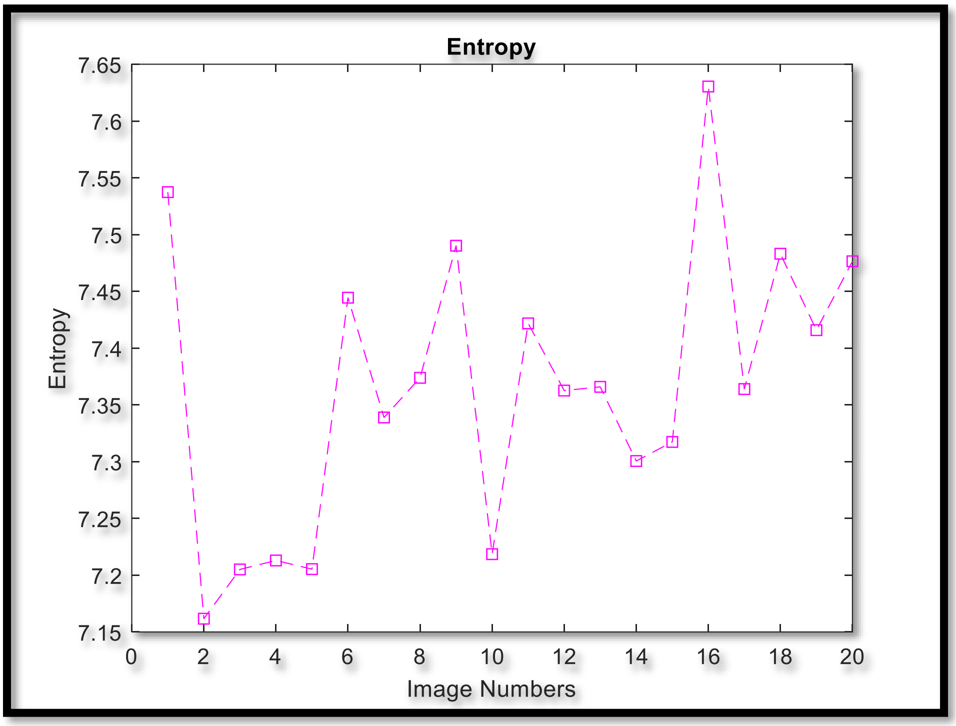


Figure 2.9: Entropy of testing images

Figure 2.9 illustrates the entropy of the given 20 test images and doesn't specifically highlight cause of misclassification. As mentioned, the MSE and PSNR clearly shows the 19th image with high error and low S/N ratio which is why that image is misclassified. The mouth features alone at times is not sufficient to provide accuracy; it is best to use multiple facial features as a combination.

References

“[1] A. Mehrabian, "Communication without words", Psychol. Today, (1986).”

“[2] U. Sderstrm and H. Li. Emotion recognition and estimation from tracked lip features.”

“[3] P. Ekman and W. Friesen, "Pictures of Facial Affect. Consulting Psychologist", (1976).”

“[4] Shbib, Reda & Zhou, Shikun. (2015). Facial Expression Analysis using Active Shape Model. International Journal of Signal Processing, Image Processing and Pattern Recognition.8.9-22. 10.14257/ijsip.2015.8.1.02.”

“[5] Widanagamaachchi WN. Facial emotion recognition with a neural network approach. University of Colombo; 2009.”

“[6] Busso, Carlos & Deng, Zhigang & Yildirim, Serdar & Bulut, Murtaza & Min Lee, Chul & Kazemzadeh, Abe & Lee, Sungbok & Neumann, Ulrich & Narayanan, Shrikanth. (2004). Analysis of emotion recognition using facial expressions, speech and multimodal information. ICMI'04 - Sixth International Conference on Multimodal Interfaces. 205-211. 10.1145/1027933.1027968. “

“[7] De Silva, L. C., Miyasato, T., and Nakatsu, R. Facial Emotion Recognition Using Multimodal Information. In Proc. IEEE Int. Conf. on Information, Communications and Signal Processing (ICICS'97), Singapore, pp. 397-401, Sept.1997.”

“[8] Goren D, Wilson HR. Quantifying facial expression recognition across viewing conditions. Vision Res. 2006 Apr;46(8-9):1253-62. Epub 2005 Dec 20. PubMed PMID:16364393.”

“[9] Michael J. Lyons, Shigeru Akemastu, Miyuki Kamachi, Jiro Gyoba. Coding Facial Expressions with Gabor Wavelets, 3rd IEEE International Conference on Automatic Face and Gesture Recognition, pp. 200-205 (1998).”

“[10] Y.Shinohara and N.Otsu, “Facial Expression Recognition Using Fisher Weight Maps”,
IEEE Conf. on Automatic Face and Guesture Recognition, pp. 499-504, 2004”

“[11] M.Lyons, J.Budynek, and S.Akamastu, “Automatic Classification of Single Facial Images”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, pp. 1357-1362,1999.”

“[12] Huang, M.W.; Wang, Z.W.; Ying, Z.L., “A novel method of facial expression recognition based on GPLVM Plus SVM”, ICSP, pp. 916-919, 2010.”

“[13] Mingwei Huang; Zhen Wang; Zilu Ying, “Facial expression recognition using Stochastic Neighbor Embedding and SVMs”, ICSSE, pp. 671-674, 2011.”

“[14] Bin Hua; Ting Liu, “Facial Expression Recognition Based on Local Feature Bidirectional 2DPCA”, ITCS, vol. 1, pp. 301–304, 2009.”

Appendix - MATLAB code for emotion recognition from facial expressions

```
clc;
clear all;
close all;

tic;
train_data = dir('C:\Users\dell\Documents\train1\*.tiff');
trainl=length(train_data);
trainf=[];
v1=[];

% Training Images loop
for trainloop = 1:trainl
   file_train = strcat('C:\Users\dell\Documents\train1\',train_data(trainloop).name);
   im_train= (imread(file_train));
   facedet = vision.CascadeObjectDetector(); % face detection in a given image
   bbox=step(facedet,im_train);
   face=imcrop(im_train,bbox);
   Mouthdet = vision.CascadeObjectDetector('Mouth'); % mouth detection in a given face image
   BB=step(Mouthdet,face);
   for i = 1:size(BB,1)
      a=BB(i,:);
      z=a(2);
      if z > 97
         b=face(a(2):a(2)+a(4),a(1):a(1)+a(3));
         figure(1);
         subplot (5,4,trainloop);
         imshow(file_train);
         figure(2);
         subplot (5,4,trainloop);
         imshow(b);
         [featureVector1,hogVisualization1] = extractHOGFeatures(b); %Extracting featues using HOG
         figure(3);
         subplot (5,4,trainloop);
         plot(hogVisualization1);
         Train_coeff1=featureVector1';
         v1=[v1,Train_coeff1(1:120,1:1)];
         trainf=v1';

      end
   end
end

Happy = [trainf(1,:);trainf(3,:);trainf(5,:);trainf(6,:);trainf(15,:);trainf(16,:)];
Surprise = [trainf(2,:);trainf(4,:);trainf(7,:);trainf(8,:);trainf(17,:);trainf(18,:)];
Sad = [trainf(9,:);trainf(10,:);trainf(11,:);trainf(12,:);trainf(19,:)];
Anger = [trainf(13,:);trainf(14,:);trainf(20,:)];
aT1=mean(Happy);
aT2=mean(Surprise);
aT3=mean(Sad);
aT4=mean(Anger);
```

```
train_group=[1;2;1;2;1;1;2;2;3;3;3;3;4;4;1;1;2;2;3;4];

Model=fitcknn(trainf,train_group','Distance','euclidean'); % Training the KNN Classifier

test_data = dir('C:\Users\dell\Documents\test1\*.tiff');
testl=length(test_data);
testf=[];
test_group=[1;2;3;2;3;1;4;4;1;2;3;2;4;4;4;1;3;3;2;1];
k=1;
for testloop = 1:testl
   file_test = strcat('C:\Users\dell\Documents\test1\',test_data(testloop).name);
   im_test= imread(file_test);
   facedet1 = vision.CascadeObjectDetector(); % face detection in a given image
   bbox1=step(facedet1,im_test);
   face1=imcrop(im_test,bbox1);
   Mouthdet1 = vision.CascadeObjectDetector('Mouth');  % mouth detection in a given face image
   BB1=step(Mouthdet1,face1);
   y1=size(BB1,1);
   for j = 1:y1
      a1=BB1(j,:);
      z1=a1(2);
      if z1 > 100
         b1=face1(a1(2):a1(2)+a1(4),a1(1):a1(1)+a1(3));
         figure(4);
         subplot (5,4,testloop);
         imshow(file_test);
         figure(5);
         subplot (5,4,testloop);
         imshow(b1);
         [featureVector2,hogVisualization2] = extractHOGFeatures(b1); %Extracting featues using HOG
         figure(6);
         subplot (5,4,testloop);
         plot(hogVisualization2);
         Train_coeff2=featureVector2';
         tc=Train_coeff2(1:120,1:1);
         tc=tc';
         [label,score,cost] = predict(Model,tc);
         TestPredict(k)=round(label);
         if test_group(testloop) == 1
            ref_tc=aT1;
         elseif test_group(testloop) == 2
            ref_tc=aT2;
         elseif test_group(testloop) == 3
            ref_tc=aT3;
         elseif test_group(testloop) == 4
            ref_tc=aT4;
         end
         PSNR_cal(k)=psnr(tc,ref_tc);
         mse_cal(k)=immse(tc,ref_tc);
         entropy_cal(k)=entropy(b1);
         k=k+1;
```

```
      end
   end
end

% Accuracy, mean time of executions, PSNR, MSE and Entropy Calculations
cMatf = confusionmat(test_group,TestPredict) %compare test data with new features
acc = 100*(sum(diag(cMatf))/length(TestPredict));
fprintf('Accuracy: ');
fprintf('%d\n', acc);
execution_time=toc

X=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20];
figure(7);
plot(X,PSNR_cal,'--bs');
xlabel ('Image Numbers');
ylabel ('Signal to Noise Ratio');
title('PSNR (Peak Signal to Noise Ratio)');

figure(8);
plot(X,mse_cal,'--ro');
xlabel ('Image Numbers');
ylabel ('Mean square error');
title('MSE (Mean Square Error)');

figure(9);
plot(X,entropy_cal,'--ms');
xlabel ('Image Numbers');
ylabel ('Entropy');
title('Entropy');
```