Bachelor Thesis Project on

Real Time Emotion Detection

Using Facial Expressions by Application of Convolutional Neural Networks

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT DEGREE OF UNIVERSITY OF DELHI

Bachelor of Engineering

COMPUTER ENGINEERING

SUBMITTED BY:

Deepak Rathi (244/CO/14)

Devesh Saini (247/CO/14)

Harshil Yadav (258/CO/14)

GUIDED BY:

Dr. M.P.S.Bhatia

(Professor, Division of Computer Engineering)

DIVISION OF COMPUTER ENGINEERING NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

UNIVERSITY OF DELHI 2017-2018

नेताजी सुभाष प्रौद्योगिकी संस्थान
NETAJI SUBHAS INSTITUTE OF TECHNOLOGY
(An Institution of Govt. of NCT of Delhi-Formerly, Delhi Institute of Technology)
Azad Hind Fauj Marg, Sector-3, Dwarka, New Delhi - 110 078
Telphone : 25099050; Fax : 25099025, 25099022 Website : http://www.nsit.ac.in

CERTIFICATE

The project titled **"REAL TIME EMOTION DETECTION USING FACIAL EXPRESSIONS BY APPLICATION OF CONVOLUTIONAL NEURAL NETWORKS"** by **Deepak Rathi (244/CO/14), Devesh Saini (247/CO/14) and Harshil Yadav (258/CO/14)** is a record of bonafide work carried out by us, in the Division of Computer Engineering, Netaji Subhas Institute of Technology, New Delhi, under the supervision and guidance of **Dr. MPS Bhatia** in partial fulfilment of requirement for the award of the degree of Bachelor of Engineering in Computer Engineering, University of Delhi in the academic year 2017 - 2018.

Dr. MPS Bhatia

Division of Computer Engineering

Netaji Subhas Institute of Technology

New Delhi

Date: _____

नेताजी सुभाष प्रौद्योगिकी संस्थान
NETAJI SUBHAS INSTITUTE OF TECHNOLOGY
(An Institution of Govt. of NCT of Delhi-Formerly, Delhi Institute of Technology)
Azad Hind Fauj Marg, Sector-3, Dwarka, New Delhi - 110 078
Telphone : 25099050; Fax : 25099025, 25099022 Website : http://www.nsit.ac.in

DECLARATION

This is to certify that the work which is being hereby presented by us in this project titled "Real Time Emotion Detection using Facial Expressions by application of Convolutional Neural Networks" in Partial fulfilment of the award of the Bachelor of Engineering submitted at the Department of Computer Engineering, Netaji Subhas Institute of Technology Delhi, is a genuine account of our work carried out during the period from January 2018 to May 2018 under the guidance of Dr. MPS Bhatia, Department of Computer Engineering, Netaji Subhas Institute of Technology Delhi. The matter embodied in the project report to the best of our knowledge has not been submitted for the award of any other degree elsewhere.

Deepak Rathi	**Devesh Saini**	**Harshil Yadav**
(244/CO/14)	**(247/CO/14)**	**(258/CO/14)**

Date: _____

Acknowledgment

This project's success is directly attributed to the assistance and support from all of those individuals involved in this five month period. We would like to express our sincere gratitude towards our mentor **Dr. MPS Bhatia**, Professor, Computer Engineering Department, Netaji Subhas Institute of Technology, Delhi under whose supervision we completed our work. His invaluable suggestions, enlightening comments and constructive criticism always kept our spirits up during our work. He was always available to help whenever we faced any problems. Our experience in working together has been great and fruitful. We learnt a lot in the process. The knowledge, practical and theoretical, that we have gained through this project will help us in our future endeavors in the field. We also learnt the importance of working as a team effectively and efficiently. We are also grateful to our friends who providing critical feedback and support whenever required.

TABLE OF CONTENTS

List of Figures

List of Tables

Abstract

It is argued that for the computer to be able to interact with humans, it needs to have the communication skills of humans. One of these skills is the ability to understand the emotional state of the person. The aim of this thesis is to build a cost effective system for Real Time Emotion Detection using Facial Features. Built on the Convolution Neural Networks, our strategy takes the live Webcam feed and analyses the feed to compute the emotions of the people based on the model developed using suitable weights.

Behaviors, actions, poses, facial expressions and speech; these are considered as channels that convey human emotions. Extensive research has being carried out to explore the relationships between these channels and emotions. This paper proposes a system which automatically recognizes the emotion represented on a face. Thus a neural network based solution combined with image processing is used in classifying the universal emotions: Happiness, Sadness, Anger, Disgust, Surprise and Fear. Colored frontal face images are given as input to the system. After the face is detected, image processing based feature point extraction method is used to extract a set of selected feature points. Finally, a set of values

obtained after processing those extracted feature points are given as input to the neural network to recognize the emotion contained.

1. Introduction

1.1. Motivation

As humans, we classify emotions all the time without knowing it. We can see if someone is happy or sad or frustrated and in need of help. However, this is a very complex problem that involves many subtleties about facial expression. Even just the tiniest change in someone's face can be a signal of a different emotion. Training models that understand human emotion will be critical to building truly intelligent machines that can interact with us as humans do. In the near future with the rise of augmented reality, there could also be many applications for an emotion classifier to help people who have trouble recognizing emotions interact in a world where this is an essential skill. Other applications include aiding federal interrogations and advertisement targeting based on emotional state.

Humans are well-trained in reading the emotions of others, in fact, at just 14 months old, babies can already tell the difference between happy and sad. **But can computers do a better job than us in accessing emotional states?** So we try to automate this process of emotion classification through the use of neural networks that receives video feed as its input and provides the detected emotion as its output. This process has many potential applications in real life. A noteworthy one is : adjusting the presentation style of an online tutor in E-learning applications.

1.2. PROBLEM STATEMENT

The problem that we attempt to tackle in this project is Real time detection of Emotions using Convolutional Neural Networks (CNN).

1.3. EMOTIONS

Emotion is any conscious experience characterized by intense mental activity and a certain degree of pleasure or displeasure.Emotion is often intertwined with mood, temperament, personality, disposition, and motivation.The physiology of emotion is closely linked to arousal of the nervous system with various states and strengths of arousal relating, apparently, to particular emotions. Emotion is also linked to behavioral tendency. Extroverted people are more likely to be social and express their emotions, while introverted people are more likely to be more socially withdrawn and conceal their emotions. Emotion is often the driving force behind motivation, positive or negative.For more than 40 years, Paul Ekman has supported the view that emotions are discrete, measurable, and physiologically distinct.His research findings led him to classify six emotions as basic: **anger, disgust, fear, happiness, sadness and surprise.**

1.4. EMOTION RECOGNITION

Emotion classification, the means by which one may distinguish one emotion from another, is a contested issue in emotion research and in

affective science. Researchers have approached the classification of emotions from one of two fundamental viewpoints:

1. that emotions are discrete and fundamentally different constructs
2. that emotions can be characterized on a dimensional basis in groupings

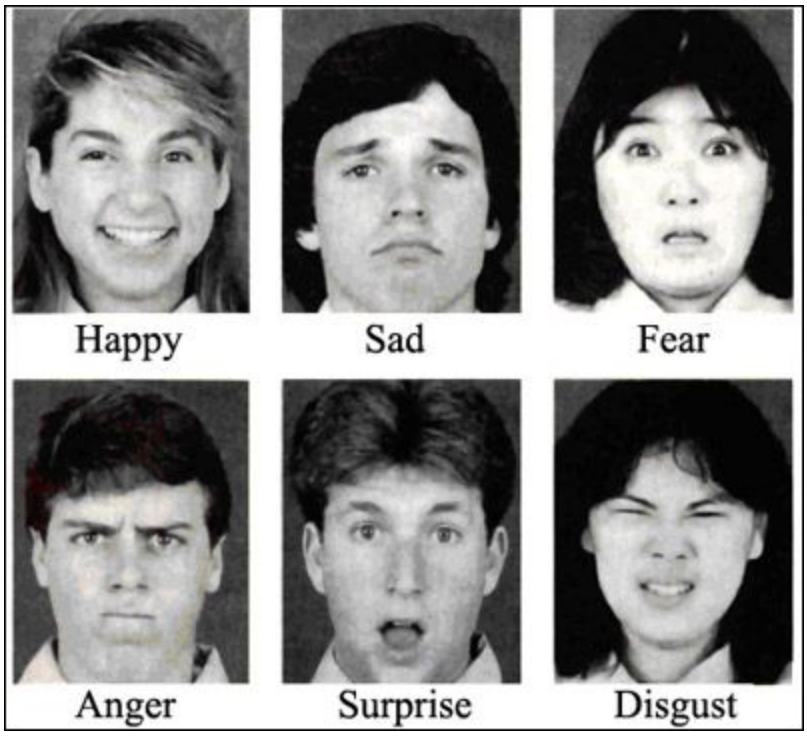

Fig 1.1 : Six Basic Emotions

1.5. CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (ConvNets or CNNs) are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. ConvNets have been successful in identifying faces, objects and traffic signs apart from powering vision in robots and self driving cars.

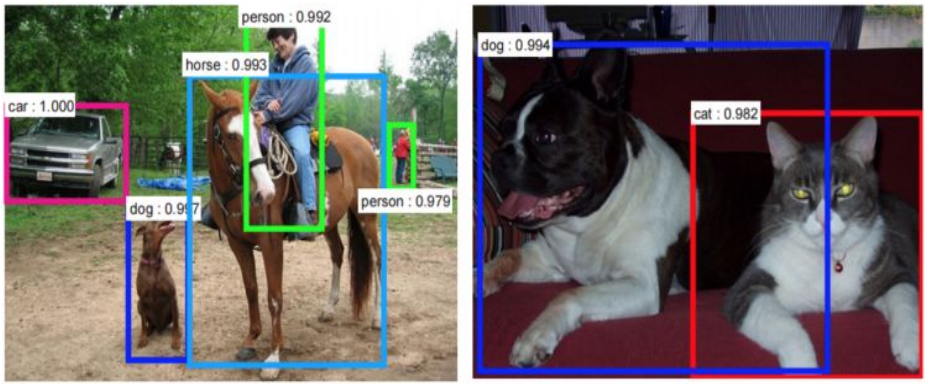

Fig 1.2 : Classification using Convolutional Neural Networks

Image : An Image is a matrix of pixel values Essentially, every image can be represented as a matrix of pixel values. Channel is a conventional term used to refer to a certain component of an image. An image from a standard digital camera will have three channels – red, green and blue. A grayscale image has just one channel.

There are four main operations in the ConvNet are:

1. Convolution
2. Non Linearity (ReLU)
3. Pooling or Sub Sampling
4. Classification (Fully Connected Layer)

The Convolution Step : The primary purpose of Convolution in case of a ConvNet is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. A filter slides over the input image (convolution operation) to produce a feature map. In practice, a CNN *learns* the values of these filters on its own during the training process . The more number of filters we have, the more image features get extracted and the better our network becomes at recognizing patterns in unseen images.

Non Linearity (ReLU) : ReLU stands for Rectified Linear Unit and is a non-linear operation.ReLU is an element wise operation (applied per pixel)

and replaces all negative pixel values in the feature map by zero. The purpose of ReLU is to introduce non-linearity in our ConvNet, since most of the real-world data we would want our ConvNet to learn would be non-linear.

The Pooling Step : Spatial Pooling (also called subsampling or downsampling) reduces the dimensionality of each feature map but retains the most important information. In case of Max Pooling, we define a spatial neighborhood (for example, a 2×2 window) and take the largest element from the rectified feature map within that window. Instead of taking the largest element we could also take the average (Average Pooling) or sum of all elements in that window. In practice, Max Pooling has been shown to work better.

Fully Connected Layer : The Fully Connected layer is a traditional Multi Layer Perceptron that uses a softmax activation function in the output layer (other classifiers like SVM can also be used, but will stick to softmax in this post). The term "Fully Connected" implies that every neuron in the previous layer is connected to every neuron on the next layer. The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset.

Training using Backpropagation : Backpropagation is a method used in artificial neural networks to calculate a gradient that is needed in the calculation of the weights to be used in the network. backpropagation is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of loss function. This technique is also sometimes called backward propagation of errors, because the error is calculated at the output and distributed back through the network layers.

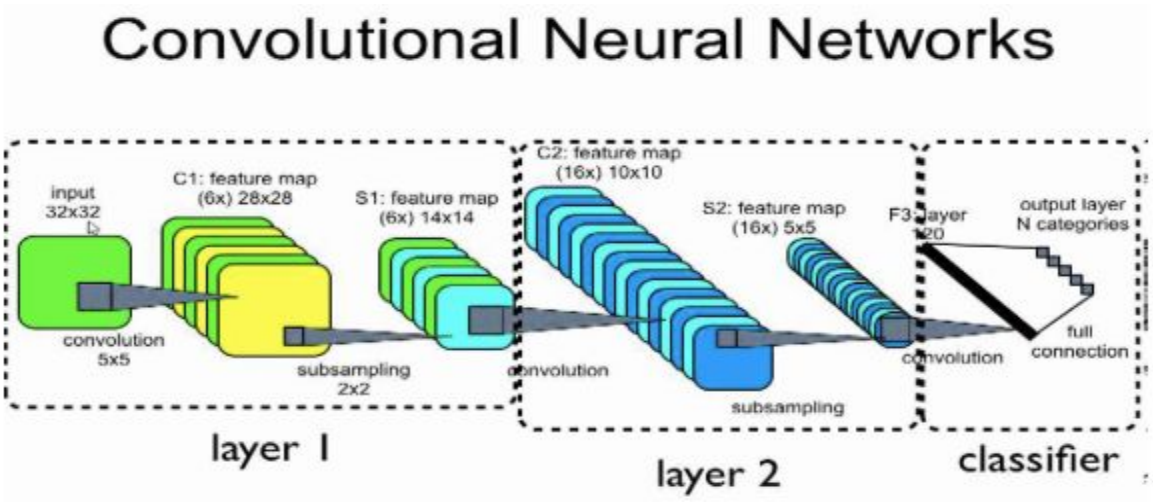

Fig 1.3 : Layers in Convolutional Neural Networks

1.6. Objective

The main objective is to develop an Facial Emotion Recognition System Application using the best architecture after comparing performance results of multiple architectures. Application developed should :

1. Take a live video feed as its input.
2. Process the video input frame-by-frame and pass this video input through a Convolution Neural Network (CNN) to identify various objects and relationships between them in the image.
3. Important features extracted from CNN are used to identify the emotions in real time and display the emotion to the user.
4. Achieve state of the art convolution neural network for emotion classification problem.
5. Perform emotion classification on the Kaggle ILSVRC2016 dataset with good accuracy.

1.7 Thesis Overview

This thesis is divided into 5 chapters:

Chapter 1 deals with the introduction to the thesis. It covers the problem statement, what actually is emotion recognition, its need, what are its features and how can they be put to use.

Chapter 2 carries out the literature review concerning the previous work done, different techniques and presents in detail, the current state of the art in this area,

Chapter 3 discusses about the implementation details of the thesis covering key areas like what are the environment and software requirements of the project, actual implementation of the proposed models and also gives detailed analysis about various result metrics.

Chapter 4 briefs about the user interface of the application, how to use and its functioning.

Chapter 5 summarizes the thesis with conclusions and avenues of future work.

2. Literature Review

2.1. Theoretical Background :

This emerging field has been a research interest for scientists from several different scholastic tracks, i.e. computer science, engineering, psychology, and neuroscience. In the past 20 years there has been much research on recognizing emotion through facial expressions.There are several approaches taken in the literature for learning classifiers for emotion recognition [1] [2]:

1. **Static approach :** Here the classifier classifies each frame in the video to one of the facial expression categories based on the tracking results of that frame. Bayesian network classifiers were commonly used in this approach. While Naive-Bayes classifiers were often successful in practice, they use a very strict and often unrealistic assumption that the features are independent given the class. Therefore, another approach using Gaussian TAN classifiers have the advantage of modeling dependencies between the features without much added complexity compared to the Naive-Bayes classifiers.

2. **Dynamic approach :** These classifiers take into account the temporal pattern in displaying facial expression. Hidden Markov model (HMM) based classifiers for facial expression recognition has been previously used in recent works. Cohen and Sebe [1] further advanced this line of research and proposed a multi-level HMM classifier, combining the temporal information, which allowed not only to perform the classification of a video segment to the corresponding facial expression, as in the previous works on HMM based classifiers, but also to automatically segment an arbitrary long video sequence to the different expressions segments without sorting to empirical methods of segmentation.

An important problem in the emotion recognition field is the lack of agreed upon benchmark database and methods for compare different methods' performance. There are many datasets available on the internet on which many people have trained their models :

1. **Wild Images from the Labeled Faces in the Wild database** [3] : Wild images are spontaneous, unplanned and variety of angles,noises,occlusion and illumination levels. The Labeled Faces in the Wild database consists of 13,000 unconstrained images of public figures collected from the internet. These images are not centered or processed in any way, except that we convert them to grayscale for consistency.

2. **Dataset from Kaggle (Facial Expression Recognition Challenge)** [4] **:** The provided dataset contains preprocessed images that are centered and adjusted with faces occupying almost the same amount of space in each image. The Kaggle dataset (from the Facial Expression Recognition Challenge) meets all the following attributes:

    1. 35,887 images

2. Image Format: 48 x 48 pixels (8-bit grayscale)
3. Various individuals across the entire spectrum of: ethnicity, race, gender and race, with all these images being taken at various angles
4. Contains the seven key emotions shown in the figure :

Fig 2.1 : Example Images of seven emotions in Kaggle Dataset

5. These seven key emotions are relatively equally distributed with the one exception being disgust, at ~1.5% shown in the figure.

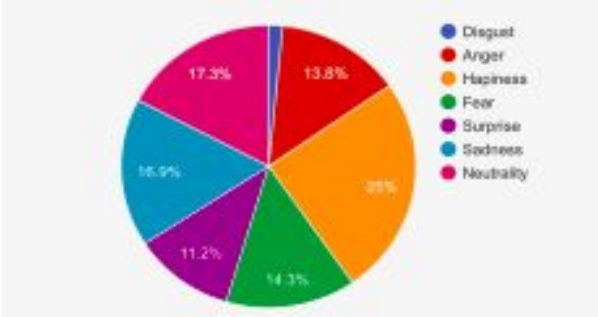

Fig 2.2 : Data distribution of Kaggle dataset across seven emotions

3. **Karolinska Directed Emotional Faces datasets :** The key attributes of Karolinska Directed Emotional Faces datasets are :

1. 4900 images
2. Image Format: 562 x 762 (32-bit RGB)
3. 70 individuals, each displaying the seven different emotional expressions, and each expression is photographed twice from five different angles
4. Representative across ethnicity, race, sex and gender
5. The seven key emotions are uniformly distributed

4. **Tiny ImageNet Dataset :** The original ILSVRC2016 dataset is a large set of hand labeled photographs consisting of 10,000,000 labeled images depicting 10,000+ object categories. These pictures are collected from flickr and other search engines, labeled with the presence of absence of 1000 object categories. The Tiny ImageNet dataset consists of the same data but the images are cropped into size of 64x64 from 224x224. It has 200 classes instead of 1,000 of ImageNet challenge, and 500 training images for each of the classes. In addition to its 100,000 training data, it has 10,000 validation images and 10,000 test images (50 for each class). It uses one-hot labels.

2.2 IMAGE AUGMENTATION :

In spite of all the data availability, fetching the right type of data which matches the exact use-case of our experiment is a daunting task. Moreover, the data has to have good diversity as the object of interest needs to be present in varying sizes, lighting conditions and poses if we desire that our network generalizes well during the testing (or deployment) phase. To overcome this problem of limited quantity and limited diversity of data, preprocessing of existing data is needed. This methodology of generating own data is known as data augmentation.These frameworks are giving in-built packages for data augmentation. To state a few of the frameworks, Keras has ImageDataGenerator, Tensorflow has TFLearn's DataAugmentation and MXNet has Augmenter classes.

Here is the index of techniques that can be used:

1. Scaling
2. Translation

3. Rotation (at 90 degrees)
4. Rotation (at finer angles)
5. Flipping
6. Adding Salt and Pepper noise
7. Lighting condition
8. Perspective transform

Before any technique, Image resizing should be done. Images gathered from Internet will be of varying sizes. Due to presence of fully connected layers in most of the neural networks, the images being fed to network will be required of a fixed size. Because of this, before the image augmentation happens, the images are preprocessed to the size which the network needs.

1. **Scaling :** Having differently scaled object of interest in the images is the most important aspect of image diversity. When network is in hands of real users, the object in the image can be tiny or large. Also, sometimes, object can cover the entire image and yet will not be present totally in image.
2. **Translation :** The network should recognize the object present in any part of the image. Also, the object can be present partially in the corner or edges of the image. For this reason,the object can be shifted to various parts of the image. This may also result in addition of a background noise.
3. **Rotation(at 90 degrees) :** The network has to recognize the object present in any orientation. Assuming the image is square, rotating the image at 90 degrees will not add any background noise in the image.
4. **Rotation (at finer angles) :** Depending upon the requirement, there maybe a necessity to orient the object at minute angles. However problem with this approach is, it will add background noise. If the background in image is of a fixed color (say white or black), the newly added background can blend with

the image. However, if the newly added background color doesn't blend, the network may consider it as to be a feature and learn unnecessary features.

5. **Flipping :** This scenario is more important for network to remove biasness of assuming certain features of the object is available in only a particular side. Also notice that flipping produces different set of images from rotation at multiple of 90 degrees. Consider, data can be generated with good amount of diversity for each class and time of training is not a factor.
6. **Adding Salt and Pepper noise:** Salt and Pepper noise refers to addition of white and black dots in the image. Though this may seem unnecessary, it is important to remember that a general user who is taking image to feed into network may not be a professional photographer. His camera can produce blurry images with lots of white and black dots. This augmentation aides the above mentioned users.
7. **Lighting condition:** This is a very important type of diversity needed in the image dataset not only for the network to learn properly the object of interest but also to simulate the practical scenario of images being taken by the user. The lighting condition of the images are varied by adding Gaussian noise in the image.
8. **Perspective transform:** In perspective transform, image is projected from a different point of view. For this, the position of object should be known in advance. Merely calculating perspective transform without knowing the position of the object can lead to degradation of the dataset. Hence, this type of augmentation has to be performed selectively. The greatest advantage with this augmentation is that it can emphasize on parts of object in image which the network needs to learn.

2.3 EMOTION RECOGNITION SYSTEMS :

The interaction between human beings and computers will be more natural if computers are able to perceive and respond to human non-verbal communication such as emotions. Although several approaches have been proposed to recognize human emotions based on facial expressions or speech, relatively limited work has been done to fuse these two, and other, modalities to improve the accuracy and robustness of the emotion recognition system.All the researchers compared the existing models with their own improved models and then presented their results based on the parameters like accuracy, hypertunning of weights, loss, gradient or overfitting of the model.

2.3.1 Emotion recognition by speech :

Several approaches to recognize emotions from speech have been reported [5][6]. Most researchers have used global suprasegmental/prosodic features as their acoustic cues for emotion recognition, in which utterance-level statistics are calculated. For example, mean, standard deviation, maximum, and minimum of pitch contour and energy in the utterances are widely used features in this regard. Dellaert et al. attempted to classify 4 human emotions by the use of pitch-related features [7]. They implemented three different classifiers:

1. Maximum Likelihood Bayes classifier (MLB)

2. Kernel Regression (KR)
3. K-nearest Neighbors (KNN).

Roy and Pentland classified emotions using a Fisher linear classifier [8]. Using short-spoken sentences, they recognized two kinds of emotions: approval or disapproval. They conducted several experiments with features extracted from measures of pitch and energy, obtaining an accuracy ranging from 65% to 88%.

The main limitation of those global-level acoustic features is that they cannot describe the dynamic variation along an utterance. To address this, for example, dynamic variation in emotion in speech can be traced in spectral changes at a local segmental level, using short-term spectral features.

In [9],[10] Mel-frequency cepstral coefficients (MFCC) were used to train a Hidden Markov Model (HMM) to recognize four emotions. Nwe et al. used 12 Mel-based speech signal power coefficients to train a Discrete Hidden Markov Model to classify the six archetypal emotions [11]. The average accuracy in both approaches was between 70 and 75%. Finally, other approaches have used language and discourse information, exploring the fact that some words are highly correlated with specific emotions [12]. In this study, prosodic information is used as acoustic features as well as the duration of voiced and unvoiced segments.

2.3.1.1 System based on speech

The most widely used speech cues for audio emotion recognition are global-level prosodic features such as the statistics of the pitch and the intensity. Therefore, the means, the standard deviations, the ranges, the maximum values, the minimum values and the medians of the pitch and the energy were computed using Praat speech processing software [13]. In addition, the voiced/speech and unvoiced/speech ratio were also estimated.

By the use of sequential backward features selection technique, a 11-dimensional feature vector for each utterance was used as input in the audio emotion recognition system.

2.3.1.2 Observation from Acoustic emotion classifier

Table 2.1 shows the confusion matrix of the emotion recognition system based on acoustic information, which gives details of the strengths and weaknesses of this system. The overall performance of this classifier was 70.9 percent. The diagonal components of table 2.1 reveal that all the emotions can be recognized with more than 64 percent of accuracy, by using only the features of the speech. However, table 2.1 shows that some pairs of emotions are usually confused more. Sadness is misclassified as neutral state (22%) and vise versa (14%). The same trend appears between happiness and anger, which are mutually confused (19% and 21%, respectively).

	Anger	Sadness	Happiness	Neutral
Anger	0.68	0.05	0.21	0.05
Sadness	0.07	0.64	0.06	0.22
Happiness	0.19	0.04	0.70	0.08
Neutral	0.04	0.14	0.01	0.81

Table 2.1 : Confusion Matrix of Emotion Recognition System based on audio

2.3.2 Emotion recognition by facial expressions

Facial expressions give important clues about emotions. The features used are typically based on local spatial position or displacement of specific points and

regions of the face, unlike the approaches based on audio, which use global statistics of the acoustic features.

2.3.2.1 System based on facial expressions

In the system based on visual information, which is described in figure, the spatial data collected from markers in each frame of the video is reduced into a 4-dimensional feature vector per sentence, which is then used as input to the classifier. The facial expression system, which is shown in figure, is described below.

After the motion data are captured, the data are normalized:

1. All markers are translated in order to make a nose marker be the local coordinate center of each frame
2. One frame with neutral and close-mouth head pose is picked as the reference frame
3. Three approximately rigid markers (manually chosen) define a local coordinate origin for each frame
4. Each frame is rotated to align it with the reference frame. Each data frame is divided into five blocks: forehead, eyebrow, low eye, right cheek and left cheek area. For each block, the 3D coordinate of markers in this block is concatenated together to form a data vector.
5. Principal Component Analysis (PCA) method is used to reduce the number of features per frame into a 10-dimensional vector for each area, covering more than 99% of the variation. Notice that the markers near the lips are not considered, because the articulation of the speech might be recognized as a smile, confusing the emotion recognition system [6].

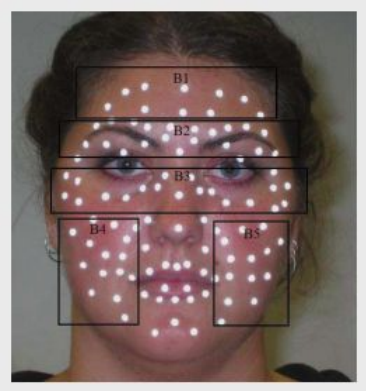

Fig 2.3 : Areas of the face considered for Emotion Recognition System

Therefore, several approaches have been proposed to classify human affective states:

1. Classifier to detect human emotion using facial expressions : CNN architecture[14]

- **DataSet :** The dataset from the Kaggle challenge on Facial Expression Recognition is used, which gives 48x48 pixel grayscale images of faces and labels them using the established seven types of emotions: anger, disgust, fear, joy, neutral, sadness and surprise. This dataset is split between training, validation, and test as follows: 28,709 - 3,589 - 3,589. The Kaggle dataset contains images that vary in viewpoint, lighting, and scale.

(a) (b)

Fig 2.4 : Sample Images in training dataset (a) Image with label “happy” but have watermark on it (b) image with label “happy”

- **Architecture Experiments** : The experiments are done with the different layer types and iterates on architecture design in response to training and validation results. The brief overview of the network layers experimented with :
    1. **Convolutional Layer**: Convolutional Layers convolve the input by applying a filter with kernel size of n x m. Each CNN neuron has n x m connections to a local region of the input. The final output is a 3D volume of multiple filters.

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{ab} y_{(i+a)(j+b)}^{l-1}$$

    2. **Max Pooling**: Pool layers will perform a downsampling operation along the spatial dimensions of the input layer to reduce the size of the input max.

$$max\{x_{i+k,i+l} | k| \leq \frac{m}{2}, |l| \leq \frac{m}{2}\}$$

    3. **Batch Normalization**: These layers help avoid issues in initialization of the input by forcing activations throughout the network to take on a unit Gaussian distribution.
    4. **Rectified Linear Unit**: ReLU layers apply an elementwise activation function, such as thresholding at zero with max(0,x). The size of the input volume remains unchanged.

$$R(x) = max(0, x)$$

    5. **Fully Connected Layer**: Fully connected layers have, as the name implies, neurons that are connected to every other weight in the input volume. The final output results in a vector with dimensions the size of the number of classes.

6. **Softmax and Cross Entropy Loss**: The softmax classifier outputs intuitive normalized class probabilities by squashing the input between zero and one. The cross entropy between these probabilities and the true distribution, with all probability on a single label per sample, is minimized during back propagation of the error.

$$S(x)_j = \frac{e^{x_j}}{\sum_{i=1}^{N} e^{x_i}}$$

7. **Adam Optimizer**: Adam is used to descend down the gradient of the loss smoothly based upon the magnitudes of the previous gradients.

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 m_{t-1} + (1-\beta_2) g_t^2 \\ \hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\ \theta_{t+1} &= \theta_t \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \end{aligned}$$

- **Results :**
    1. **Accuracy** : Results are good even with baseline model. For training accuracy, 18147 / 28709 images are classified correctly and for validation 1429 / 3589 images are classified correctly.
    2. **Confusion Matrix** : The classification is fairly good, as seen by the strong presence along the diagonal.

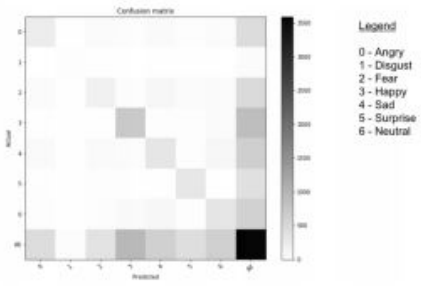

Fig 2.5 : Confusion matrix on the test set using final model

3. **Saliency Maps** : Saliency maps show which pixels contribute the most to the classification, helping us to gain insight into how our convolutional net is making classifications.For example, in the saliency maps, for the images classified as “happy” the pixels around the smile or presence of dimples contribute the most to the classification of happy. This particularly interesting because the CNN has learned to pick up on the subtle features that we as humans use to classify emotions subconsciously.

2. Facial Expression Recognition for wild images with analysis from Saliency maps[15]

- **Dataset :** Dataset from the Kaggle “Challenges in Representation Learning: Facial Expression Recognition Challenge” competition [4] is used. The provided dataset for this challenge consists of 35887 48 × 48 pixel grayscale images. These are preprocessed images that are centered and adjusted with faces occupying almost the same amount of space in each image. The dataset is divided into 28709 images for training, 3589 for validation, and 3589 for testing. This dataset contains the pixel values and emotion label for each image. The labels are numerical and represent one of seven categories of facial expressions: anger (0), disgust (1), fear (2), happiness (3), sadness (4), surprise (5), and neutrality (6).

  For our test data, models are tested on both the test images from the Kaggle dataset (grayscale images) as well as sampled wild images from the Labeled Faces in the Wild database [3]. The Labeled Faces in the Wild database consists of 13,000 unconstrained images of public figures collected from the internet. These images are not centered or processed in any way, except that they converted to grayscale for consistency before running on models on them.

- **Architectures :** Several different architectures – a shallow 3-layer CNN, AlexNet, VGG-16, Inception, and Inception-Resnet are used. Existing architectures are trained on the Kaggle dataset and benchmark their performance against their performance when pre-trained on ImageNet. The architecture for the three-layer CNN is [conv - relu - 2x2 max pool] – [affine – relu] – [affine.]For the architectures mentioned, different filter sizes, network depths, and update rules are tried to see how they impact performance. Saliency maps are used on the existing architectures to understand the differences in what they look for in "wild" images.
- **Results :** Various experiments are done with different weight scales and numbers of hidden dimensions and learning rates and found the following as our current optimum. Final accuracy with the shallow three-layer CNN was 49.74%.In the confusion matrix shown in Fig. 18, Except for disgust and fear, the three-layer CNN classifies an image into the correct often classified as anger. Fear and sadness share the same characteristics of puemotion class the majority of the time. Fear is most often classified as sadness and disgust is mostly apart lips and tense foreheads and are often both present in an expression, so this may be why they are confused by the CNN.

		Predicted						
		Anger	Disgust	Fear	Happy	Sad	Surprise	Neutral
Actual	Anger	35.6%	0.0%	11.8%	9.6%	30.5%	2.0%	10.4%
	Disgust	30.9%	23.6%	3.6%	5.5%	29.1%	1.8%	5.5%
	Fear	13.3%	0.0%	27.7%	8.5%	33.5%	9.3%	7.8%
	Happy	4.7%	0.0%	4.8%	68.4%	17.3%	1.4%	3.5%
	Sad	9.4%	0.2%	9.6%	9.3%	57.6%	3.4%	10.6%
	Surprise	4.3%	0.0%	14.2%	6.3%	10.6%	60.1%	4.6%
	Neutral	8.0%	0.3%	5.1%	12.6%	29.6%	3.2%	41.2%

Table 2.2 : Confusion matrix for the three-layer CNN

Similarly, disgust and anger share the same characteristics of burrowed eyebrows, narrow/pursed lips, and glaring eyes. They are also emotions that tend to occur together in an expression

Saliency Maps for Deep Neural Networks To get a better understanding of what the neural network learns during training, A study is conducted on how the saliency maps change over different training epochs, as shown in Fig 19. For example, when learning the expressions “sad”, “happy”, and “fear”, the AlexNet model is initially confused with gradients diverging everywhere. However, after two or three epochs, the model learns to focus on the eyebrow for the sad image, mouth for the happy image and eyes for the image with fear.

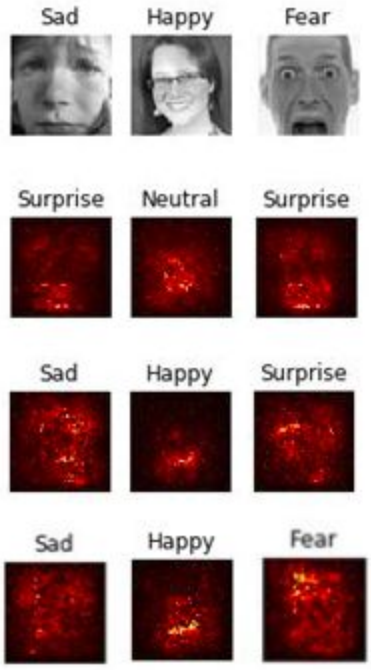

Fig 2.6 : The samples images from Fer2013 dataset (first row), saliency map and the result expression labels after first epoch (second row), after two epoch (third row) and after three epoch (fourth row)

After training, the test accuracy for AlexNet was 57% and 55% for VGG-16. We found that the saliency maps are quite different with these two models even though they have very similar test accuracies. In general, the saliency map from Alexnet is darker than that from VGG-16 and converges closer to the face area. Especially for VGG-16, facial expression recognition tends to be wrong when the model fails to find the face itself. In addition, for both models, the mouth and eyes are the key points for expression detection.

2.3.3 Emotion recognition by bimodal data :

Relatively few efforts have focused on implementing emotion recognition systems using both facial expressions and acoustic information. De Silva et al. proposed a rule-based audio-visual emotion recognition system, in which the outputs of the unimodal classifiers are fused at the decision-level [17]. From audio, they used prosodic features, and from video, they used the maximum distances and velocities between six specific facial points.

A similar approach was also presented by Chen et al. [18], in which the dominant modality, according to the subjective experiments conducted in [19], was used to resolve discrepancies between the outputs of mono-modal systems. In both studies, they concluded that the performance of the system increased when both modalities were used together.

Yoshitomi et al. proposed a multimodal system that not only considers speech and visual information, but also the thermal distribution acquired by infrared camera [20]. They argue that infrared images are not sensitive to lighting conditions, which is one of the main problems when the facial expressions are acquired with conventional cameras. They used a database recorded from a female speaker that read a single word acted in five emotional states. They integrated these three modalities at decision-level using empirically determined weights. The performance of the system was better when three modalities were used together.

In [21] and [22], a bimodal emotion recognition system was proposed to recognize six emotions, in which the audio-visual data was fused at feature-level. They used prosodic features from audio, and the position and movement of facial organs from 206 video. The best features from both unimodal systems were used as input in the bimodal classifier. They showed that the performance significantly increased from 69.4% (video system) and 75% (audio system) to 97.2% (bimodal system). However they use a small database with only six clips per emotion, so the generalizability and robustness of the results should be tested with a larger data set.

All these studies have shown that the performance of emotion recognition systems can be improved by the use of multimodal information. However, it is not clear which is the most suitable technique to fuse these modalities.

Four emotions -- sadness, happiness, anger and neutral state --are recognized by the use of bimodal information. The main purpose is to quantify the performance of unimodal systems, recognize the strengths and weaknesses of these approaches and compare different approaches to fuse these dissimilar modalities to increase the overall recognition rate of the system.

2.3.3.1 System Based on Bimodal system

To fuse the facial expression and acoustic information, two different approaches are implemented: feature-level fusion, in which a single classifier with features of both modalities are used (left of Figure 20); and, decision level fusion, in which a separate classifier is used for each modality, and the outputs are combined using some criteria (right of Figure 20).

In the first approach, a sequential backward feature selection technique was used to find the features from both modalities that maximize the performance of the classifier. The number of features selected was 10.
In the second approach, several criteria were used to combine the posterior probabilities of the mono-modal systems at the decision level: maximum, in which the emotion with greatest posterior probability in both modalities is selected; average, in which the posterior probabilities of each modalities are equally weighted and the maximum is selected; product, in which the posterior probabilities are multiplied and the maximum is selected; and, weight, in which different weights are applied to the different unimodal systems.

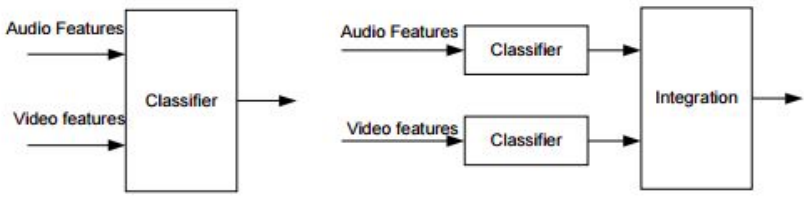

Fig 2.7 : Features-level and decision-level fusion

2.3.3.2 Observation from Bimodal system

Table 2.3 displays the confusion matrix of the bimodal system when the facial expressions and acoustic information are fused at feature-level. The overall performance of this classifier was 89.1 percent. As can be observed, anger,

happiness and neutral state are recognized with more than 90 percent of accuracy. As it was expected, the recognition rate of anger and neutral state was higher than unimodal systems. Sadness is the emotion with lower performance, which agrees with our previous analysis. This emotion is confused with neutral state (18%), because none of the modalities we considered can accurately separate these classes. Notice that the performance of happiness significantly decreased to 91 percent.

	Anger	Sadness	Happiness	Neutral
Anger	0.95	0.00	0.03	0.03
Sadness	0.00	0.79	0.03	0.18
Happiness	0.02	0.00	0.91	0.08
Neutral	0.01	0.05	0.02	0.92

Table 2.3 : Confusion matrix of the feature-level integration bimodal classifier

3. Implementation Details

3.1 Environment

3.1.1 Hardware Used

Laptop :

1. The machine used for analyzing data, for visualizing results and testing the application was manufactured by HP, it has the following specifications - Intel® Core™ i5-4210U CPU @ 1.70GHz × 4, GeForce 840M/PCIe/SSE2, 8GB DDR3 RAM, 1TB Hard Disk.
2. The machine used for testing and running the application was manufactured by Apple, it has the following specifications - Intel®

Core™ i5@ 1.80GHz,Intel HD Graphics 6000,128GB SSD, 8GB LPDDR3 RAM.

3.1.2 Google Colaboratory

Colaboratory is a Google research project created to help disseminate machine learning education and research. It's a Jupyter notebook environment that requires no setup to use and runs entirely in the cloud. Colaboratory notebooks are stored in Google Drive and can be shared just as you would with Google Docs or Sheets. Colaboratory is free to use.

1. GPU used in the backend is distributed K80(at this moment).
2. The 12-hour limit is for a continuous assignment of VM. It means we can use GPU compute even after the end of 12 hours by connecting to a different VM.

3.1.3 Software Requirements

3.1.3.1 Operating System

This project was performed on machines running Ubuntu 18.04 (Bionic Beaver) and MacOS High Sierra.

1. **Ubuntu :** Ubuntu is a free and open source operating system and Linux distribution based on Debian Ubuntu is offered in three official editions: Ubuntu Desktop for personal computers, Ubuntu Server for servers and the cloud, and Ubuntu Core for Internet of things devices.

2. **MacOS :** MacOS is a series of graphical operating systems developed and marketed by Apple Inc.since 2001. It is the primary operating system for Apple's Mac family of computers. Within the market of desktop, laptop and home computers, and by web usage, it is the second most widely used desktop OS, after Microsoft Windows. The version used for the thesis is High SIerra which is the latest version of MacOS at the time of writing thesis.

3.1.3.2 Python 3 or later

Python is a widely used, general purpose, multi-paradigm dynamically-typed high-level programming language. Python was chosen for this project for its ability to allow rapid prototyping of applications and for its wide support-base.The project was implemented using Python 3.6.

3.1.3.3 Python Libraries Used

3.1.3.2.1 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hard copy formats and interactive environments across platforms. It can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, etc. Version 2.2.2 was used in the project.

3.1.3.2.2 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.

OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Version 3.4.0.14 was used in the project.

3.1.3.2.3 NumPy

NumPy is a general-purpose array-processing package designed to efficiently manipulate large multi-dimensional arrays of arbitrary records without sacrificing too much speed for small multi-dimensional arrays. Version 1.14.3 was used in the project.

3.1.3.2.4 Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow,CNTK or Theano. It was developed with a focus on enabling fast experimentation. Version 2.1.6 was used in the project. Key features of Keras are :

1. Allows for easy and fast prototyping(through user friendliness,modularity and extensibility).
2. Supports both convolutional networks and recurrent networks, as well as combinations of the two.
3. Runs seamlessly on CPU and GPU.

3.1.3.2.5 TensorFlow

TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains. Version 1.8.0 was used in the project.

3.1.3.2.6 Theano

Theano is a Python library that allows you to define, optimize and evaluate mathematical expressions involving multi-dimensional arrays efficiently. Version 1.0.1 was used in the project. Some keys features are : tight integration with NumPy, transparent use of GPU, efficient symbolic differentiation and speed optimizations.

3.1.3.2.7 Scikit-learn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Version 0.19.1 was used in the project.

3.1.3.2.8 SciPy

SciPy is a collection of mathematical algorithms and convenience functions built on the Numpy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data. Version 1.1.0 was used in the project.

3.1.3.2.9 Graphviz

Graphviz is open source graph visualization software. Graph visualization is a way of representing structural information as diagrams of abstract graphs and networks. Version 0.8.3 was used in the project.

3.1.3.2.10 Pydot

This module provides with a full interface to create handle modify and process graphs in Graphviz's dot language. Version 1.2.4 was used in the project.

3.1.3.2.11 Brewer2mpl

Brewer2mpl is a pure Python package for accessing colorbrewer2.org color maps from Python. With brewer2mpl you can get the raw RGB colors of all 165 colorbrewer2.org color maps. The color map data ships with brewer2mpl so no

internet connection is required.Version 1.4.1 was used in the project.

3.1.3.1.12 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.Version 0.23.0 was used in the project.

3.1.4 Other Softwares Used

3.1.4.1 Anaconda-Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands.

3.1.4.2 Spyder

Spyder is the Scientific Python Development Environment:

1. A powerful interactive development environment for the Python language with advanced editing, interactive testing, debugging and introspection features
2. A numerical computing environment thanks to the support of IPython (enhanced interactive Python interpreter) and popular Python libraries such as NumPy (linear algebra), SciPy (signal and image processing) or matplotlib (interactive 2D/3D plotting).

3.1.4.3 Virtual Environment

Virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them. This is one of the most important tools that most of the Python developers use. Basically Virtual environment helps to work on multiple projects on same machine, each one having different dependencies without clashing them.

3.2 Actual Implementation

3.2.1 Dataset Collection

We used the dataset from the Kaggle challenge on Facial Expression Recognition, which gives 48x48 pixel grayscale images of faces and labels them using the established seven
types of emotions: anger, disgust, fear, joy, neutral, sadness and surprise. We split between the data between training, validation, and test as follows: 28,709 - 3,589 - 3,589. The Kaggle dataset contains images that vary in viewpoint, lighting, and scale.The faces were automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. Each face

based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The data set consists of 35,887 images.This dataset was prepared by Pierre-Luc Carrier and Aaron Courville, as part of an ongoing research project.

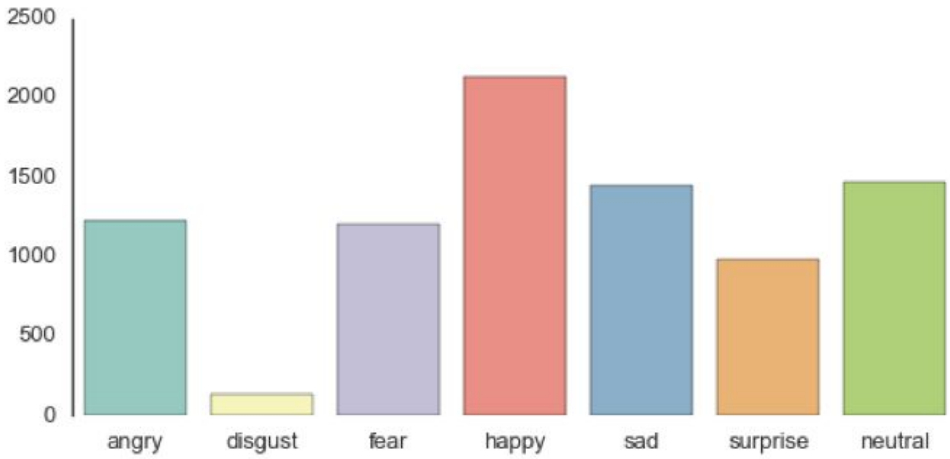

Fig. 3.1 : Data Set Distribution Visualization

Kaggle API

```
[1]  !pip install kaggle
     from google.colab import files
     uploaded = files.upload()
```

Downloading Dataset

```
[2]  import os
     ! mkdir .kaggle
     ! mv kaggle.json .kaggle/
     ! kaggle competitions download -c
     challenges-in-representation-learning-facial-expression-recognition-challenge --force
     ! cp ".kaggle/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge
     ! tar -xzf fer2013.tar.gz
     os.listdir()
```

Fig 3.2 : Code for Downloading Dataset

3.2.2 Data Preprocessing

The train.csv file contains two columns, "emotion" and "pixels". The "emotion" column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The "pixels" column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order. The data is preprocessed as explained in Fig 23. The data set is split into training , validation and test set in the ratio 80:10:10.

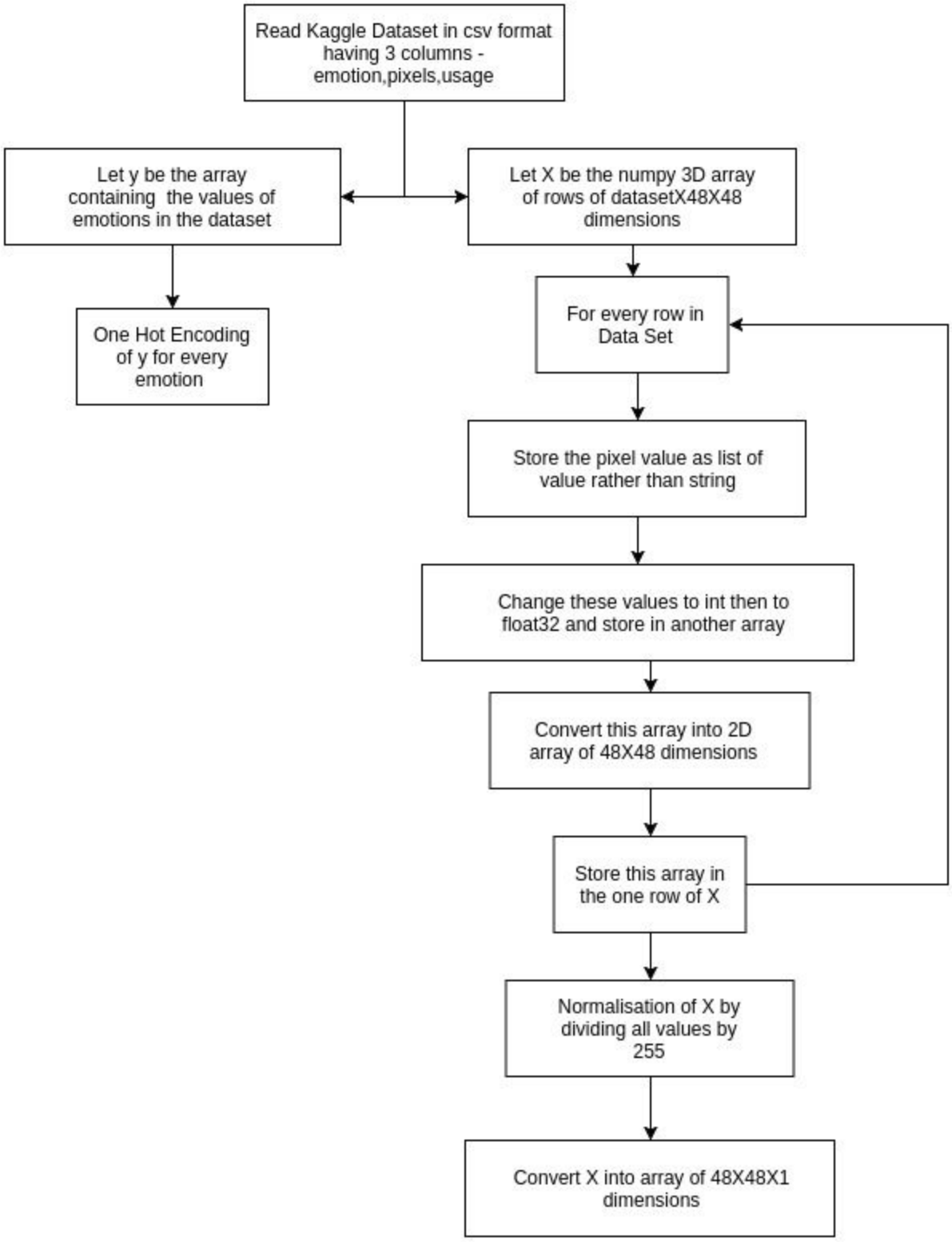

Fig 3.3 : Flow Chart of Data Preprocessing

3.2.3 Architecture of Neural Network

3.2.3.1 Layers used

The Sequential model is a linear stack of layers.The model needs to know what input shape it should expect. For this reason, the first layer in a Sequential model needs to receive information about its input shape. Before training a model, we need to configure the learning process, which is done via the compile method. Different layers used in our architecture are :

1. **Convolutional layers**, which apply a specified number of convolution filters to the image. For each subregion, the layer performs a set of mathematical operations to produce a single value in the output feature map. Convolutional layers then typically apply a ReLU activation function to the output to introduce nonlinearities into the model.
2. **Pooling layers**, which downsample the image data extracted by the convolutional layers to reduce the dimensionality of the feature map in order to decrease processing time. A commonly used pooling algorithm is max pooling, which extracts subregions of the feature map, keeps their maximum value, and discards all other values.

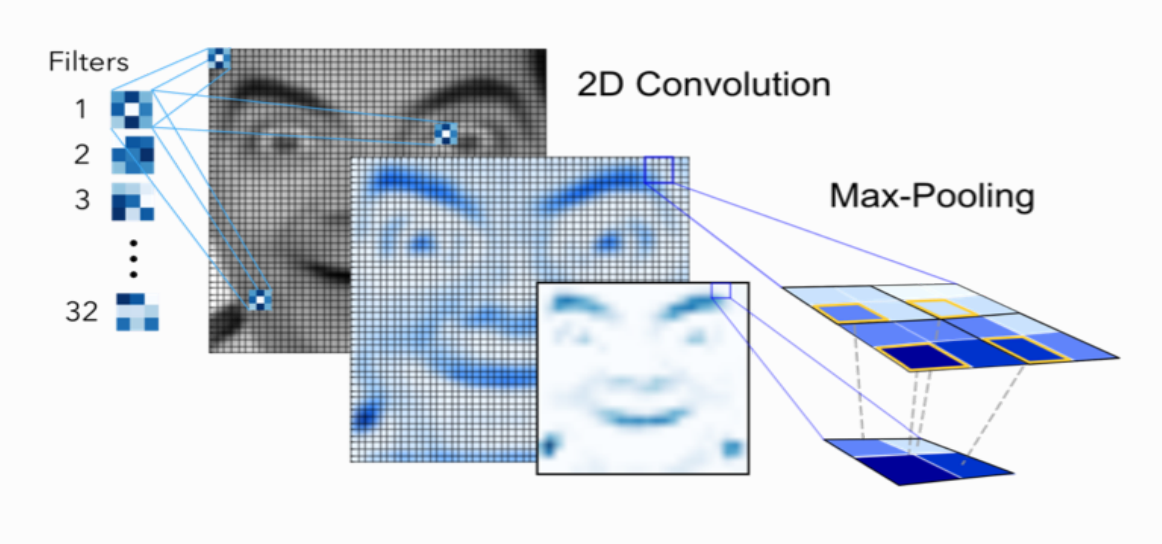

Fig 3.4 : 2D Convolution and Max-Pooling

3. **Rectifier** is an activation function defined as the positive part of its argument. This is also known as a ramp function and is analogous to half-wave rectification in electrical engineering. It has been demonstrated for the first time in 2011 to enable better training of deeper networks, compared to the widely used activation functions prior to 2011, i.e., the logistic sigmoid function.A unit employing the rectifier is also called a rectified linear unit (ReLU)

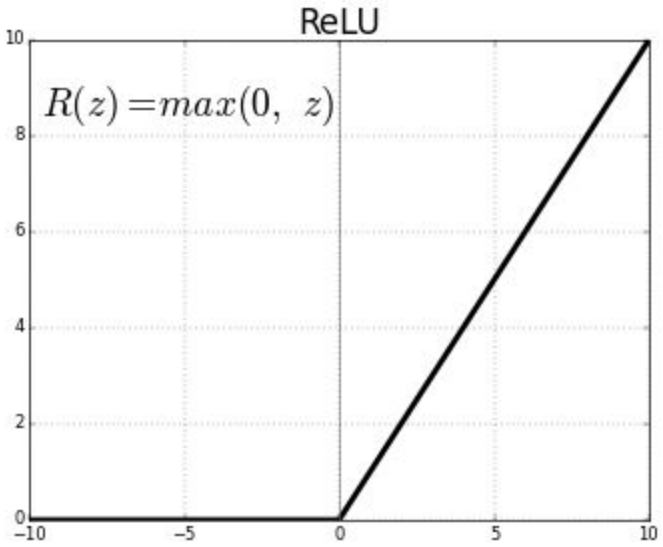

Fig 3.5 : Rectifier Linear Unit Function

4. **Dense (fully connected) layers**, which perform classification on the features extracted by the convolutional layers and downsampled by the pooling layers. In a dense layer, every node in the layer is connected to every node in the preceding layer.

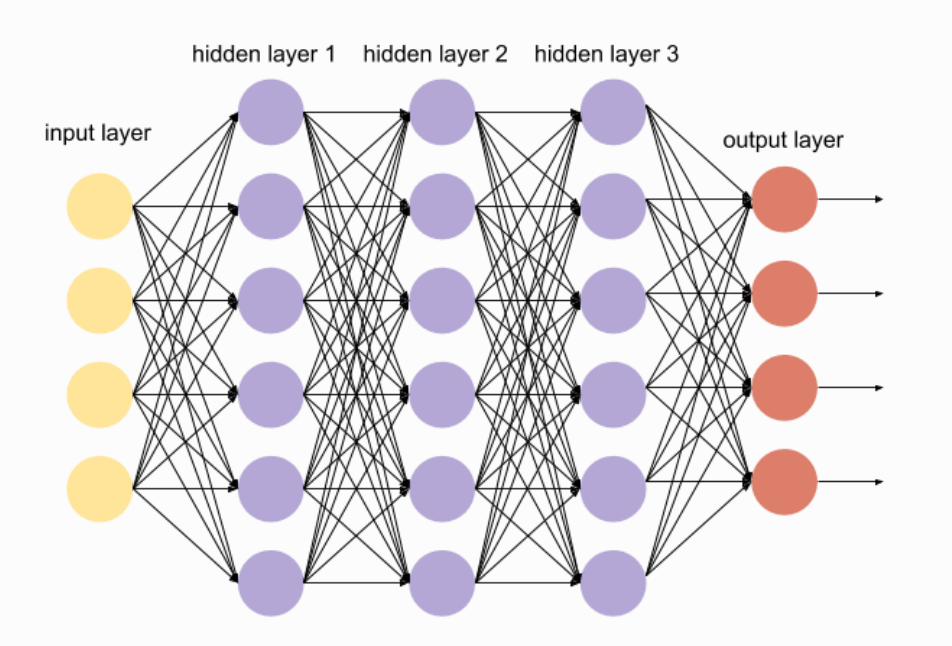

Fig 3.6 : Dense Layers

5. **Softmax function**, is a generalization of the logistic function that "squashes" a K-dimensional vector z of arbitrary real values to a K-dimensional vector sigma (z) of real values, where each entry is in the range (0, 1), and all the entries adds up to 1.

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

6. **Dropout** is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. It is a very efficient way of performing model averaging with neural networks. The term "dropout" refers to dropping out units (both hidden and visible) in a neural network.
7. **Batch normalization** is a technique for improving the performance and stability of neural networks. It is a technique to provide any layer in a neural network with inputs that are zero mean/unit variance.It is used to normalize the input layer by adjusting and scaling the activations.

3.2.3.2 Model Used

We built our model entirely using Keras. For development, we used an iPython Notebook provided by Google Colaboratory and we developed using GPUs on Google Cloud. We would run each of our models for 16 epochs on the cloud after which the performance seemed to level off. The 16 iterations would take anywhere from 20 to 45 minutes, so we were able to iterate fairly quickly with all of us testing parameters and architectures at the same time.

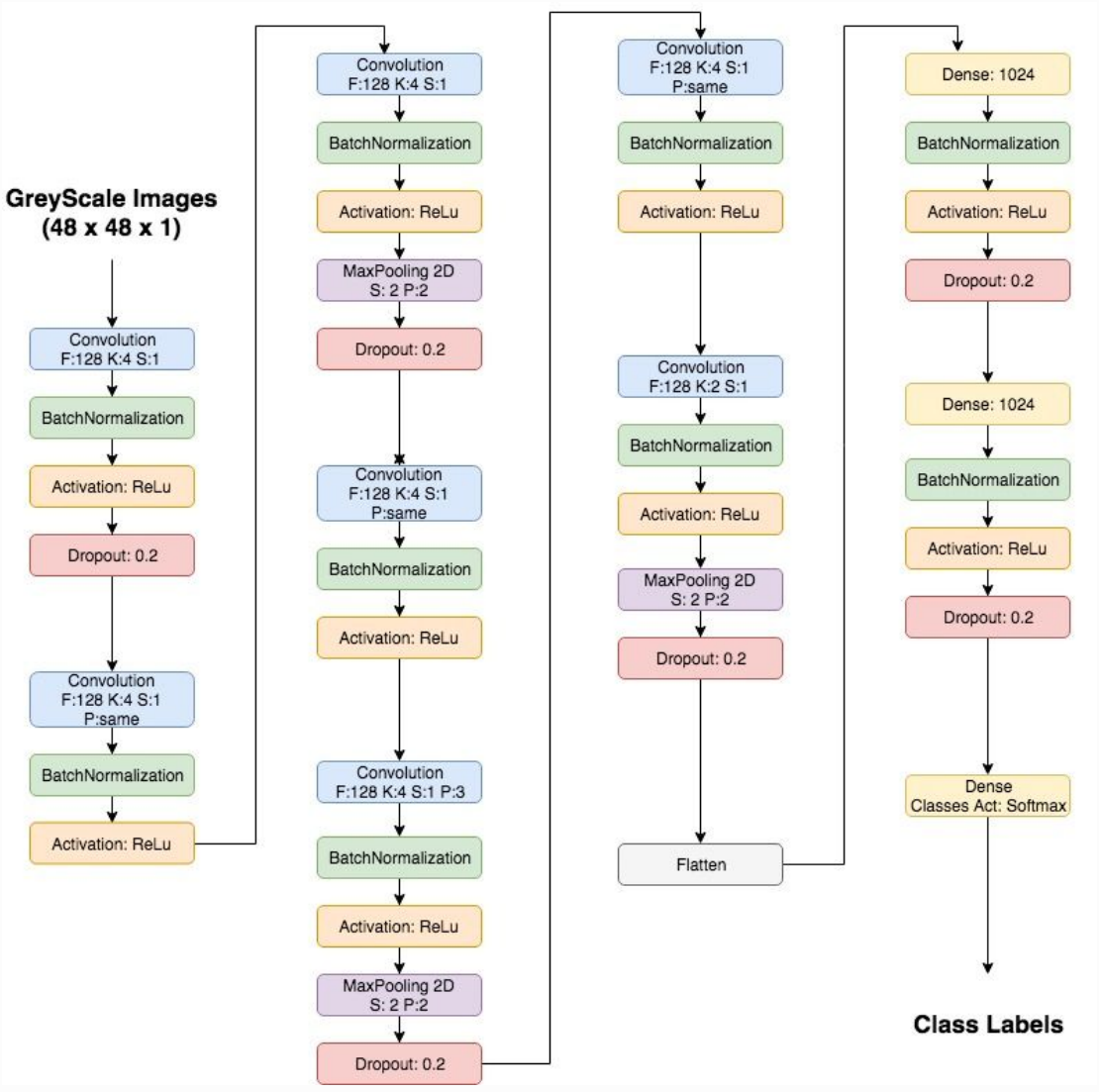

Fig 3.7 : Model Architecture

3.2.3.3 Callbacks

A callback is a set of functions to be applied at given stages of the training procedure. We used callbacks to get a view on internal states and statistics of the model during training. We passed a list of callbacks to the .fit() method of the Sequential or Model classes.

On epoch end , accuracy on training and validation set are computed and appended to the list.

```
[ ]  class AccHistory(keras.callbacks.Callback):
         def on_train_begin(self, logs={}):
             self.train_acc = []
             self.val_acc = []

         def on_epoch_end(self, batch, logs={}):
             self.train_acc.append(logs.get('acc'))
             self.val_acc.append(logs.get('val_acc'))
```

Fig 3.8 : Class for storing Accuracy after every epoch

On epoch end , loss on training and validation set are computed and appended to the list.

```
[ ]  class LossHistory(keras.callbacks.Callback):
         def on_train_begin(self, logs={}):
             self.train_losses = []
             self.val_losses = []

         def on_epoch_end(self, batch, logs={}):
             self.train_losses.append(logs.get('loss'))
             self.val_losses.append(logs.get('val_loss'))
```

Fig 3.9 : Class for storing Loss after every epoch

Checkpointing is setup to save the network weights only when there is an improvement in classification accuracy on the validation dataset (monitor='val_acc' and mode='max') or in loss (monitor='val_loss' and mode='min'). The weights are stored in a file that includes the weights for

best accuracy and minimum loss.The weight files obtained are downloaded to be used on the local machine.

```
[ ]  history1 = LossHistory()
     history2 = AccHistory()
     from keras.callbacks import ModelCheckpoint
     filepath1="weights.best.acc.221_model.hdf5"
     filepath2="weights.best.loss.221_model.hdf5"
     checkpoint1 = ModelCheckpoint(filepath1, monitor='val_acc',
                                   verbose=1, save_best_only=True, mode='max')
     checkpoint2 = ModelCheckpoint(filepath2, monitor='val_loss',
                                   verbose=1, save_best_only=True, mode='min')

     callbacks_list = [checkpoint1,checkpoint2,history1,history2]
```

Fig 3.10 : Code snippet for adding Model Checkpoints

3.2.4 Live Video Processing

Convolutional Neural Network is compiled on the local machine and weight files downloaded are used to initialize the model. Webcam is used to obtain live stream. This stream is used to extract frames. Haar Cascade Classifier is used to detect faces from the frames and create a list of faces having top leftmost pixel coordinates as well as height and width of face. Individual faces are converted to grayscale and resized to (48,48,1) shape. Resized image is fed to neural network for prediction. Colour scheme is used to differentiate between emotions. Emotion and a rectangle around the face is displayed.

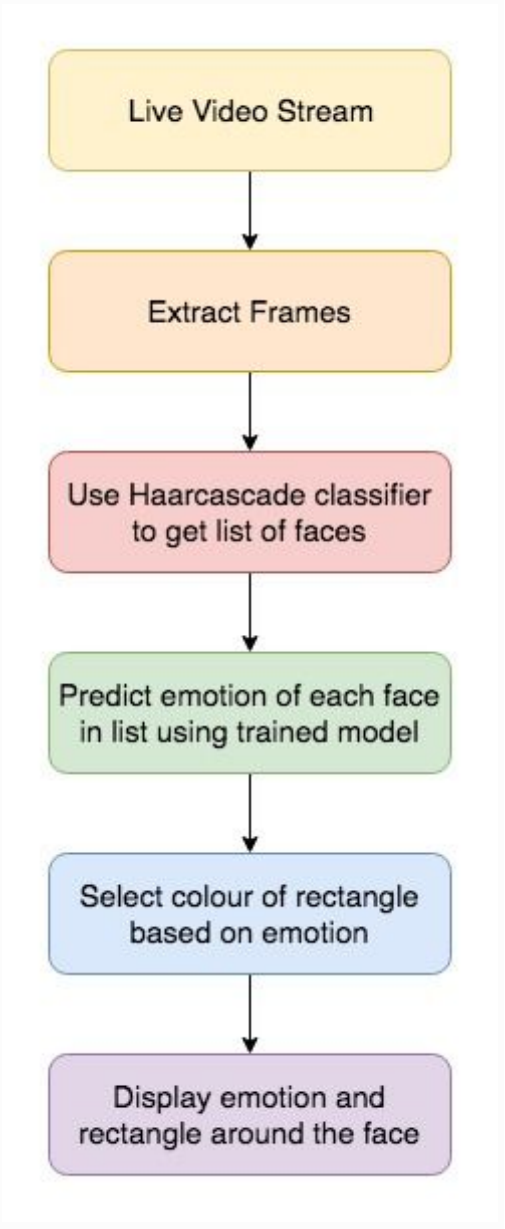

Fig. 3.11 : Flowchart of Working of Video Processing

3.3 Result Visualization and Analysis

3.3.1 Accuracy and Loss

We were able to achieve good results with our model. Our final model performed well on the test set, reaching state-of-the-art accuracy levels using CNNs.

Set	**Accuracy**	**Loss**
Training Set	84.78	0.434
Validation Set	61.52	1.170
Test Set	62.72	1.157

Table 3.1 : Accuracy and Loss on Training, Validation and Test Set

3.3.2 Accuracy and Loss Curve Analysis

Looking at graphs , we see a healthy loss curve. The loss decreases substantially in the third epoch and then slowly after that. We do see the model is learning fairly slowly in the later epochs, so there might be a way to set the learning rate to achieve more efficient learning, but by running for more epochs we were able to get the algorithm to converge nevertheless. In the Accuracy plot , we can see the training and validation accuracy as at the end of each epoch for our best model. While there is a gap between the training accuracy and the validation accuracy which suggests overfitting, we found that when we increased the regularization parameter to bring the validation accuracy closer to the training accuracy, the training performance decreased substantially and overall the validation performance was worse.

Fig 3.12 : Training v/s Validation Accuracy Graph

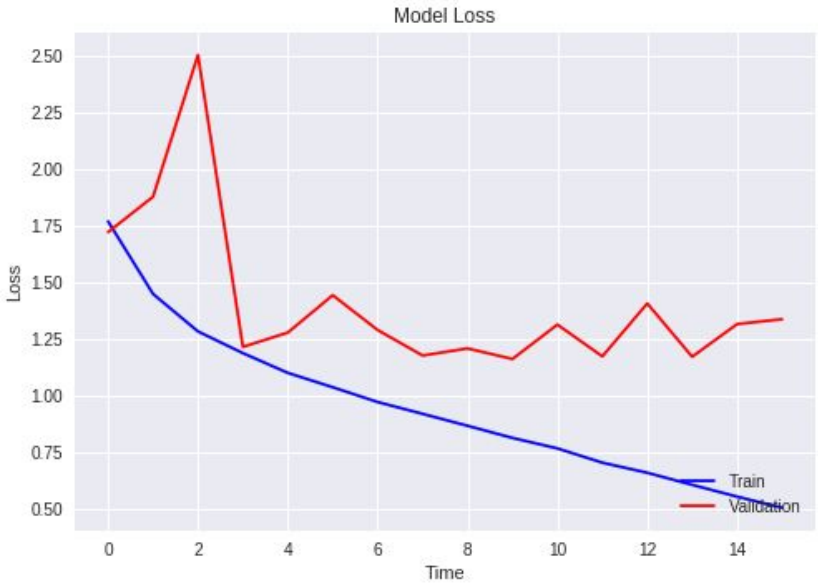

Fig 3.13 : Training v/s Validation Loss Graph

3.3.3 Dataset Images with Probability Graphs

Following figure shows Test Set images along with their true labels. Correct predictions have text color set to black while incorrect predictions have text color set to cyan. Graph shows the probability distribution of emotions as predicted by our model.

Fig 3.14 : Model Predictions on Dataset Images

3.3.4 True and Predicted Emotion Distribution

After analyzing the graph we can see that our model has high accuracy at detecting Disgust, Fear, Happy, Surprise and Neutral emotions. While model faced difficulties while classifying Angry and Sad emotions.

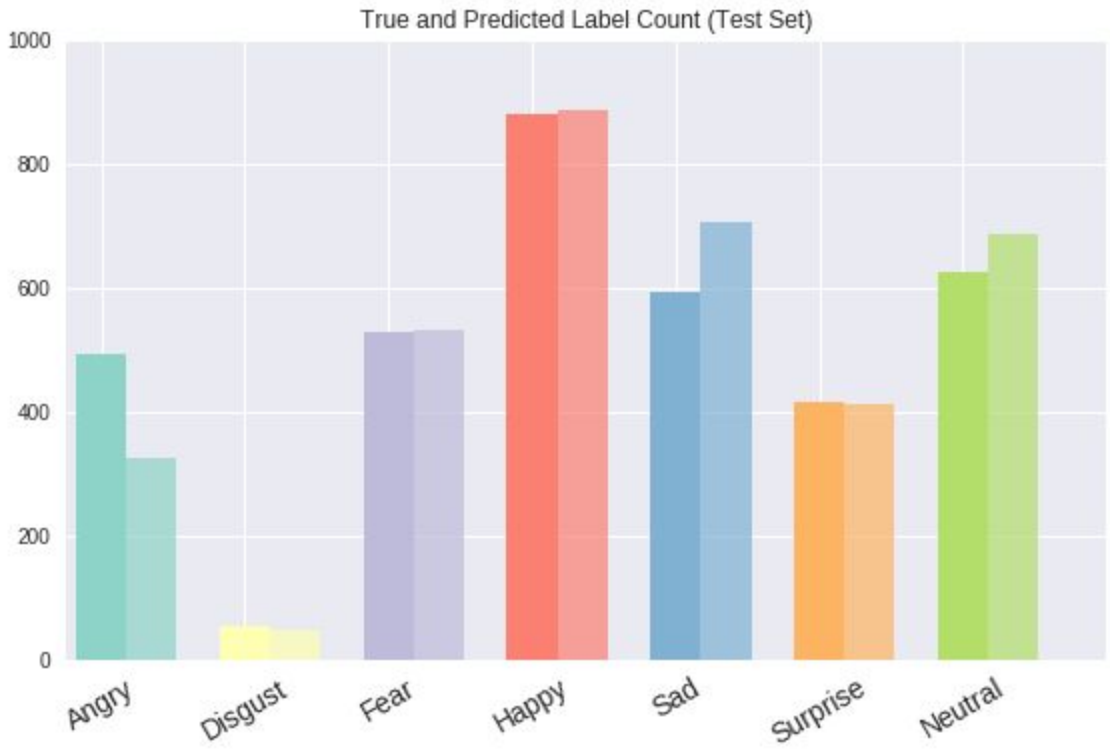

Fig 3.15 : True and Predicted Label Count of Test Set

3.3.5 Confusion Matrix

We examine the confusion matrix for the test set to understand better where misclassifications occur. We see that on the whole, our classification is fairly good, as seen by the strong presence along the diagonal. The model is most successful at classifying “happy correctly.

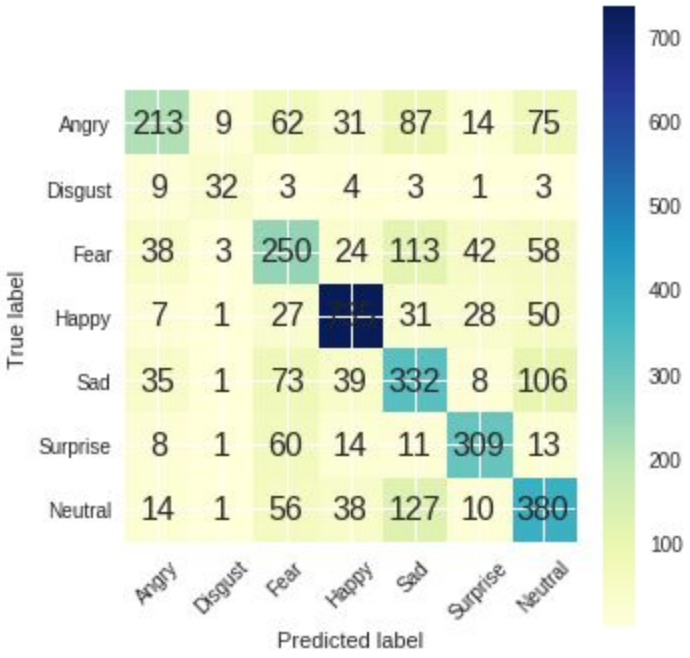

Fig 3.16 : Confusion Matrix Heat Map

3.3.6 Classification Report

True Positives (TP) - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.

True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.

False Positives (FP) – When actual class is no and predicted class is yes.

False Negatives (FN) – When actual class is yes but predicted class in no.

Precision :- It is the ratio of a number of events you can correctly recall to a number all events you recall (mix of correct and wrong recalls). It gives us the number of positive predictions divided by the total number of positive class values predicted. It is also called the Positive Predictive Value (PPV).

Precision = TP/TP+FP

Recall :- It is the ratio of a number of events you can correctly recall to a number of all correct events. It can also be called as the number of True Positives divided by the number of True Positives and the number of False Negatives. Put another way it is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate.

Recall = TP/TP+FN

F1-Score :- The F1 Score is the 2*((precision*recall)/(precision+recall)). It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall. It gives the harmonic mean of precision and recall. The scores corresponding to every class will tell you the accuracy of the classifier in classifying the data points in that particular class compared to all other classes.

	precision	recall	f1-score	support
angry	0.57	0.53	0.55	491
disgust	0.95	0.38	0.55	55
fear	0.58	0.33	0.42	528
happy	0.83	0.84	0.84	879
sad	0.45	0.62	0.52	594
surprise	0.80	0.76	0.78	416
neutral	0.57	0.65	0.60	626
avg / total	0.65	0.64	0.63	3589

Fig 3.17 : Classification Report

Based on the F1 Score , we can conclude that our model performs well on Positive Emotions such as Happy, Surprise and Neutral. F1 Score of Disgust emotion can be increased if more data is provided.

3.3.7 Misclassification Distribution

From Misclassification Distribution, we can conclude that 62% were correct predictions and around 81% were predicted correct on first or second prediction. Hence, we were able to obtain satisfactory results from out model.

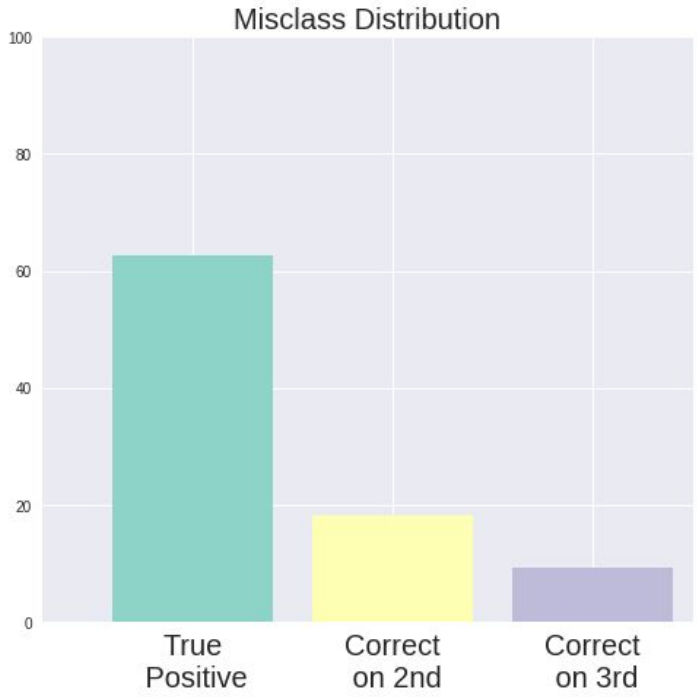

Fig 3.18 : Misclassification Distribution

3.4 Comparison with Other Architecture

3.4.1 Intuition

As Fear and Surprise Emotions have common facial features, we tried to classify them in the same class. Further, due to unavailability of large dataset for Disgust Emotion, we included it in the Fear-Surprise dataset. Hence, our model now classified emotions into 5 classes. We experienced improved accuracy on a smaller neural network.

3.4.2 Architecture Used

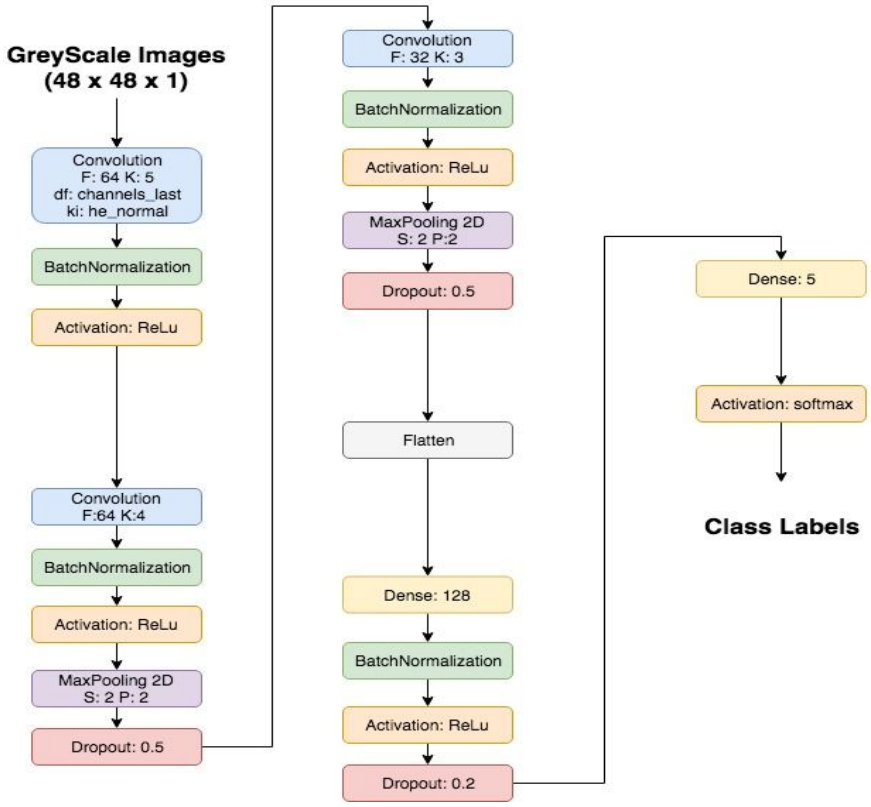

Fig 3.19 : Architecture of 5-classes Recognition Model

3.4.3 Accuracy and Loss Graphs

From the graph, we can see that it is a healthy loss curve. Also, the validation set accuracy follows training set accuracy which shows that the model is not overfitting.

Fig 3.20 : Training v/s Validation Set Accuracy Graph (Model-2)

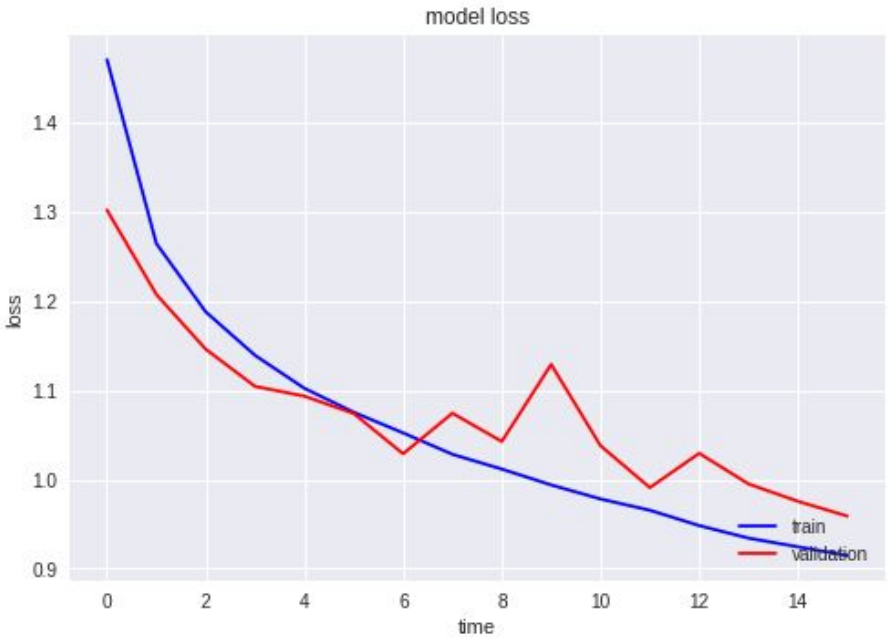

Fig 3.21 : Training v/s Validation Set Loss Graph (Model-2)

3.4.4 Result Comparison

Metrics	Model 1	Model 2
Validation Set Accuracy	61.52	62.94
Test Set Accuracy	62.72	64.59
Classification Classes	7	5
Trainable Parameters	4,540,039	358,533
Total Parameters	4,545,927	359,173

Table 3.2 : Performance Comparison of Model-1 and Model-2

3.4.5 Conclusion

Challenges in Representation Learning: Facial Expression Recognition Challenge was hosted on Kaggle.com in 2013. The best solution had an accuracy of 71% and top 10 submissions had an accuracy of 60%. We were able to achieve an accuracy of 64.59% on the test set by using model 2. Hence, we were able to reach a state-of-the-art test accuracy.

4. User Interface

User interface of the project is made using a web application. The web application is developed using Ruby as a language and Rails as the platform. Rails is a model–view–controller (MVC) framework, providing default structures for a database, a web service, and web pages. It encourages and facilitates the use of web standards such as JSON or XML for data transfer, and HTML, CSS and JavaScript for display and user interfacing.
Rails emphasizes the use of other well-known software engineering patterns and paradigms, including Convention-over-Configuration (CoC), Don't Repeat Yourself (DRY), and the active record pattern.

4.1 Implementation of Web Application

4.1.1 Front-end

For implementing front-end of the web application, technologies like HTML/CSS are used. As per the project requirements, we designed a single page web application. For implementing this, concepts like layouts and partial files are used.

Layout

The layout of the application consists of two buttons, a text description and a fully-responsive user interactive design. The layout is developed in such a way that user can initiate and terminate the process any number of times as per the requirements.
The user is not required to reload the page again and again because both buttons are provided on the same web page and neither starting nor stopping the process requires a page reload.

Buttons

The web application consists of two buttons: Start button and Stop button. For implementation of the buttons styling, Cascading Style Sheets mechanism (CSS) is used. Different selectors, properties and values are used to provide a very interactive design to user.
On clicking Start button, user can initiate the process. It will trigger the webcam to start and the python script starts running at the backend.
Similarly, on clicking the Stop button, user can terminate the process. It triggers a few commands to run on terminal which terminates the running processes.

4.1.2 Back-end

For implementing the back-end of the web application, Ruby language was used. Ruby acts as a back-end language for web development using Ruby on Rails. The application required the use of controllers and views for its working.

Routes

Three routes are defined for the backend implementation: root_path, start_path and stop_path. The URL's are defined in 'routes.rb' file. Corresponding actions are defined in the 'home_controller.rb'.

Controller

Controller of the application defines the actions associated with different routes. Two actions were defined to implement start and stop functions.

Start

For implementing the start action, process spawning is used. The python script is executed and the results are displayed. As soon as the user clicks 'Start' button, control reaches 'start' action and the process is spawned using 'Process.spawn' command.

Stop

For implementing the stop action, 'pidof' command is used to fetch the pid of the running python script. Then all the processes and subprocesses running using these pid's are terminated using 'kill' command.

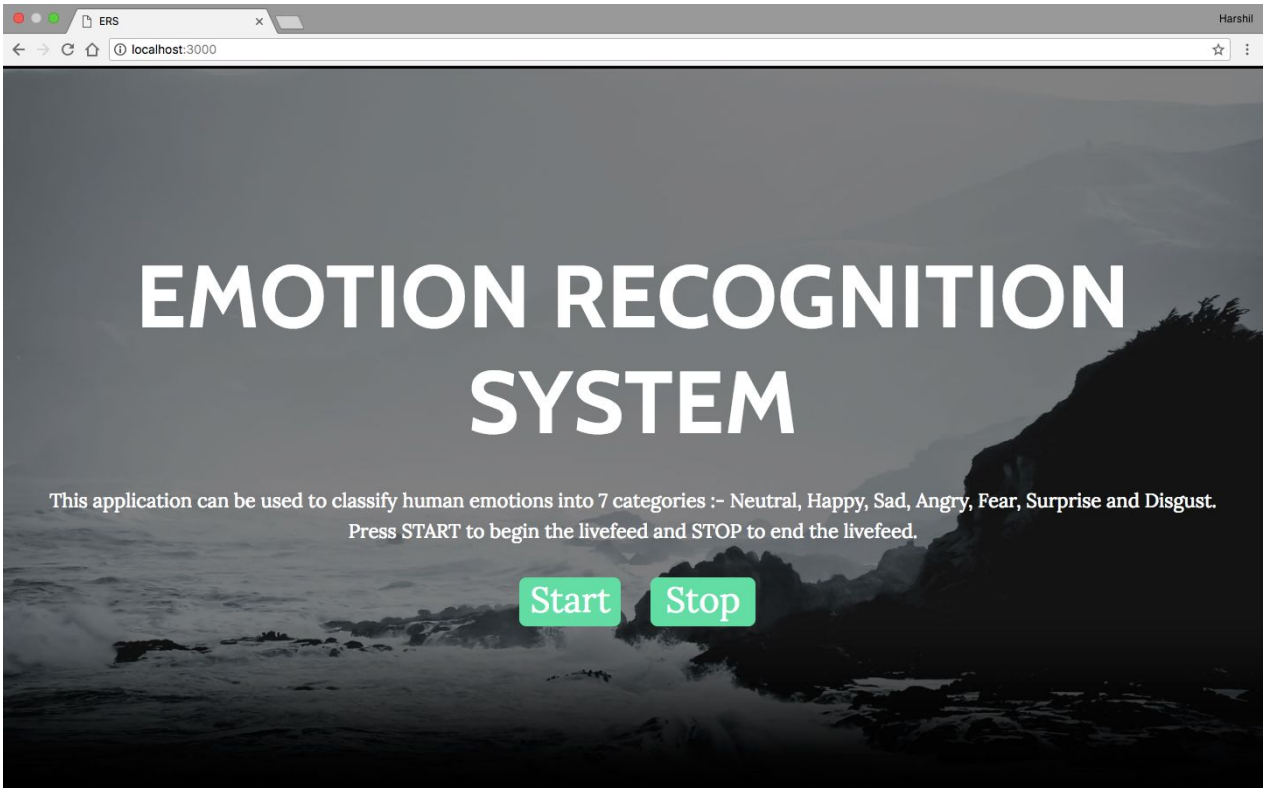

Fig 4.1 : Web Application

5. Conclusion and Future Work

5.1 Conclusion

In this project, we sought to classify the image of a face into the seven basic human emotions. We developed and experimented with the architecture of a deep convolutional neural network ourselves, and performed a hyperparameter search to optimize our results. We compared different architectures to classify emotions and we were able to reach the state-of-the-art test accuracy of approximately 64.59%.

We have described a holistic, non intrusive approach to emotion detection, by checking for user's facial expressions. These are powerful measures, even for low image resolution and in-the-wild circumstances such as bad illumination, facial expressions, non-frontality etc.

However, we believe if we addressed the overfitting of the training data, we could reach even higher test accuracies. The objectives defined in the thesis were accomplished. We also developed a web application to demonstrate our project and provide interactive interface to the user.

5.2 Future Work

5.2.1 Build a REST API and add functionalities to Website

REST API can be built that finds human faces within images and make prediction about each facial emotions. User must be able to paste the url of an image or drag-and-drop an image file. In addition, user must have the option to have the API return the image with annotated faces and cropped thumbnail of each face. The API must return the probabilities of emotions for each face (indexed) and an unique ID for each image in json format.

5.2.2 Extend to Bimodal Classification

While facial expressions contributes for 55% to the effect of the speaker's message , vocal part contributes to 38%. Ensemble Learning can be used to obtain predictions by combining two different classifiers - images and audio. This will increase the accuracy of emotion detection as only body movements aren't used in the prediction process.

5.2.3 Transfer Learning

Transfer Learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. As for our problem, model trained on ImageNet dataset can be fine tuned on other dataset and results can be analyzed.

References

[1] Nicu Sebe , Michael S. Lew , Ira Cohen , Ashutosh Garg , Thomas S. 59 Huang Emotion Recognition Using a Cauchy Naive Bayes Classifier ICPR, 2002

[2] Cohen Nicu Sebe , Larry Chen, Ashutosh Garg , Thomas Huang , Facial Expression Recognition from Video Sequences: Temporal and Static Modeling Computer Vision and Image Understanding (CVIU) special issue on Face recognition (http://www.ifp.uiuc.edu/~iracohen/publications.htm)

[3]Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49, October, 2007.

[4]Kaggle competition: Challenges in representation learning: Facial expression recognition challenge, 2013.

[5]Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., Taylor, J.G. Emotion recognition in human-computer interaction. Signal Processing Magazine, IEEE, Volume: 18, Issue: 1, Jan 2001. Pages: 32 – 80

[6] Pantic, M., Rothkrantz, L.J.M. Toward an affect-sensitive multimodal human-computer interaction. Proceedings of the IEEE , Volume: 91 Issue: 9 , Sept. 2003. Page(s): 1370 – 1390.

[7]Dellaert, F., Polzin, T., Waibel, A. Recognizing emotion in speech. Spoken Language, 1996. ICSLP 96. Proceedings. Fourth International Conference on, Volume: 3, 3-6 Oct. 1996. Pages: 1970 - 1973 vol.3.

[8]Roy, D., Pentland, A. Automatic spoken affect classification and analysis. Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on , 14-16 Oct. 1996. Pages: 363 – 367

[9]Lee C. M., Narayanan, S.S., Pieraccini, R. Classifying emotions in human-machine spoken dialogs. Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International. Conference on , Volume: 1 , 26-29 Aug. 2002. Pages:737 - 740 vol.1

[10] Lee, C. M., Yildirim, S., Bulut, M., Kazemzadeh A., Busso,C., Deng, Z., Lee, S., Narayanan, S.S. Emotion Recognition based on Phoneme Classes. to appear in Proc. ICSLP'04, 2004.

[11] Nwe, T. L., Wei, F. S., De Silva, L.C. Speech based emotion classification. Electrical and Electronic Technology, 2001. TENCON. Proceedings of IEEE Region 10 International Conference on, Volume: 1 , 19-22 Aug. 2001. Pages: 297 - 301 vol.1

[12] Lee C. M., Narayanan S.S. Towards detecting emotions in spoken dialogs. IEEE Trans. on Speech & Audio Processing, in press, 2004.

[13] Boersma, P., Weenink, D., Praat Speech Processing Software, Institute of Phonetics Sciences of the University of Amsterdam. http://www.praat.org

[14]Touchy Feely: An Emotion Recognition Challenge Authors : Dhruv Amin Stanford University dhruv92@stanford.edu, Patrick Chase Stanford University pchase@stanford.edu, Kirin Sinha Stanford University ksinha@stanford.edu

[15] Facial Expression Recognition for wild images with analysis from Saliency maps Authors: Priyanka Rao Stanford University prao96@standford.edu, Ling Li Stanford University lingli6@stanford.edu

[16] Deep Convolutional Neural Networks for Tiny ImageNet Classification Hujia Yu Stanford University hujiay@stanford.edu

[17]De Silva, L.C., Ng, P. C. Bimodal emotion recognition. Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, 28- 30 March 2000. Pages: 332 – 335.

[18]Chen, L.S., Huang, T. S., Miyasato T., and Nakatsu R. Multimodal human emotion / expression recognition, in Proc. of Int. Conf. on Automatic Face and Gesture Recognition, (Nara, Japan), IEEE Computer Soc., April 1998

[19]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.

[20]Yoshitomi, Y., Sung-Ill Kim, Kawano, T., Kilazoe, T. Effect of sensor fusion for recognition of emotional states using voice, Workshop on, 27-29 Sept. 2000. Pages: 178 – 18.