CHAPTER-1

INTRODUCTION

1.1 Definition

Human beings naturally and intuitively use facial expression as an important and powerful modality to communicate their emotions and to interact socially. Major component of human communication are facial expressions which constitute around 55 percent of total communicated message. There has been a global rush for facial expression recognition over the last few years. A number of methods have been proposed but no single method which is both efficient in terms of memory and time complexity has yet been found. Facial behaviour recognition is an application of computer vision that uses technology like image processing, and artificial intelligence, and expert knowledge of psychology. From the application point of view, computer vision is concerned with the theory behind artificial systems that extract information from images.

In general, Automatic facial behaviour recognition is divided in three major phases face detection, feature extraction and classification of expression. Face detection is a process of identifying face area from the input image. Feature extraction refers to the identification of facial feature points such eyes and lip contour which are responsible for particular facial behaviour. And final step is the identification of that facial expression using information gathered in the second phase.

1.2 Features of Project

The salient features of this project include the multitude of emotions expressed by human beings which can be achieved by accurately detecting faces in an image obtained from near real-time feed i.e. from the webcam of end-user's local device. The 8 different emotions that are used to derive a feedback from a user are Happy, Sad, Anger, Surprise, Contempt, Fear, Disgust and Neutral.

Face expression detection is an attractive field of research nowadays because it has wide application areas such as Human Machine Intelligent Interaction, Smart-rooms, Advance Driver Assistance Systems, Intelligent Robotics, Monitoring and Surveillance, Gaming, Research on pain and depression, Health support appliances, Deception Detection etc., As it has a wide application range it is challenging task to make a robust system that works under different circumstances.

An application to detect the genre of the song with the help of facial expression of the end-user while he/she is listening to song can be used to determine real-time feedback. Similarly another application interprets the category of news by reading the facial expression of the user.

1.3 Motivation

Generally, giving feedback in writing becomes a burden for people. So, the people either try to avoid sharing their feedback or they treat it as formality, hence they provide a fake feedback. As a result, this contradicts the importance of taking response.

Therefore, in order to overcome the weakness of written feedback, this project is developed to rule out the disadvantages of conventional mode of feedback. Moreover, the shortcomings of giving response in written were solved by using webcam for real-time feedback.

In engineering colleges, feedbacks are taken regularly for teachers in which the response given by students is similar for almost all the teachers because no student bothers to carefully enter the response.

The advantage of real-time facial feedback is that the end-user will always provide a genuine feedback. Moreover, the burden to forcefully provide response is eliminated by developing an automated model of real-time feedback. As a result, the motivation to carry out this project under the esteemed supervisors is achieved.

1.1 The concept of feedback

CHAPTER-2

LITERATURE SURVEY

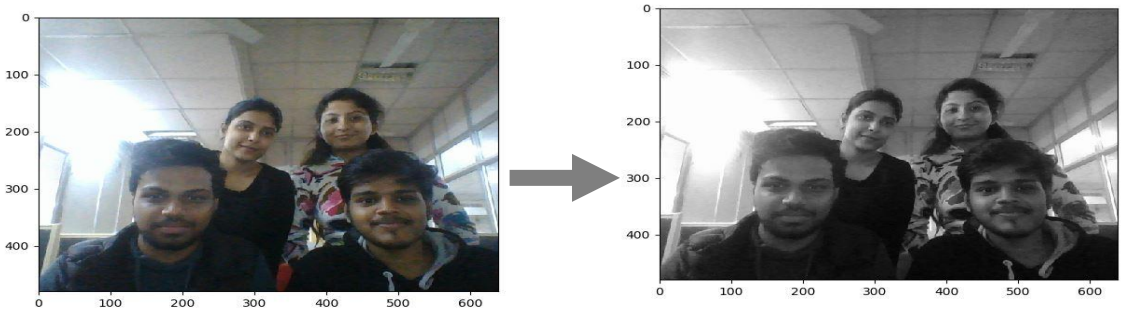
The process of facial expression recognition is classified into three stages:

1. Pre-processing of input images
2. Face detection methods
3. Facial expression recognition
4. Facial expression classification

This literature study gives an outline of the techniques and algorithms used to implement the above mentioned stages of facial expression recognition.

2.1 Pre-Processing of Input Images

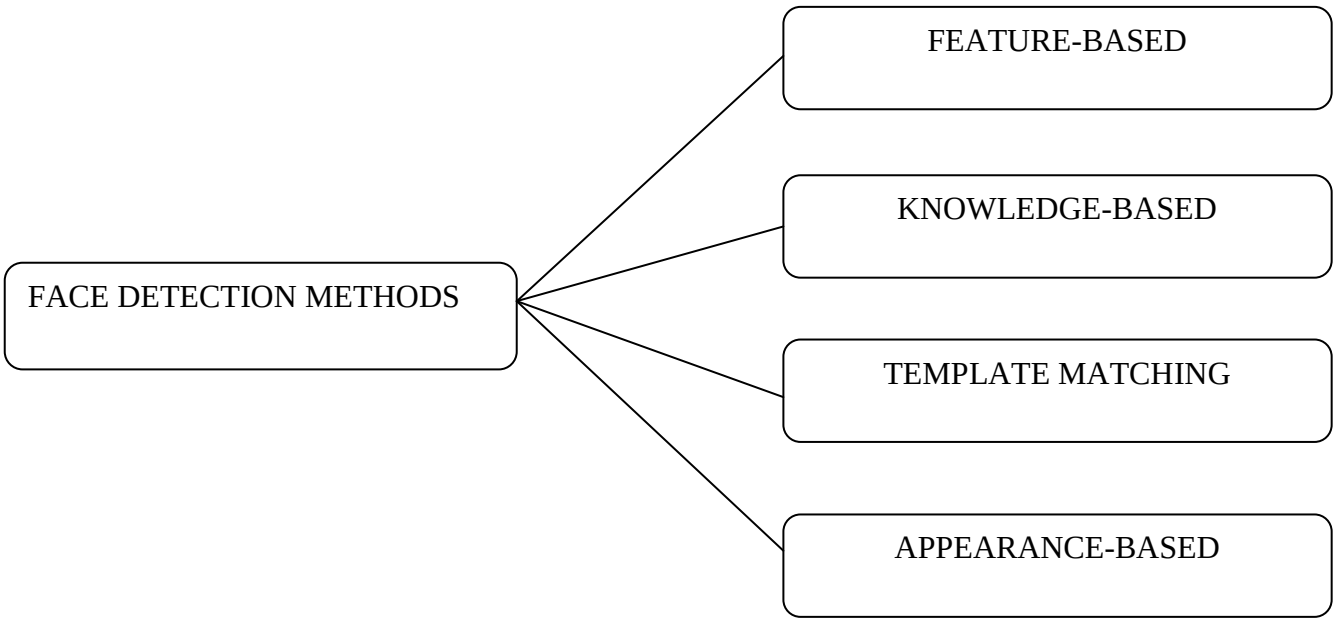
The aim of image pre-processing is to improve the image data by suppressing unwanted distortions and enhancements of some important image features. When we load an image using OpenCV, it loads it into BGR colour space by default. To show the colour image using matplotlib we have to convert it to RGB space. **cvtColor** is an OpenCV function to convert images to different colour spaces. It takes as input an image to transform, and a colour space code (like cv2.COLOR_BGR2RGB) and returns the processed image. In Short load our input image, convert it to greyscale mode and then display it with enhanced performance.



2.1 Image captured by webcam **2.2 RGB_TO_GRAY conversion**

2.2 Face Detection Methods

The primary aim of face detection algorithms is to determine whether there is any face in an image or not. These methods of Face Detection are divided into four categories, and the face detection algorithms could belong to two or more groups [5]. These categories are as follows-



2.3 Face detection method

Knowledge-Based

The knowledge-based method depends on the set of rules, and it is based on human knowledge to detect the faces. Ex- A face must have a nose, eyes, and mouth within certain distances and positions with each other. The big problem with these methods is the difficulty in building an appropriate set of rules. There could be many false positive if the rules were too general or too detailed. This approach alone is insufficient and unable to find many faces in multiple images.

Feature-Based

The feature-based method is to locate faces by extracting structural features of the face. It is first trained as a classifier and then used to differentiate between facial and non-facial regions. The idea is to overcome the limits of our instinctive knowledge of faces. This approach divided into several steps and even photos with many faces they report a success rate of 94%.

Template Matching

Template Matching method uses pre-defined or parameterised face templates to locate or detect the faces by the correlation between the templates and input images. Ex- a human face can be divided into eyes, face contour, nose, and mouth. Also, a face model can be built by edges just by using edge detection method. This approach is simple to implement, but it is inadequate for face detection. However, deformable templates have been proposed to deal with these problems.

Appearance-Based

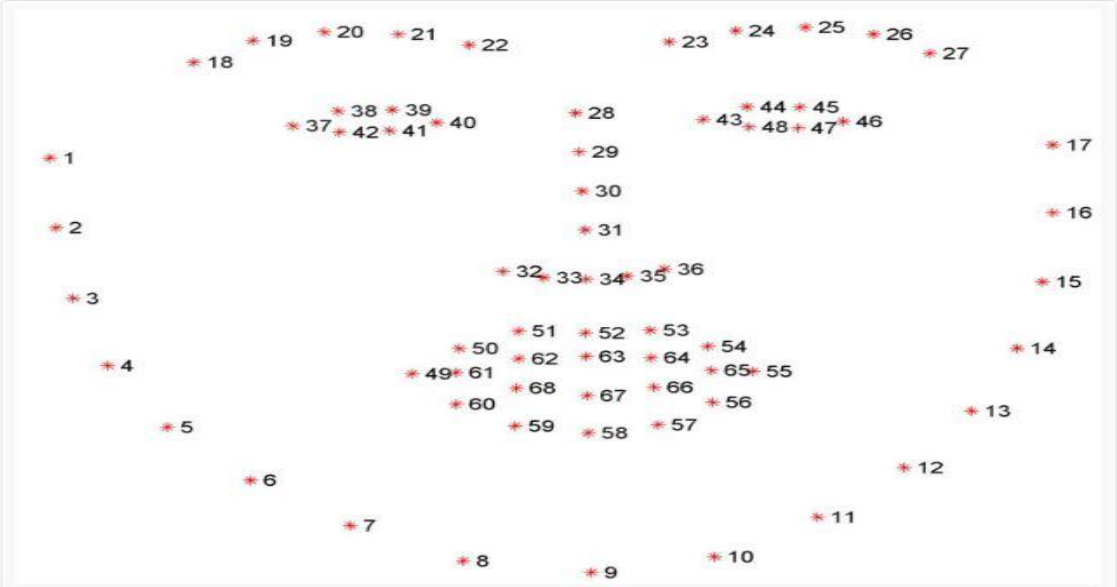
The appearance-based method depends on a set of delegate training face images to find out face models. The appearance-based approach is better than other ways of performance. In general appearance-based method rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face images. This method also used in feature extraction for face recognition.

2.3 Facial Expression Recognition

There are two types of face representations for face expression recognition i.e. holistic template based methods and geometric feature based method. In holistic method the whole face image is processed to obtain a template which is either a pixel image or a

feature vector. In the geometric feature based method, the shape and location of facial components are used to obtain feature vector.

For the facial recognition, the face is divided into number of local regions**.** The facial landmark detector detects these local regions which is implemented inside dlib produces 68 *(x,y)*-coordinates that map to *specific facial structures*. These 68 point mappings were obtained by training a shape predictor.[4]



2.4 Feature point extraction

Examining the image, we can see that facial regions can be accessed via simple Python indexing

- The mouth can be accessed through points [48, 68].
- The right eyebrow through points [17, 22].
- The left eyebrow through points [22, 27].
- The right eye using [36, 42].
- The left eye with [42, 48].
- The nose using [27, 35].
- And the jaw via [0, 17].

2.4 Facial Expression Classification

Eight categories of emotions are used in the classification stage: neutral, happy, anger, surprise, sadness, fear and disgust. Facial expressions can be classified by using some face parts such as eyes and mouth. Different classification methods are used to classify facial expressions in still images. A classifier also known as distance metric or matching criteria, is used to retrieve the similar face image from a large dataset. Some types of classifiers are Euclidean distance, histogram intersection distance, chi square distance, Support Vector Machine (SVM) etc.

Euclidean Distance measures the summation of difference among the paired values of the feature set. After taking the square root of the summation the closest distance measure is taken as the final result for that particular image. [1] The histogram intersection distance is a very simple method and is very useful in similarity measure where a large database is involved and quick replies are required. [1]

Support Vector Machines are linear classifiers that maximise the margin between the decision hyper plane and the examples in the training set. Osuna et al. first applied this classifier to face detection.

The classification algorithm is divided into three phases of processing: training, validation and testing [3] . In the training phase, the important characteristic properties of the image features are taken. The input images and the database images can be trained by using a classifier. They are then compared to find out the final expression. Using Support Vector Machine (SVM) classifier, images are trained by finding the Eigen faces which is detected by calculating the Eigen values. [2]

2.5 Facial expression classification

CHAPTER-3

PROBLEM STATEMENT & OBJECTIVES

3.1 Problem Statement

There has been a global rush for facial expression recognition over the last few years. A number of methods have been proposed but no single method which is both efficient in terms of memory and time complexity has yet been found. This gives rise to our problem statement i.e. to be able to infer emotions in real time using a live video feed and to be able to classify the emotions into one of the emotions such as anger, happy, surprise, disgust, sad, fear, contempt and neutral.

3.2 Major Objectives

The objectives of the project are as follows:-

- Describes how to train a model for a given dataset of 10,708 images downloaded from Cohn-Kanade database.
- Describes how to select the best tuning parameters for Support Vector Machine. The parameters are kernels, regularization parameter and gamma parameter.
- Describe how to implement various sampling techniques to improve accuracy. Best alternative out of Random Subsampling, Hold-Out, K-Fold, StratifiedK-Fold is then used to train the model.
- Describe how to ensemble different types of model trained to improve accuracy.
- Describe how to calculate the accuracy of the model by creating the confusion matrix.
- Describes the performed experiments and their results as well as the comparison of different classifiers.
- Using the model to develop application to detect genre of the song.
- Developing an application to detect category of news.

CHAPTER-4

PROJECT REQUIREMENT

4.1 Prerequisites

- **Python v3.5**

  Python is developed under an OSI-approved open source license, making it freely usable and distributable, even for commercial use. Python's license is administered by the python software foundation. Python can be easy to pick up whether you're first programmer or you're an experienced with other languages.

  Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects.Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

- **Sublime Text**

  Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface. It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

  The following is a list of features of Sublime Text

  - "Goto Anything," quick navigation to files, symbols, or lines
  - "Command palette" uses adaptive matching for quick keyboard invocation of arbitrary commands

- Simultaneous editing: simultaneously make the same interactive changes to multiple selected areas
- Python-based plugin API
- Project-specific preferences
- Extensive customizability via JSON settings files, including project-specific and platform-specific settings
- Cross-platform (Windows, mac OS, and Linux) and Supportive Plugins for cross-platform
- Compatible with many language grammars from Text Mate

- **Microsoft Visual Studio**

  Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.

  Microsoft Visual Studio 2018 is used in this project to install cmake which is basically an inter-dependency to use dlib library. dlib is a machine learning library which is a cross platform library. Since dlib is popularly known as C++ machine learning library but here we have used it in Python. Here in order to make use of dlib, cmake was necessary, therefore we had to install Microsoft Visual Studio 2018.

4.2 Dependencies

Modules used in the project are:

- **OpenCV(3.2.0)**

  OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. The library is cross-platform and free for use under the open-source BSD license.

- **Matplotlib 2.0**

  Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib
  can be used in Python scripts, the Python and IPython shells, the Jupiter notebook, web application servers, and four graphical user interface toolkits.

  Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

- **Imutils**

  A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges and much more easier with openCV and both Python 2.7 and Python 3.

- **Sklearn**

  It is simple and efficient tools for data mining and data analysis which is accessible to everybody and reusable in various contexts. It can be built on NumPy, SciPy, and matplotlib. It is Open source, commercially usable-BSD license. It contains inbuilt packages such as GridSearchCV, KFold, model_selection etc.

- **Numpy**

  NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

- **Pandas**

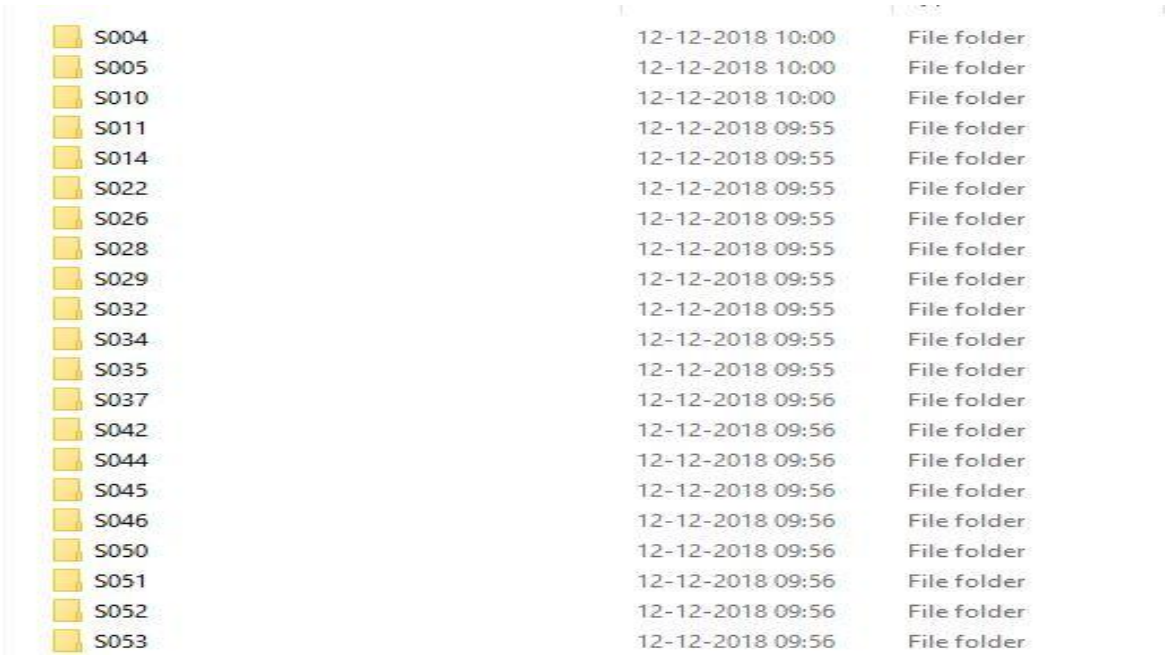
  Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

- **Pickle**

  The pickle module keeps track of the objects it has already serialized, so that later references to the same object won't be serialized again.

4.3 Dataset Used

Cohn-Kanade database:[6]This database consists of 10708 images and 8 expressions: neutral, sadness, surprise, happy, anger, fear, contempt and disgust. The images present in this database are mostly gray and it is a database consisting of American faces.123 subjects were used to create this database.Another work on the Cohn-Kanade database makes use of Gabor filtering for image processing and Support Vector Machine (SVM) for classification. A Gabor filter is particularly suitable for pattern recognition in images and is claimed to mimic the function of the human visual system [10].A big disadvantage of the approach however is that very precise pre-processing of the data is required, such that every image complies to a strict format before feeding it into the classifier.



4.1 Data set hierarchy

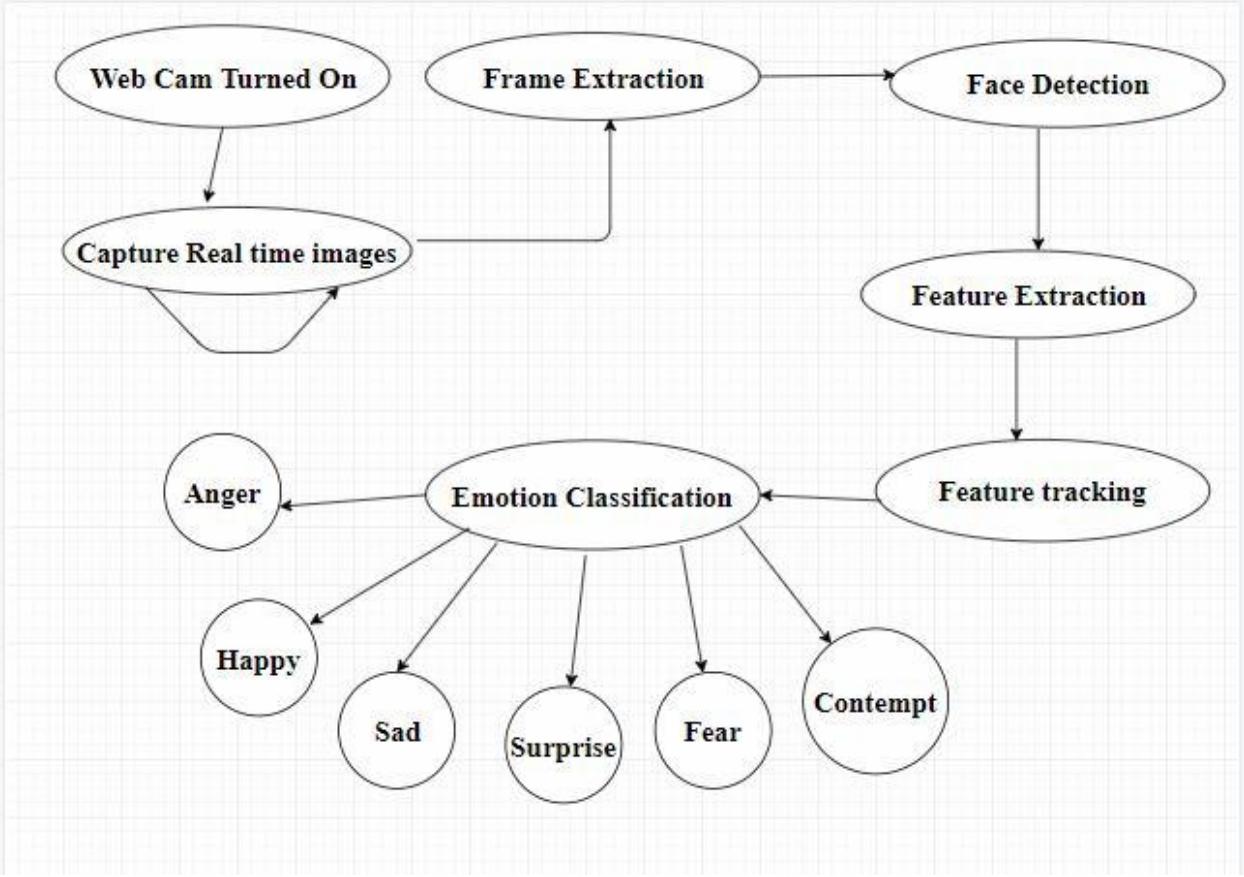
4.2 Training images

CHAPTER-5

IMPLEMENTATION

5.1 Architecture

The project is implemented with the help of the following workflow. Each step in the workflow from Webcam Turned-On to Emotion Classification has been given a detailed explanation with their respective code snippets. Code snippets are included in this project to better understand the nomenclature used in classifying the emotions. Along with the code snippets, screenshots of the output wherever necessary are also included to have an insight of practical knowledge.



5.1 Project workflow

5.2 Face Detection

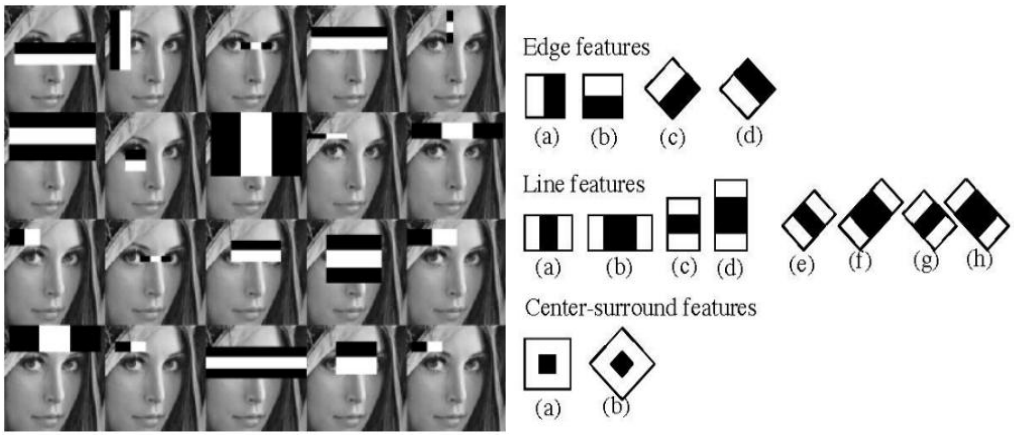
Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene.

An efficient algorithm for accurate detection and extraction of facial features has been proposed. The colour image is converted into a gray scale image. Then pre-processing is performed by applying contrast equalization.

5.2.1.1 Feature Based Methods

(a) Haar Classifier

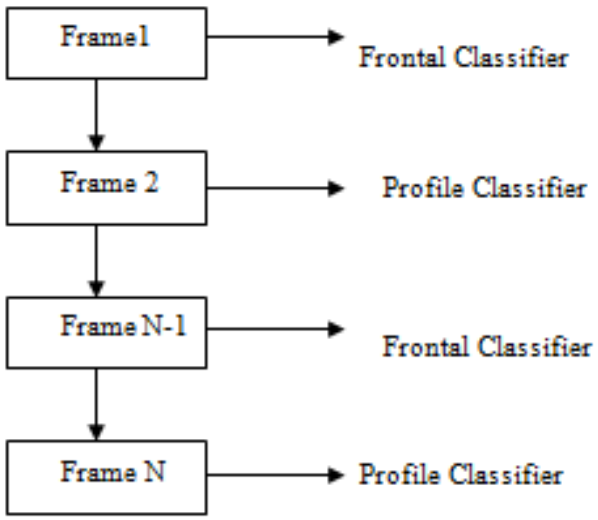
- The Haar Classifier is a machine learning based approach, an algorithm created by Paul Viola and Michael Jones; which are trained from many positive images (with faces) and negatives images (without faces).
- Each window is placed on the picture to calculate a single feature. This feature is a single value obtained by subtracting the sum of pixels under the white part of the window from the sum of the pixels under the black part of the window.
- Haar classifier calculates all features , most of them are irrelevant and to discard them we use adaboost and hence it improves classification accuracy



5.2 Haar-like features

Steps involved in Haar Classifier Face Detection

- Each frame is processed firstly through Haar classifiers trained for profile faces. To further improve frame rate and compensate for pose variation we propose to use interleaved Haar Classifiers. Interleaving is done between front and profile classifiers.
- Second to reduce false results produce, output of last step is passed through skin detection algorithm.
- Next step makes sure that the region selected is a face. If both profile and frontal faces are not detected and if face has been initialized then Optical Flow is applied to track those feature points.



5.3 Steps of haar classification

Disadvantages

- This feature based method takes longer training time due to which the execution speed of it is slow and has less performance due to these issues.
- It is not easy to implement. Therefore, the computation involved in this method is complex and takes an expert who has previous knowledge in this field to implement it. Hence, it is computationally complex.

- Another major drawback is that method's efficiency drastically decreases when tested against black faces i.e. susceptible to light invariant faces.
- Due to the above shortcomings, this method has limitations in difficult lightening conditions.
- This method is vulnerable to ambiguous requirements. Hence, it is less robust to occlusion.

Due to the above major drawbacks, Haar Classifier Feature Based Method was not used to develop the final model. Instead one of the models included in Appearance Based Method to detect faces is used to train our dataset to produce the final model.

(b) Local Binary Patterns

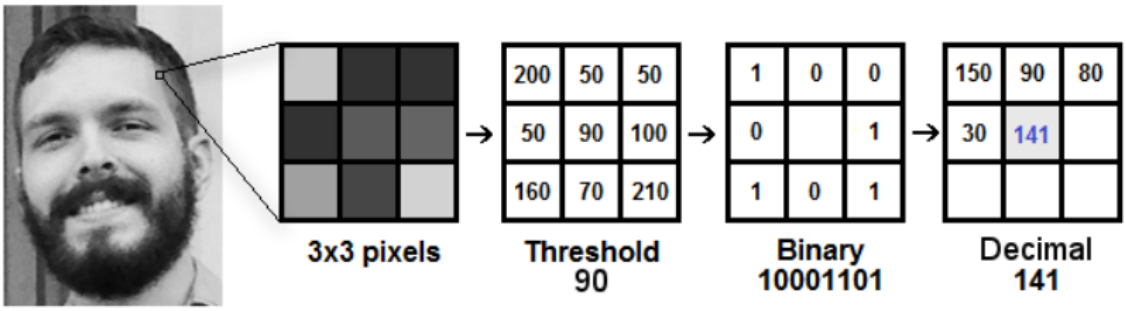
It is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighbourhood of each pixel and considers the result as a binary number.

As LBP is a visual descriptor it can also be used for face recognition tasks, as can be seen in the following step-by-step explanation.

Steps involved in Local Binary Pattern Face Detection

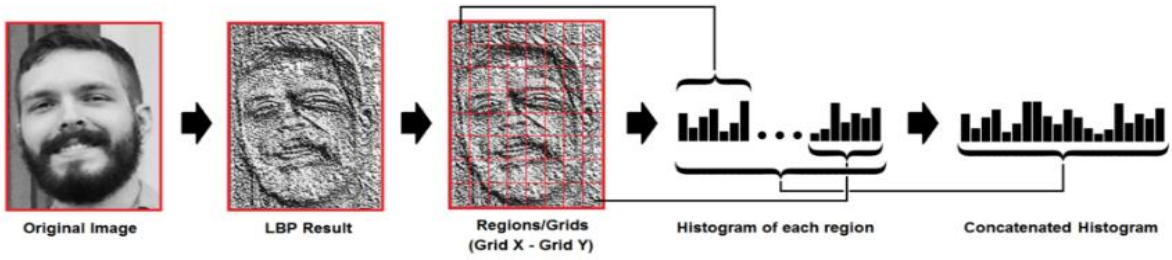
- **Training the Algorithm**: First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps.

- **Applying the LBP operation**: The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters **radius** and **neighbours**.



5.4 LBP operation

- **Extracting the Histograms**: Now, using the image generated in the last step, we can use the **Grid X** and **Grid Y** parameters to divide the image into multiple grids, as can be seen in the following image:



5.5 LBP histogram extraction

Based on the image above, we can extract the histogram of each region as follows:

- As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.
- Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

Disadvantages

- This method has less accuracy compared to Haar Classifier feature based method.
- Due to less accurate results, this method was also not used to provide us with final model.

5.2.1.2 Appearance Based Method

(a) Support Vector Machine

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models which are associated with machine learning algorithms that analyse the given data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

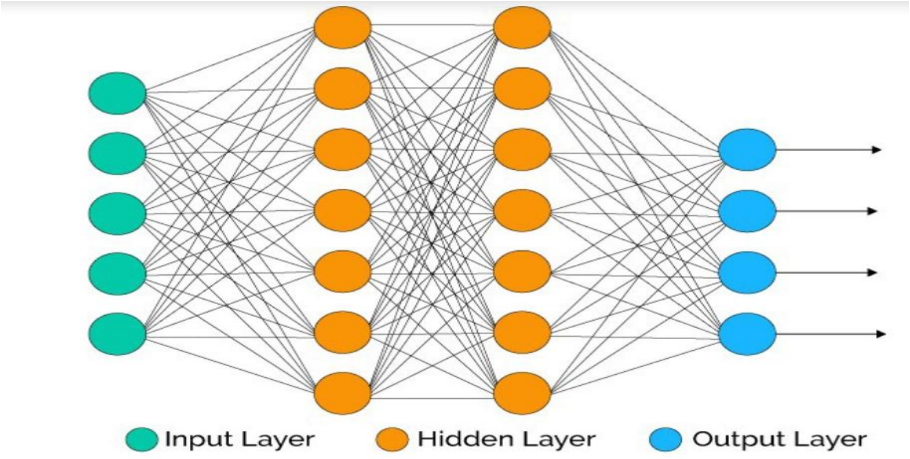
More detailed explanation is given in the later topics. This method ideally suits our type of dataset. The earlier drawbacks which were mentioned in previous methods are eliminated in SVM model.

(b) Neural Network

Neural networks are a set of algorithms, modelled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labelling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.

Neural networks help us cluster and classify. You can think of them as a clustering and classification layer on top of the data you store and manage. They help to group unlabeled data according to similarities among the example inputs, and they classify data when they have a labelled dataset to train on. (Neural networks can also extract features that are fed to other algorithms for clustering and classification; so you can think of deep neural networks as components of larger machine-learning applications involving algorithms for reinforcement learning, classification and regression.)

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems.



5.6 Neural networks

(c) Naïve Bayes

Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

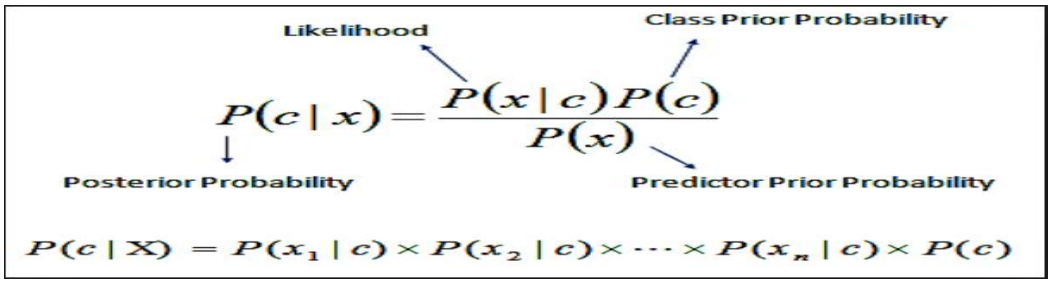
We assume that no pair of features are dependent. For example, the temperature being 'Hot' has nothing to do with the humidity or the outlook being 'Rainy' has no effect on the winds. Hence, the features are assumed to be independent.

Secondly, each feature is given the same weight(or importance). For example, knowing only temperature and humidity alone can't predict the outcome accurately. None of the attributes is irrelevant and assumed to be contributing equally to the outcome.
Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred.

- Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence.
- P(A) is the priori of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
- P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

Naïve Assumption

This algorithm, Naïve Bayes has an important assumption to the Bayes' theorem, which is, **independence** among the features.
Bayes' theorem is stated mathematically as the following equation:



5.7 Naïve bayes classification

5.3 Feature Extraction

After face has been detected, we narrow our work zone to detected region thereby reducing the
search area and improving frame rate. Now face is divided geometrically to find feature points
and further confine the search area.

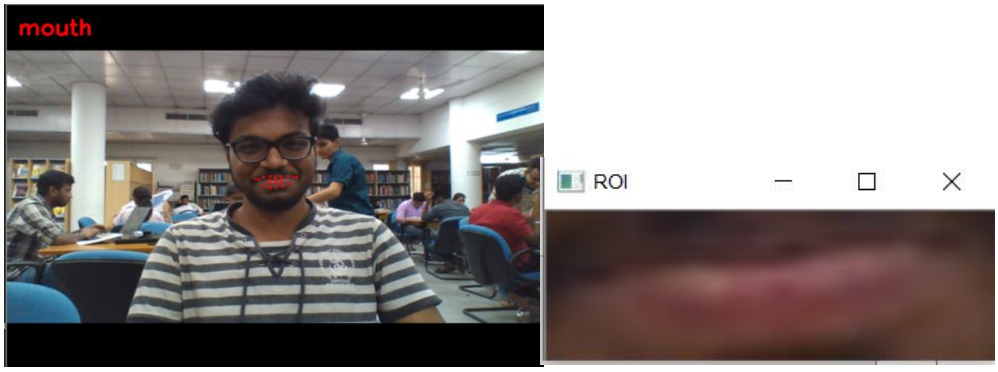
Then classifiers for eyes, nose, and mouth are used in the narrowed output region. The facial landmark detector implemented inside dlib produces 68 *(x, y)*-coordinates that map to specific facial structures. These 68 point mappings were obtained by training a shape predictor on the labelled Cohn-Kanade Dataset.

The following landmarks are detected in any image:-

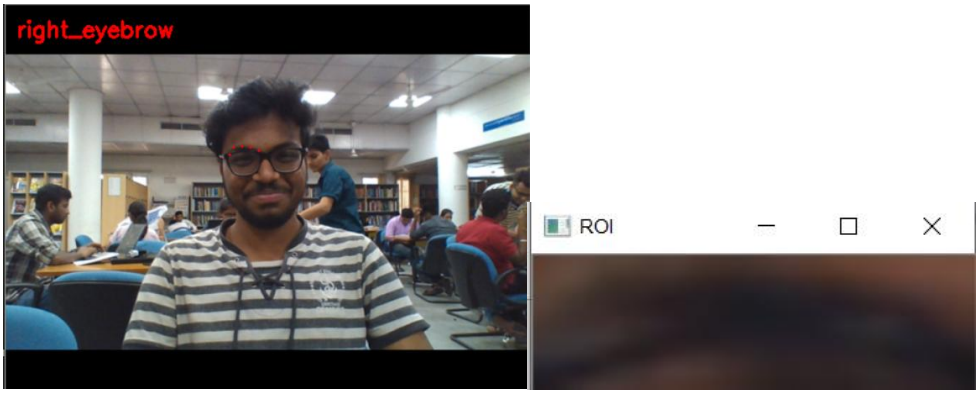
- Mouth
- Right eyebrow
- Left eyebrow
- Right eye
- Left eye
- Nose
- Jaw

After face detection, facial landmarks are detected are extracted separately which are shown as below:-

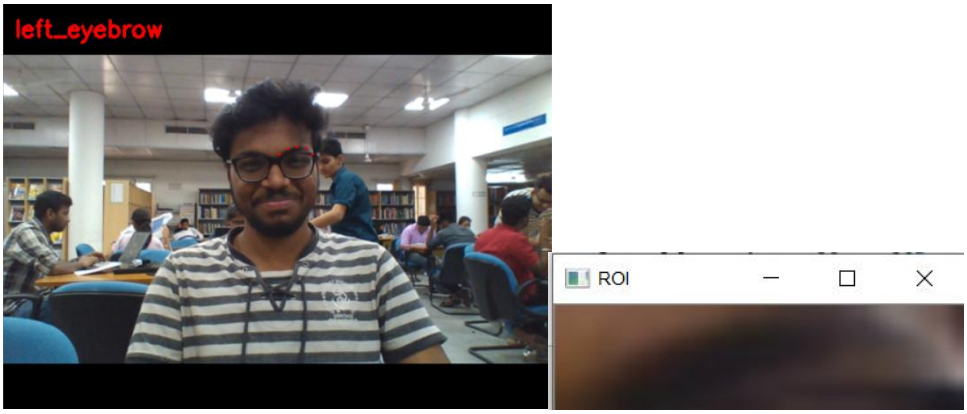
Mouth



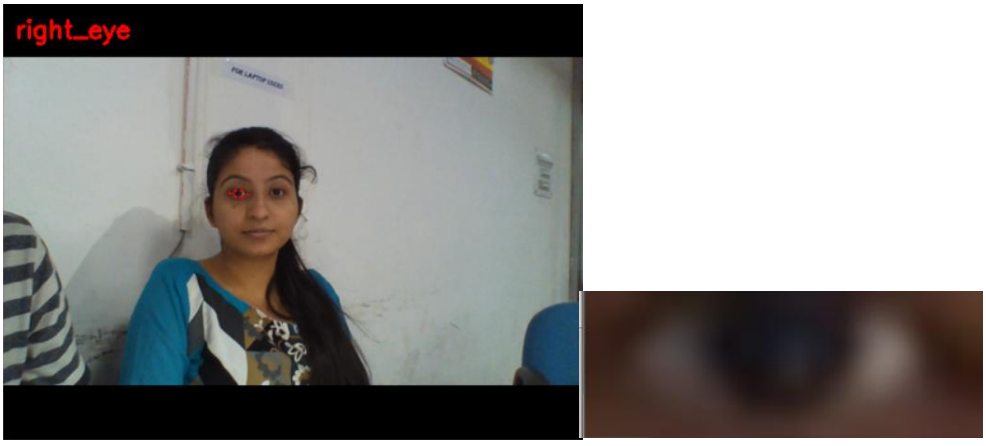
Right Eyebrow



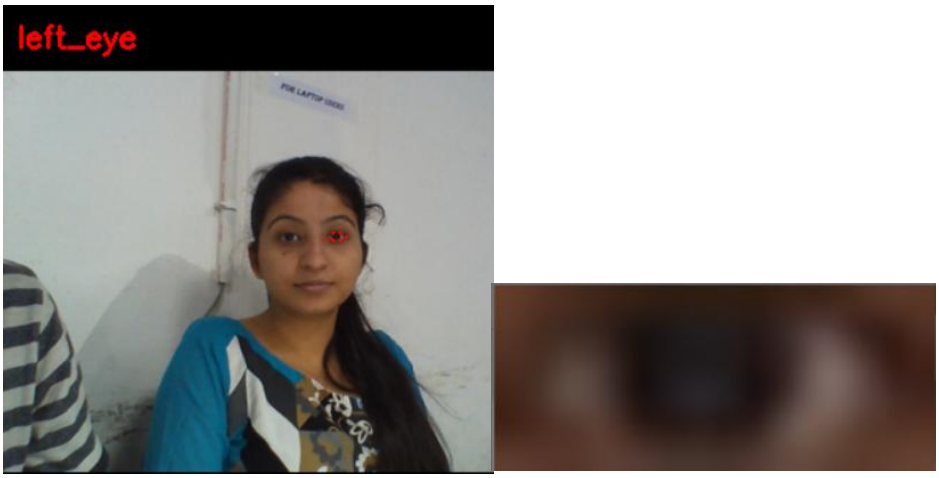
Left Eyebrow



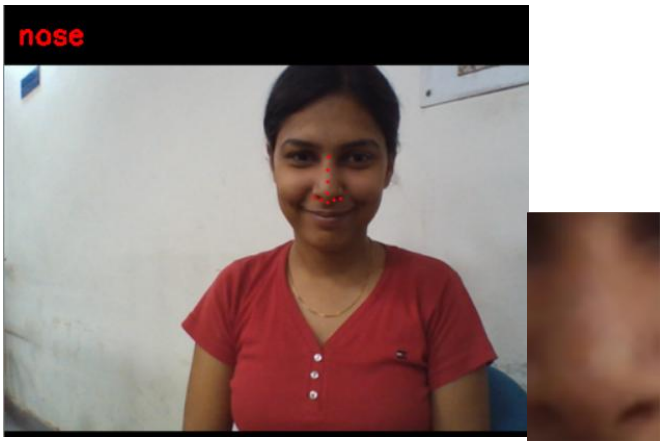
Right Eye



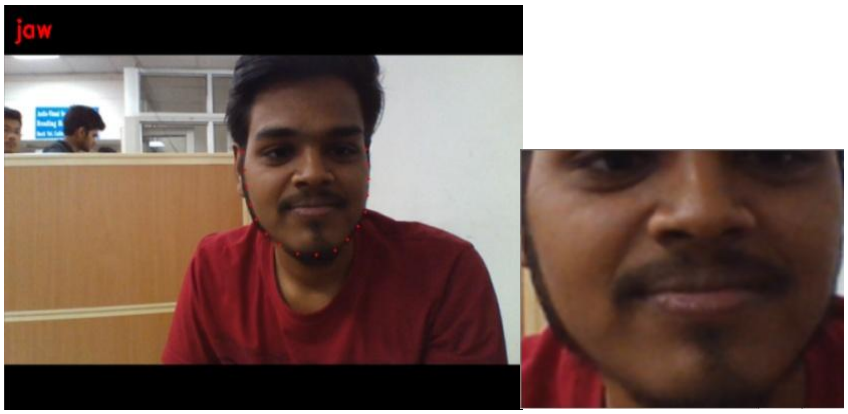
Left Eye



Nose



Jaw



5.8 Feature extraction

- **predictor()** function detects the individual landmarks on the grey coloured face which is stored in the variable ‘rect’.
- **visualize_facial_landmarks()** returns an image where all facial landmarks are visualized on a single image as depicted in the figure given below.

5.4 Classification Technique

In machine learning, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm that maps input data to a category.

5.9 Classes of classification

5.4.1.1 SVM Classification

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyper plane. In other words, given labelled training data the algorithm outputs an optimal hyper plane which categorizes new examples. In two dimensional space this hyper plane is a line dividing a plane in two parts where in each class lay in either side.

Displacements of 21 feature points is used as an input to SVM classifier first a model is created with help of libSVM [9] by training and then Evaluator is used to classify emotions based on the learned model created.

Following steps for using libSVM were used

• Transform data to the format of an SVM package

• Conduct simple scaling on the data

$$K(\mathbf{x}, \mathbf{y}) = e^{-\gamma\|\mathbf{x}-\mathbf{y}\|^2}$$)]

• Consider the RBF kernel

• Use cross validation to find the best parameter C and gamma

• Use the best parameter C and gamma to train the whole training set

• Test

This kernel nonlinearly maps samples into a higher dimensional space so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is nonlinear. Furthermore, the linear kernel is a special case of RBF.

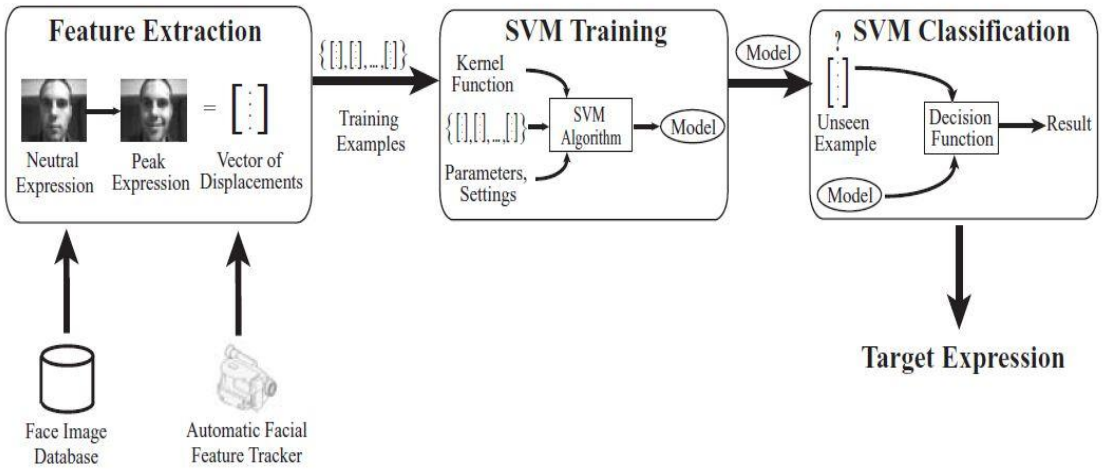
The second reason is the number of hyperparameters which influences the complexity of model selection. The polynomial kernel has more hyperparameters than the RBF kernel.

Most computational overhead resides in the training phase. However, due to the fact that the training set is interactively created by the user and hence limited in magnitude and that the

individual training examples are of constant and small size, overhead is low for typical training

run [8].

5.10 Hyper plane separation by SVM



5.11 Stages of automated expression recognition approach

5.4.1.2 Tuning Parameters

A tuning parameter is parameter used in statistics algorithm in order to control their behaviour. The algorithm creates normally for each value of the tuning parameter a different model.

Translating this into common sense, tuning is essentially selecting the best parameters for an algorithm to optimize its performance given a working environment such as hardware, specific workloads, etc. And tuning in machine learning is an automated process for doing this.

Kernel

The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra. This is where the kernel plays role.

For **linear kernel** the equation for prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows [7]:

f(x) = B(0) + sum(ai * (x,xi))

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

The **polynomial kernel** can be written as K(x,xi) = 1 + sum(x * xi)^d and **exponential** as K(x,xi) = exp(-gamma * sum((x—*xi²)).*

Polynomial and exponential kernels calculates separation line in higher dimension. This is called **kernel trick**

Regularization

The Regularization parameter (often termed as C parameter in python's sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example [7].

For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

The images below (same as image 1 and image 2 in section 2) are example of two different regularization parameter. Left one has some misclassification due to lower regularization value. Higher value leads to results like right one.

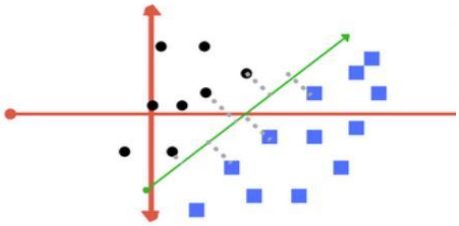
Low regularization value And High regularization value

5.12 Regularization parameters

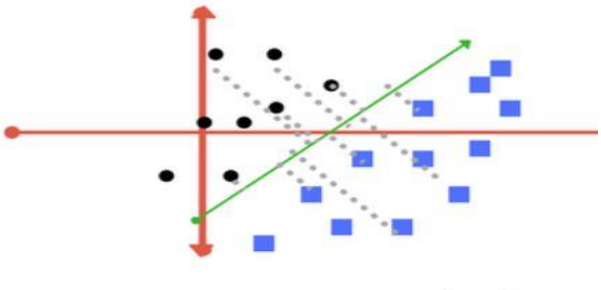
Gamma

The gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’. In other words, with low gamma, points far away from plausible separation line are considered in calculation for the separation line where as high gamma means the points close to plausible line are considered in calculation.

High Gamma



Low Gamma

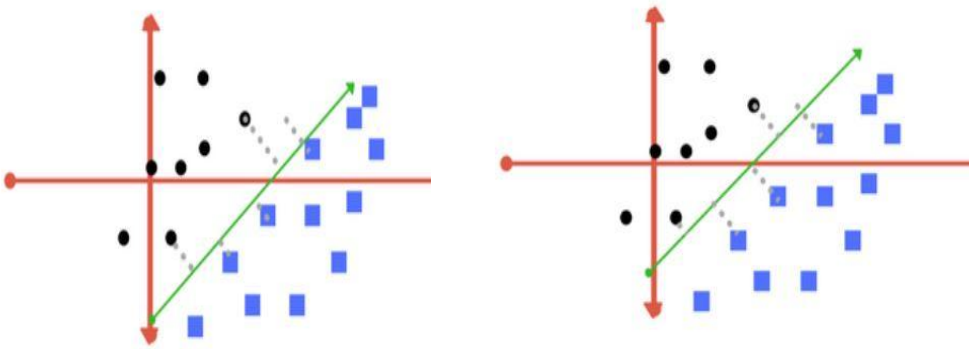


5.13 Gamma Parameter

Margin

And finally last but very important characteristic of SVM classifier. SVM to core tries to achieve a good margin. A margin is a separation of line to the closest class points.A good margin is one where this separation is larger for both the classes. Images below gives to visual example of good and bad margin. A good margin allows the points to be in their respective classes without crossing to other class.

The following images describes shows the bad and good margin (Equidistant from both the classes) respectively.

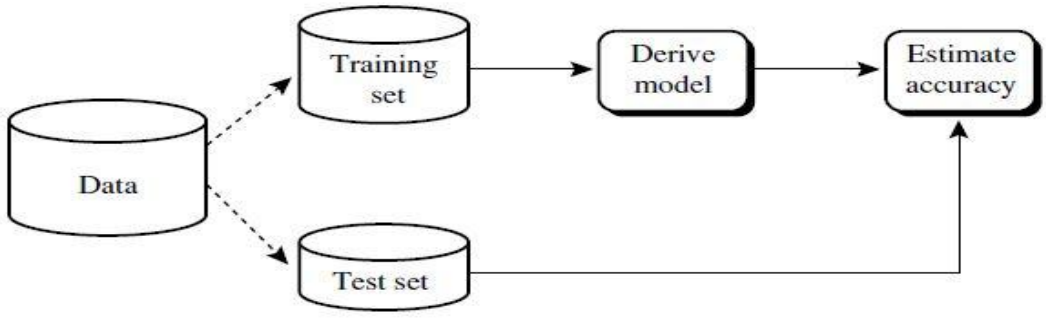


5.14 Marginal characteristics of SVM

```
# The possible values of tuning parameters of svm
tuned_parameters = [{'kernel': ['rbf'], 'gamma': [1e-2, 1e-3, 1e-4, 1e-5],
                     'C': [0.001, 0.1, 10, 25, 50, 100, 1000]},
                    {'kernel': ['sigmoid'], 'gamma': [1e-2, 1e-3, 1e-4, 1e-5],
                     'C': [0.001, 0.1, 10, 25, 50, 100, 1000]},
                    {'kernel': ['linear'], 'C': [0.001, 0.1, 10, 25, 50, 100, 1000]}
                   ]
```

5.5 Sampling Techniques

These are the techniques used to partition the dataset into subsets in order to obtain the reliable estimate of accuracy in the models. There are four main types of sampling techniques: Hold-out, Random Sub-sampling, K-Fold cross validation, stratified K-Fold cross validation.



5.15 Sampling methodology

These are described as follows:

1. **Hold-out Sub-Sampling Method**

   In this method, data is divided randomly into two subsets: training and testing. Typically, two third of the data is put into training set and one third is left for testing.

Output

```
No. of Training Images : 380
No. of Testing Images : 96

FINISHED classifying.
Average Accuracy score(%) :
0.9375

Test Dataset with maximum accuracy:


No. of Matches:
90 out of 96
```

2. **Random Sub-Sampling Method**

It is a variation of hold-out method with k times repeated sampling. The overall accuracy is taken as an average of all accuracies obtained in each iteration.

Output

```
Epoch Details:
    # Epoch  # of Training Images  # of Testing Images  Accuracy (%)
0         1                   380                   96     91.666667
1         2                   380                   96     91.666667
2         3                   380                   96     91.666667
3         4                   380                   96     92.708333
4         5                   380                   96     89.583333
5         6                   380                   96     94.791667
6         7                   380                   96     90.625000
7         8                   380                   96     93.750000
8         9                   380                   96     88.541667
9        10                   380                   96     92.708333
10       11                   380                   96     89.583333
11       12                   380                   96     91.666667
12       13                   380                   96     93.750000
13       14                   380                   96     89.583333
14       15                   380                   96     89.583333
15       16                   380                   96     87.500000
16       17                   380                   96     91.666667
17       18                   380                   96     95.833333
18       19                   380                   96     84.375000
19       20                   380                   96     89.583333

FINISHED classifying.
Average Accuracy score(%) :
91.04166666666667

No. of Matches:
92 out of 96
```

3. K-Fold Cross Validation Method

In this, the initial data is randomly divided into k mutually exclusive subsets or folds each having same size. Then, training and testing is performed k times. In i^{th}interation , the i^{th} fold is reserved as testing set and others are treated as training sets. Accuracy is obtained as the overall number of correct classifications from the k iterations divided by the total number of tuples in the initial data.

Output

```
Epoch Details:
   # Epoch  # of Training Images  # of Testing Images  Accuracy (%)
0        1                   292                   35     80.000000
1        2                   292                   35     77.142857
2        3                   292                   35     85.714286
3        4                   293                   34     79.411765
4        5                   293                   34     76.470588
5        6                   295                   32     81.250000
6        7                   295                   32     71.875000
7        8                   295                   32     78.125000
8        9                   297                   30     80.000000
9       10                   299                   28     75.000000

FINISHED classifying.
Average Accuracy score(%) :
78.49894957983193

No. of Matches:
30 out of 35
```

4. Stratified K-Fold Method

It is a variation of K-Fold but here, the division of data is done such that class distribution for the tuples in each fold is same as the class distribution in the initial data.

Output

```
Epoch Details:
   # Epoch  # of Training Images  # of Testing Images  Accuracy (%)
0        1                   425                   51     50.980392
1        2                   425                   51     88.235294
2        3                   426                   50     90.000000
3        4                   427                   49     83.673469
4        5                   427                   49     65.306122
5        6                   429                   47     63.829787
6        7                   430                   46     73.913043
7        8                   430                   46     84.782609
8        9                   432                   44     97.727273
9       10                   433                   43     51.162791

FINISHED classifying.
Average Accuracy score(%) :
74.96107809441472

No. of Matches:
43 out of 44
```

5.6 Ensembling Techniques

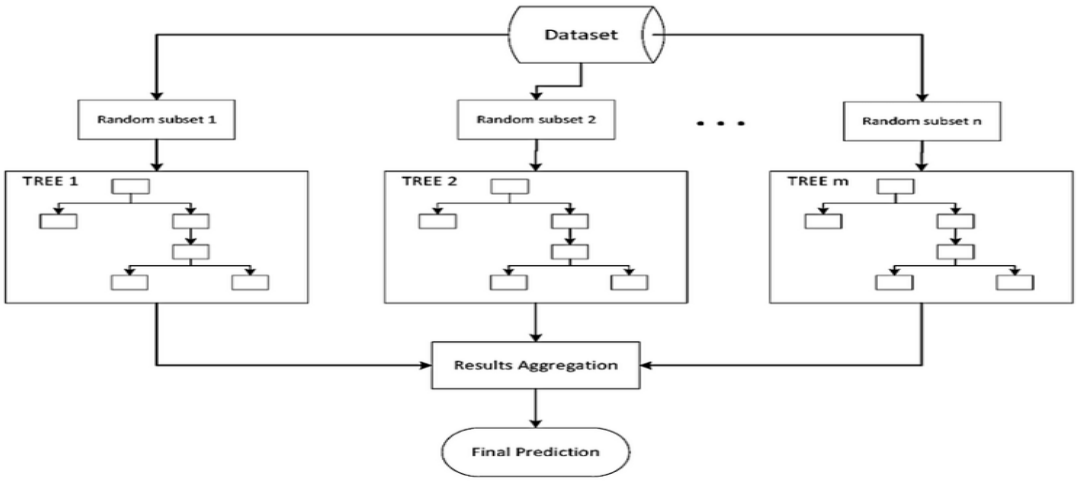
Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. That is why ensemble methods placed first in many prestigious machine learning competitions, such as the Netflix Competition, KDD 2009, and Kaggle.

Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to decrease variance(bagging), bias (boosting), or improve predictions (stacking) [11].

Types of Ensemble Methods

1. Bagging Aggregating

Bagging gets its name because it combines Bootstrapping and Aggregation to form one ensemble model. Given a sample of data, multiple bootstrapped subsamples are pulled. A Decision Tree is formed on each of the bootstrapped subsamples. After each subsample Decision Tree has been formed, an algorithm is used to aggregate over the Decision Trees to form the most efficient predictor. The image below will help explain:

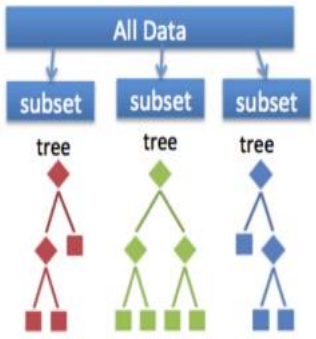


5.16 Ensembling using bagging/bootstrap aggregation

Given a Dataset, bootstrapped subsamples are pulled. A Decision Tree is formed on each bootstrapped sample. The results of each tree are aggregated to yield the strongest, most accurate predictor.

2. Random Forest Models

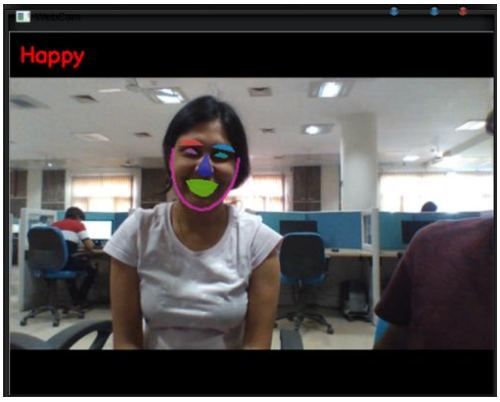
Random Forest Models can be thought of as bagging, with a slight tweak. When deciding where to split and how to make decisions, Bagged Decision Trees have the full disposal of features to choose from. Therefore, although the bootstrapped samples may be slightly different, the data is largely going to break off at the same features throughout each model. In contrary, Random Forest models decide where to split based on a random selection of features. Rather than splitting at similar features at each node throughout, Random Forest models implement a level of differentiation because each tree will split based on different features. This level of differentiation provides a greater ensemble to aggregate over, ergo producing a more accurate predictor.

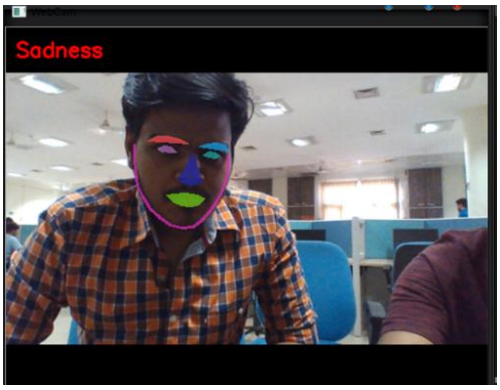


5.17 Random forest ensembling

Similar to Bagging, bootstrapped subsamples are pulled from a larger dataset. A decision tree is formed on each subsample. HOWEVER, the decision tree is split on different features (in this diagram the features are represented by shapes)

Output





CHAPTER-6

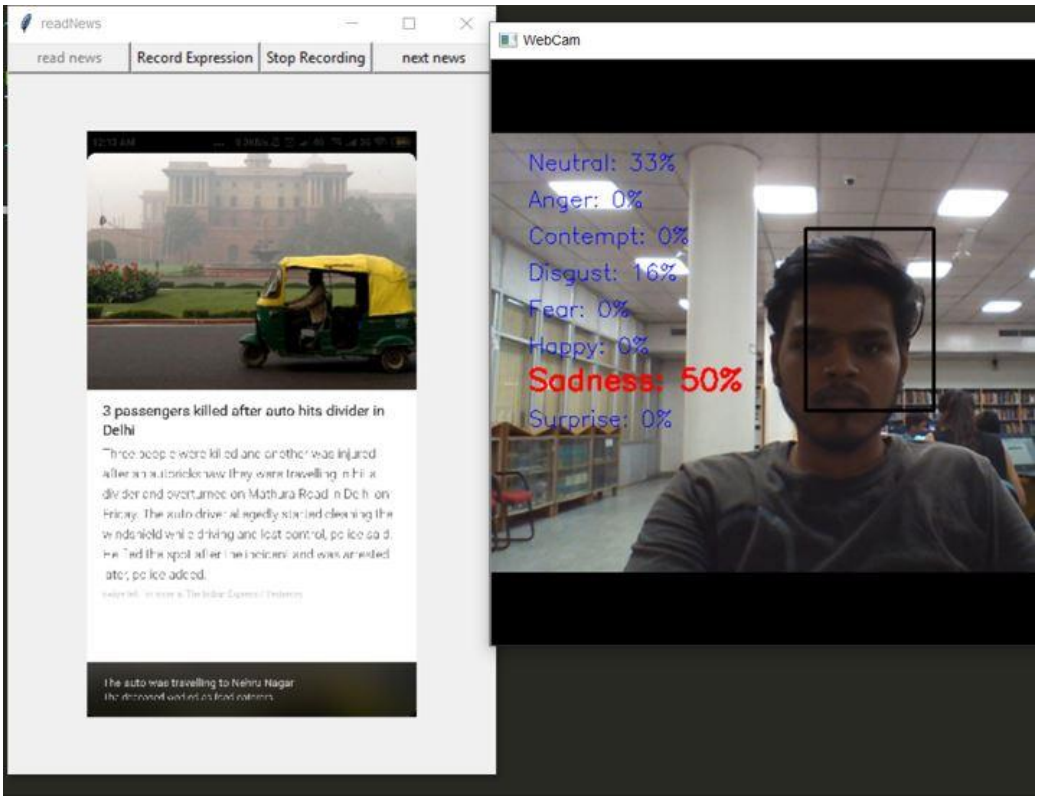
IMPLEMENTATION OF APPLICATIONS

6.1 News Categorization

News is information about current events. This may be provided through many different media: word of mouth, printing, postal systems, broadcasting, electronic communication, or through the testimony of observers and witnesses to events.

Common topics for news reports include war, government, politics, education, health, the environment, economy, business, fashion, and entertainment, as well as athletic events, quirky or unusual events.

This application deals with categorization of news by capturing the facial expressions of the end-user while reading the news. Various functions involved in its implementations are:

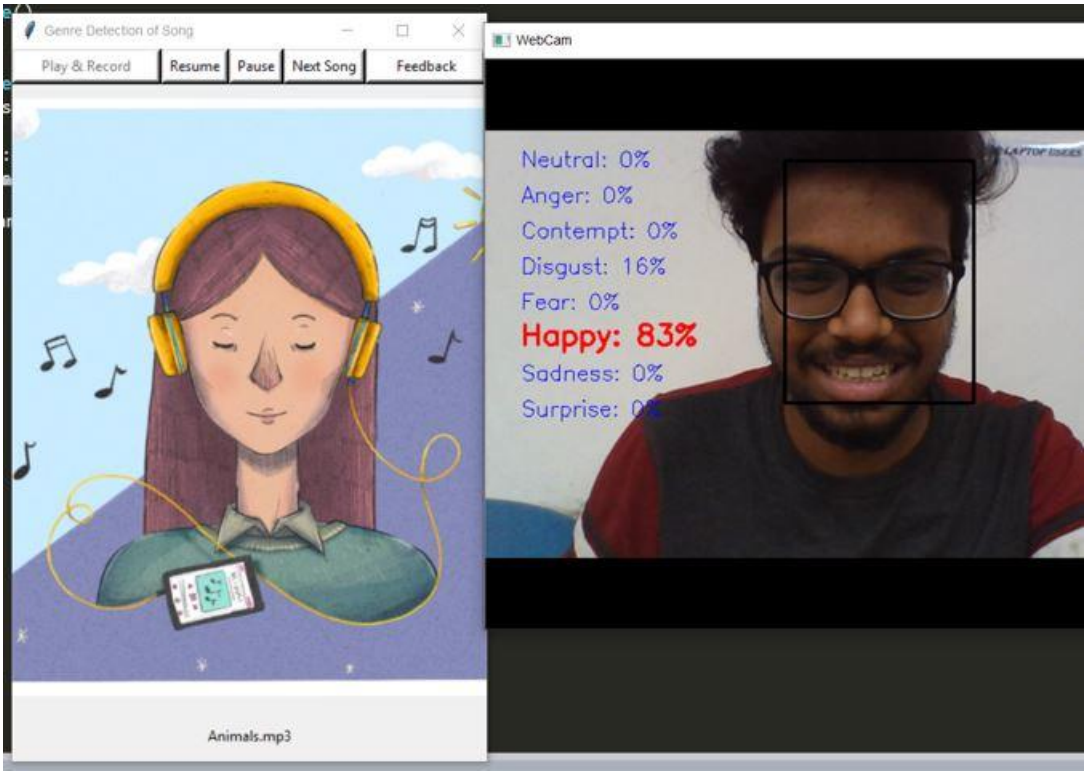


6.1 News categorization application working

6.2 Genre Detection of Songs

A song is a musical composition intended to be sung by the human voice. This is often done at distinct and fixed pitches using patterns of sound and silence. Music comes in many different types and styles ranging from traditional rock music to world pop, easy listening and bluegrass.

Some songs make our mood cheerful and happy while others can create nostalgia. Taking this into account, our application allows the user to listen to the song and his response is examined to discover the type of the song.



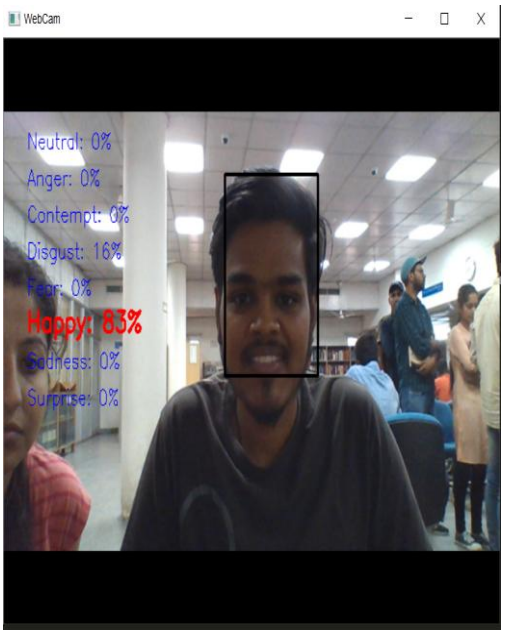
6.2 Genre detection of songs application working

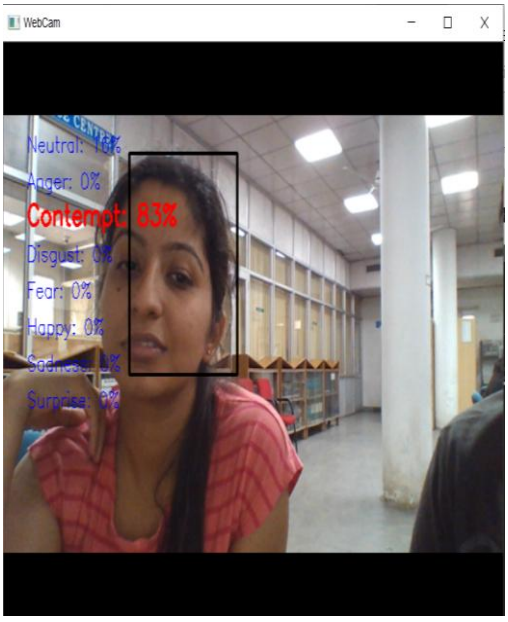
CHAPTER-7

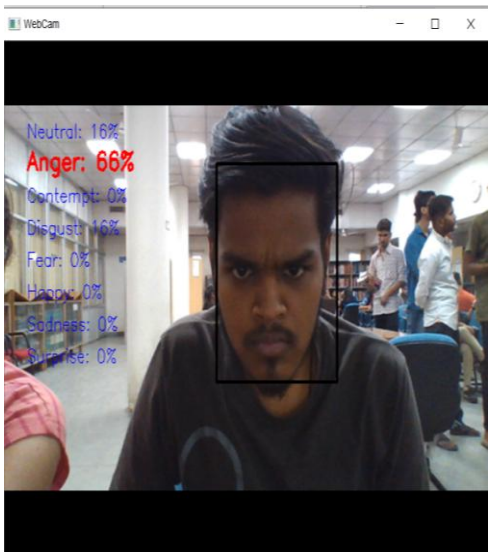
CONCLUSION

The model was tested on various samples in the real time and hence it was observed that result was directly proportional to intensity of training provided with the data set of about 10708 training images. The accuracy of nearly 67% was achieved.

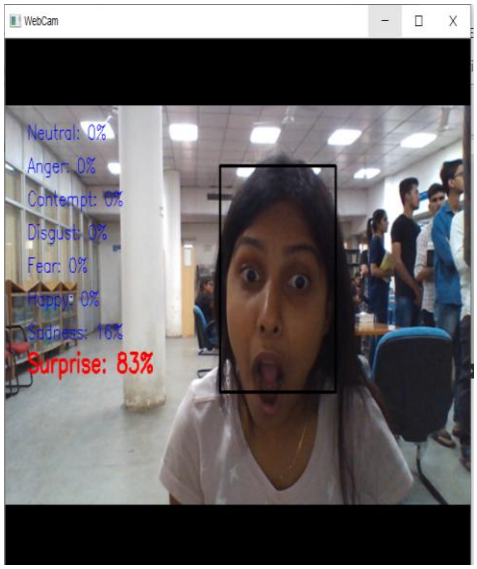
The emotion of real time samples with the probability distribution of each emotion is shown in the results.

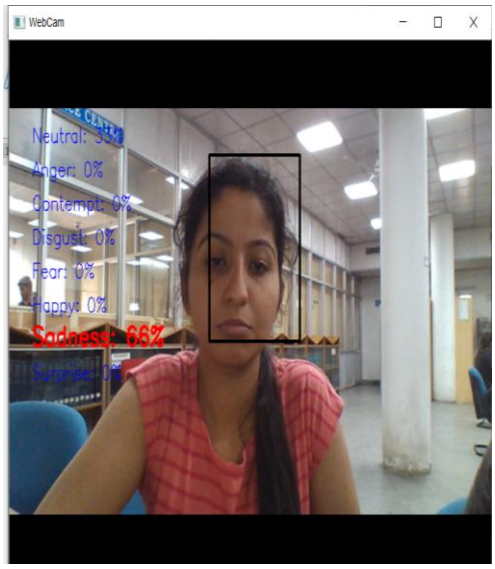





WebCam
Neutral: 16%
Anger: 66%
Contempt: 0%
Disgust: 16%
Fear: 0%
Happy: 0%
Sadness: 0%
Surprise: 0%


WebCam
Minimize
Neutral: 0%
Anger: 0%
Contempt: 0%
Disgust: 100%
Fear: 0%
Happy: 0%
Sadness: 0%
Surprise: 0%


WebCam
Neutral: 0%
Anger: 0%
Contempt: 0%
Disgust: 0%
Fear: 0%
Happy: 0%
Sadness: 16%
Surprise: 83%


WebCam
Neutral:
Anger: 0%
Contempt: 0%
Disgust: 0%
Fear: 0%
Happy: 0%
Sadness: 66%
Surprise:

CHAPTER-8

FUTURE SCOPE

This project is mainly the desktop application but it can be expanded to web application and for android applications. Accuracy is an important factor for any project and many efforts have been put for its enhancement. But still there is large scope of improvement.

This project can be extended and further implemented to be used by other models which are more efficient than SVM such as Neural Networks or the same SVM model when used with transfer learning, accuracy can be exponentially increased.

This project can be further extended to health care applications like providing counselling by analysing client's emotional state, capturing reaction of the patients about the treatment given to them. Also it makes easier for the autistic person to interpret the expressions of the other person.

It can also be used in online tutoring system to take the real time facial feedback for making learning more interactive and effective. It helps in marketing by providing the effective feedback of the product from the customer on watching the advertisement.

8.1 Future Scope Applications

CHAPTER-9

REFERENCES

[1] Mohammad Soleymani, Member, IEEE, Sadjad Asghari-Esfeden, Student member, IEEE Yun Fu, Senior Member, IEEE, MajaPantic, Fellow, IEEE, “Analysis of EEG signals and facial expressions for continuous emotion detection” IEEE Transactions on Affective Computing, volume 7, Issue :1, pg 17- 28, March 2016

[2] NirvairNeeru and Dr. Lakhwinder Kaur, “Face Recognition based on LBP and CS-LBP Technique under Different Emotions” Computational Intelligence and Computing Research(ICCIC), 2015 IEEE International Conference, Dec 2015.

[3] G. Tharshini, H. G. C. P. Dinesh, G. M. R. I. Godaliyadda and M. P. B. Ekanayake, “A Robust Expression Negation Algorithm for Accurate Face Recognition for Limited Training Data” ,Industrial and Information Systems(ICIIS), 2015 IEEE 10TH International Conference, Dec 2015.

[4]Detect eyes, nose, lips, and jaw with dlib, OpenCV, and Pythonhttps://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-python/

[5] Face Detection: https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9

[6] Kanade, T., Cohn, J. F., & Tian, Y. (2000). Comprehensive database for facial expression analysis. Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), Grenoble, France, 46-53.

[7] SVM (Support Vector Machine): https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

[8] Philipp Michel and Rana El Kalioub. Facial Expression Recognition Using Support Vector Machines. University of Cambridge, Cambridge CB3 0FD, U.K.

[9] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines,(2001). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[10] T. Ahsan, T. Jabid, and U.-P. Chong. Facial ex-pression recognition using local transitional pat-tern on gabor _ltered facial images. IETE Technical Review, 30(1):47{52, 2013.

[11] Ensemble Methods in Machine Learning: https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f