Sentimental Analysis of Text and Facial Expression Recognition

Project Report

submitted towards the partial fulfilment of

the requirements for the award of the degree of

Bachelor of Technology

in

Electronics & Communication Engineering

Submitted by

Anudeep Kumar Reddy Mummadi **15116030**

Rajendra Thottempudi **15116063**

Under the guidance of

Professor Debashis Ghosh

Department of Electronics and Communication Engineering
Indian Institute of Technology Roorkee
Roorkee

May 2019

STUDENTS' DECLARATION

We declare that the work presented in this thesis with title "**Sentimental Analysis of Text and Facial Expression Recognition**" towards the partial fulfilment of the requirements for the award of degree of **Bachelor of Technology** in **Electronics and Communication Engineering** submitted to the Department of Electronics and Communication Engineering, Indian Institute of Technology, Roorkee is an authentic record of our own work carried out during the period from August 2018 to April 2019 under the supervision of **Dr. Debashis Ghosh**, Dept. of Electronics and Communication Engineering, IIT Roorkee.

PLACE: .....

DATE: .....

.....	.....
Anudeep Kumar Reddy Mummadi	Rajendra Thottempudi
15116030	15116063

CERTIFICATE

This is to certify that the declaration made by the students is correct to the best of my knowledge and belief.

DATE: ....

Dr. Debashis Ghosh

Acknowledgements

First and foremost, we would like to express our sincere gratitude towards our guide **Dr. Debashis Ghosh**, Professor, Department of Electronics and Communication Engineering, IIT Roorkee for his ideal guidance throughout the period. His advices, insightful discussions and constructive criticisms certainly enhanced our knowledge and improved our skills. His constant encouragement, support and motivation have always been key sources of strength for us to overcome all the difficult and struggling phases.

We would also like to thank all the professors and research scholars who have been always helpful and understanding.

We would also like to thank **Department of Electronics and Communication Engineering, IIT Roorkee** for being a perfect catalyst to our growth and education over these past four years.

We also extend our gratitude to all our friends, for keeping us motivated and providing us with valuable insights through various interesting discussions.

Abstract

Emotions play an important role in effective as well as successful communication among humans. With the enormous growth of social media through various applications such as twitter, Facebook etc. people are expressing their emotions on several issues through their news feeds. For a machine which is situated remotely and wishes to understand customer emotions on a particular topic or a particular product it is necessary to perform text mining. Sentimental Analysis is an ongoing field of research in text mining and is a computational treatment of sentiments, opinions and subjectivity of text. In human-human interaction facial emotions play an integral part and act as a part of nonverbal communication and as a primary means of conveying social information. With the increase of Automation, it is necessary for a system to understand human facial expressions for effective communication. Hence, machine based real time facial expression recognition has a huge range of applications in facial animation, human-computer interaction, entertainment and psychological studies. This report presents the results of our work on understanding sentimental analysis of text and facial expression recognition. Maximum Entropy Classifiers, Convolution Neural Networks and Long Short Term Memory Networks are used to capture the sentimental polarity of a tweet. We implemented a real time Convolution Neural Network model which accomplishes the task of gender classification and facial emotion classification simultaneously. This model is compared against state-of-the-art models on FER-2013 emotion dataset. Our results are competent and are tantamount for the scale of our model.

Table of Contents

List of Figures

Figure **Page**

CHAPTER 1

Introduction

1.1 Overview

Analysing and Understanding human emotions is an integral part of developing the human - Computer interaction. With the development of deep learning technologies and the beginning of algorithms such as Recurrent Neural Networks, Convolution Neural Networks which are quite effective to understand various patterns especially in images, the research in the fields of Sentimental Analysis and Facial Emotion Recognition spiked. The purpose of Sentimental Analysis is to understand user behaviour on a large number of issues, to understand customer reviews on a product, to develop recommendation systems etc.

1.2 Sentimental Analysis

Also known as Opinion Mining, Sentimental Analysis is a field under the umbrella of Natural Language Processing that builds systems which help extract and identify opinions from text. Besides identifying the opinion, various attributes are also extracted from the expression. For example:

- Polarity: Whether the speaker expresses a positive or negative opinion.
- Subject: The topic that is being discussed about.
- Opinion Holder: The person or entity that expresses the opinion.

Opinion Mining can be considered and modelled as a classification problem with two sub- problems that need to be solved:

- Subjectivity Classification: Classifying a sentence either as subjective or objective.
- Polarity Classification: Classifying a sentence as either expressing a negative, neutral or a positive opinion.

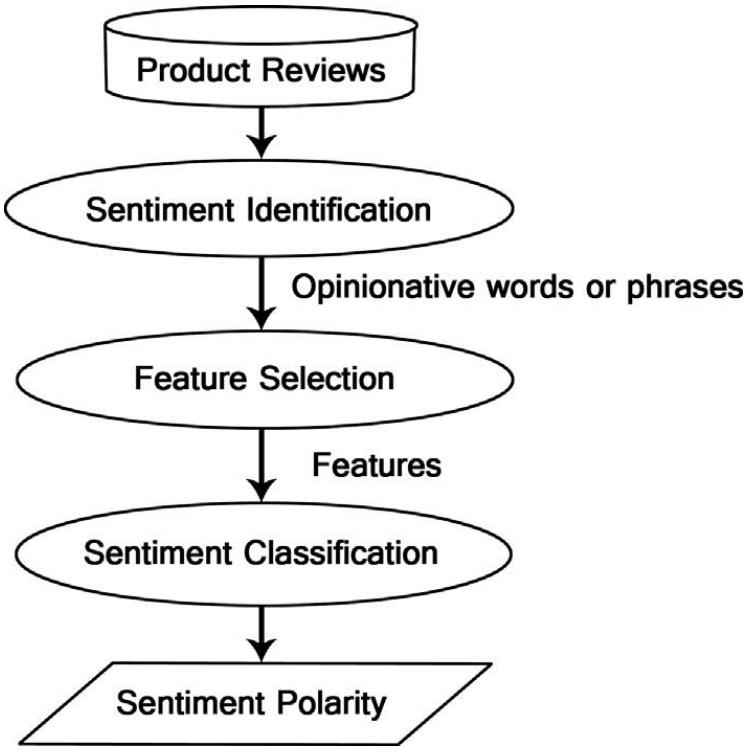


Figure 1.1 Sentiment analysis process on product reviews

In an opinion, the entity that the text is about can be an object, its features, its attributes, its components or its aspects. It could as well be a product, an individual, an event, a service, a topic or an organisation.

- Direct Opinion: These kind of texts give an opinion about an entity directly.

  "The image quality of a Nokia Camera is Unbearable."

- Comparative Opinion: In Comparative texts, an opinion is expressed by comparing two entities.

  "The image quality of a Nikon Camera is better than that of a Nokia Camera"

- Explicit Opinion: An opinion which is explicitly expressed in a subjective manner.

  "The voice quality of this speaker is great."

- Implicit Opinion: An implicit opinion is an opinion which is implied in an objective manner on a subject.

"The heater broke within a week."

Types of Sentimental Analysis:

Sentimental Analysis varies from systems that notice feelings and emotions (angry, sad, happy) to systems that detect polarity (neutral, positive, negative) or those that determine intentions (e.g.: not interested/interested).

1. ***Feelings and Emotion Detection:***

   Emotions such as Happiness, frustration, sadness and angry that are expressed via text are to be differentiated using these systems. Major portion of the emotion detection systems resort to lexicon based approaches (i.e. lists of words and corresponding emotions) and complex machine learning algorithms. One of the disadvantage of using lexicon based approach is that words might not always convey the actual meaning of the sentence which the user tends to convey. Consider two examples where the word kill is used for angry and happiness as well.

   "*I will kill you*."

   "*You are killing it.*"

2. ***Intent Analysis:***

   An intent analysis system tries to detect what a person wishes to do with a text rather than what the text says.

   "*I wish to understand how to change the cartridge.*"

   "Can you help me fill out the form?"

   Inferring the question in the first statement and the request in the second statement is easy for humans which isn't the case for machines. Sometimes the intended action can be inferred from the text but sometimes it requires contextual knowledge.

3. ***Aspect Based Sentiment Analysis:***

   While analysing the reviews of a product people would be more interested to know the reviews about various features of the product specifically instead of the product as a whole. Systems which try to implement that come under Aspect Based Sentiment Analysis.

   "The *battery life of this mobile is great."*

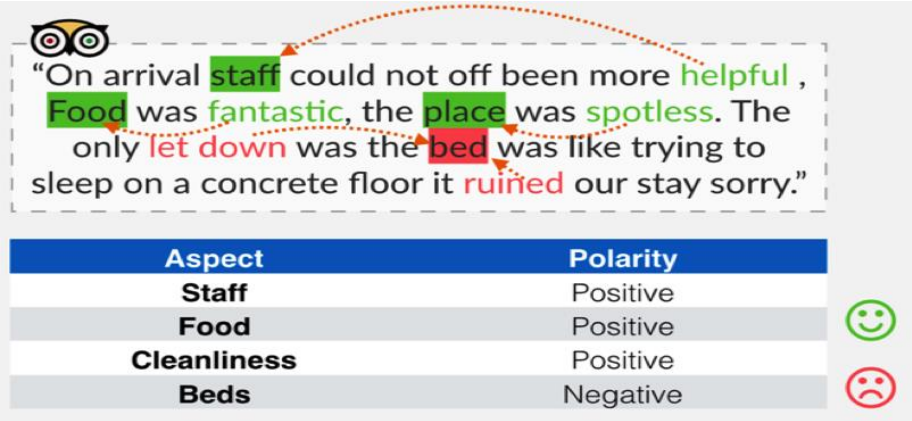


Figure 1.2 Example showing Aspect Based Analysis

1.3 Facial Expression Recognition

Facial Expression Recognition is an Image Classification problem set among the broader field of Computer Vision. In a general image classification problem, we will have a set of classes in which a particular class has to be allotted algorithmically for an input image. In case of Facial Expression Recognition, classes are a set of emotions and human faces are the input images.

Any machine learning problem requires a training set from which the ML model will be trained in order to assign weights and increase the accuracy. In this case the training dataset comprises of images with various facial expressions each labelled with a specific emotion class that it belongs to. Once the model is trained it is then tested on the test or real-time data. Angry, Disgust, Sadness, Happiness, Neutral, Fear and Surprise are a standard set of seven classifications which are often used.

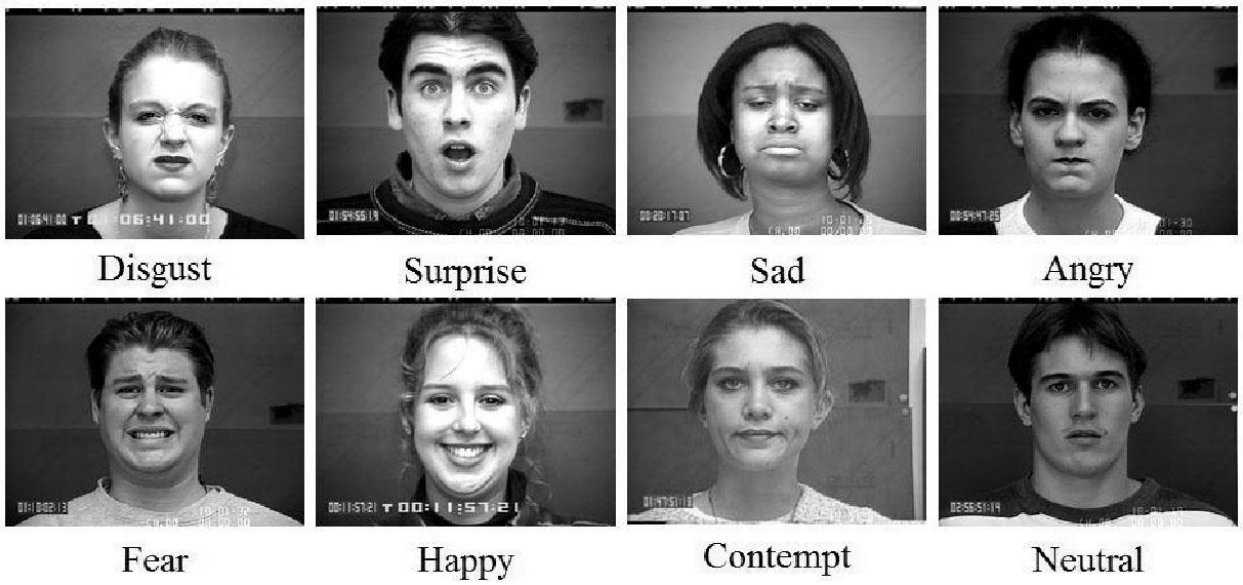


Figure 1.3 A selection of Labelled Images for Expression Analysis

An image for a machine is simply a matrix of pixel values. In order to identify an expression in an image the machine has to understand numerical patterns that are related to each and every expression and search for those patterns in any new image which the machine has to classify. These patterns can be variable, and difficult to identify for various reasons.

Several human emotions can be distinguished only by subtle differences in facial patterns, with emotions like anger and disgust often expressed in very similar ways. Each person's expressions of emotions can be highly idiosyncratic, with particular quirks and facial cues. In the photographs which are to be classified, the orientation and positions of people's heads varies a lot. For these types of reasons, Facial Expression Recognition is more difficult than most other Image Classification tasks.

Most of the typical Facial Expression Recognition systems typically use image recognition, preprocessing of image and feature extraction methods. These methods are then followed by training our architecture on a training dataset. As a result of training, a model is generated which is capable to assign emotion categories for a newly provided image.

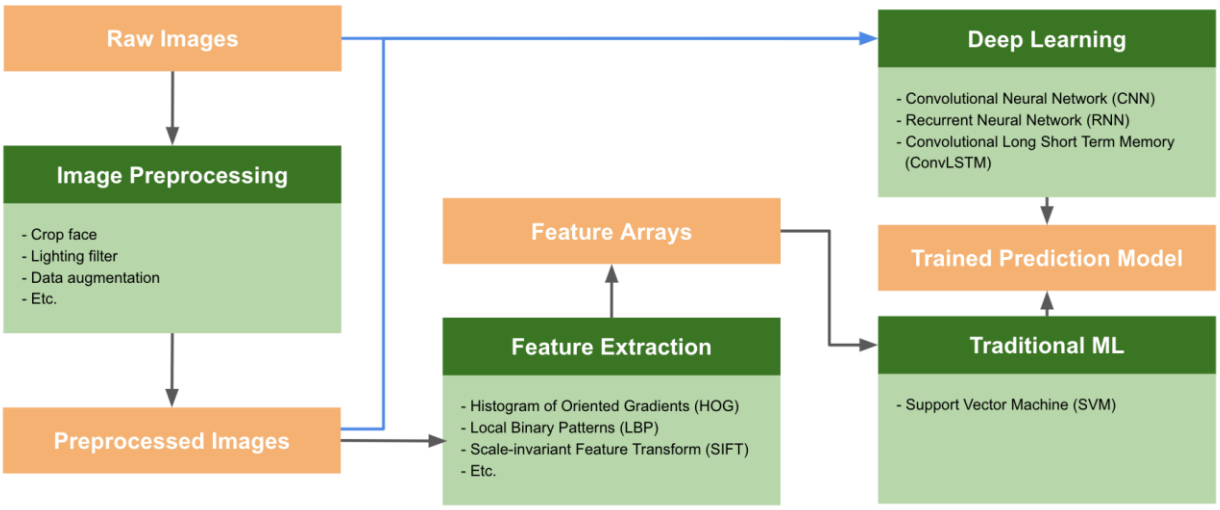


Figure. 1.4: Commonly used Facial Express Recognition System Architectures

Various image transformations such as cropping, filtering and scaling images are used in the preprocessing stage. It is often used to accentuate relevant image information, like cropping an image to remove a background. Since the datasets that are available for training do not contain sufficient data, these methods are often used to augment a dataset i.e. creating several versions from a single image by cropping or using some other transformations.

The major crux of the feature extraction stage is to find more descriptive parts of an image which often means finding relevant information which can be used to indicate and differentiate a specific class, such as edges, colours or textures.

The architecture which is used in the training stage determines various combinations of layers which are used to feed each other in the proposed neural network. These architectures are designed keeping in

mind the composition of images preprocessing stages and the feature extraction stage in mind. This is necessary because some architectural components work better with others when applied separately or together.

1.4 Research Objective

The objective of this work is to provide a comprehensive report of the major breakthroughs in the fields of Sentimental Analysis and Facial Expression Recognition and alongside analyse the various factors that affect the performance of trained models in the field. The main objective lies in studying the advantages and disadvantages of these methods and architectures and leverage this information to try and improve over them.

Deep learning techniques such as Convolution Neural Networks have clearly defeated their competition in these fields and are progressively getting better at the task. Although these methods perform better than any other model, they are far from near human level performance and this triggered our research in the subject.

CHAPTER 2

Literature Survey

In the fields of Computer-Vision and Human-Computer interaction, Sentimental Analysis and Facial Expression Recognition are two areas which have experienced and enormous amount of growth in the past few years. A lot of work has been published over the many years which lays foundation for the modern techniques. We can't do justice to all the work that we came across, hence choose to highlight only those methods which we believe are ground-breaking and have helped the Computer Vision research community around the world.

2.1 Classical Approaches

2.11 Naive Bayes Classifier

Naive Bayes is one of the simplest techniques for constructing classifiers. It consists of a family of algorithms which will be used to construct and train classification models. Given a class variable, every Naive Bayes classifier assumes that the value of one feature is completely independent of that of any other feature.

Bayes Theorem:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Where y is the class variable and X represents the parameters/features.

X is given as:

$$X = (x_1, x_2\ , x_3, \ldots, x_n)$$

Where $x_1, x_2\ , x_3, \ldots, x_n$ represents the features.

$$P(y|x_1, x_2\ , x_3, \ldots, x_n) = \frac{P(x_1|y)P(x_2\ |y)P(x_3|y)\ldots P(x_n|y)P(y)}{P(x_1)P(x_2)P(x_3)\ldots P(x_n)}$$

Since the denominator remains the same, we can write the above equation as:

$$P(y|x_1, x_2\ , x_3, \ldots, x_n)\ \alpha\ P(y)\ \prod_{i=1}^{n} P(x_i|y)$$

Now this model is converted into a classifier with the help of a decision rule i.e. to pick the hypothesis which is most probable. Thus a Bayes Classifier finds and assigns the classy with maximum probability:

$$y = argmax_y P(y)\ \prod_{i=1}^{n} P(x_i|y)$$

2.12 Maximum Entropy Classifier

Maximum Entropy Classifier is one of the probabilistic classifier techniques which belongs to the class of exponential techniques and it differs from the Naive Bayes classifier by not assuming that all the features are conditionally independent. Maximum Entropy Classifier is also based on the principle of maximum entropy i.e. from all the models which are generated to fit our training data, it selects the one with the maximum entropy. A variety of text classification Problems such as sentiment analysis, topic classification, language detection can be solved using this classifier.

Whenever a system is given with some data and is asked to make a decision, it would start finding various attributes or features among the data in order to take the most logical decision that is possible. The weightage that is given to these features vary depending on their importance. Maximum Entropy Classifier is a method which uses these attributes or features into a mathematical model based on the output of which the system takes a decision.

In this model, a weight is applied for each attribute that is found in the data and finally all these weights are added up. The weighted sum is then normalized to provide a fractional value. The machine then used this fraction to take the required decision and classify the data.
Assuming word-level features:

1. For each word w and class c ∈ C, define a joint feature f (w, c) = N where N is the number of times that w occurs in a document in class c.
2. Via iterative optimization, assign a weight to each joint feature so as to maximize the log-likelihood of the training data.
3. The probability of class c given a document d and weights λ is

$$P(c|d,\lambda) = \frac{exp\ \sum_i\ \lambda_i f_i(c,d)}{\sum_{c' \epsilon C} exp\ \sum_i\ \lambda_i f_i(c,d)}$$

Estimating the Optimal λ parameters can be easily done using an iterative scaling algorithm.

2.13 Lexicon Based Approach

Lexicon - Based approach to sentiment analysis does not require any training dataset to train the model. In this approach the sentiments that are conveyed through words are inferred on the grounds of the word polarity. In case of a statement or a document, the polarities of the individual words that compose the document conjointly convey the sentiment of the document or sentence. Therefore, the polarity of a sentence or a document is the collective total of the polarities of individual words.

A predefined list of words where each word in associated with some particular sentiment will be used in this approach.

Dictionary Based Methods: A lexicon dictionary is used to find out the sentiment of any particular word.

Corpus Based Methods: A large corpus of words is used and other opinion words can be found within the context based on syntactic patterns.

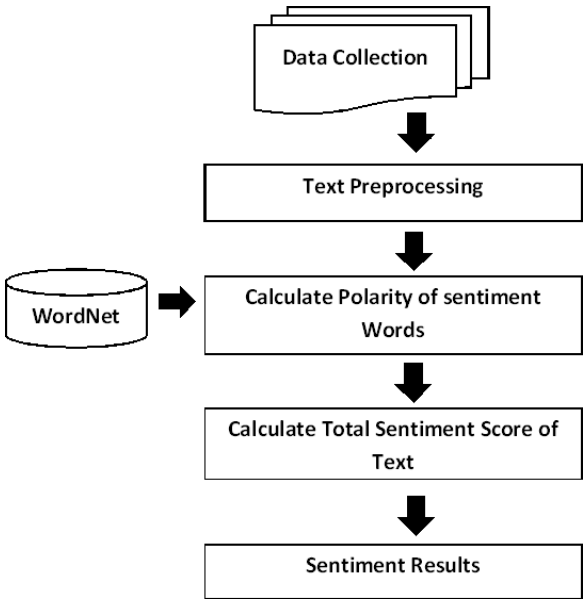


Figure 2.1: Sentiment Analysis Methodology using Lexicon Based Approach

2.14 Support Vector Machines(SVM)

The objective of a Support Vector Machine algorithm is to find an optimal hyperplane in the N Dimensional Space where N represents the number of features, such that we are able to classify all the data points in space.

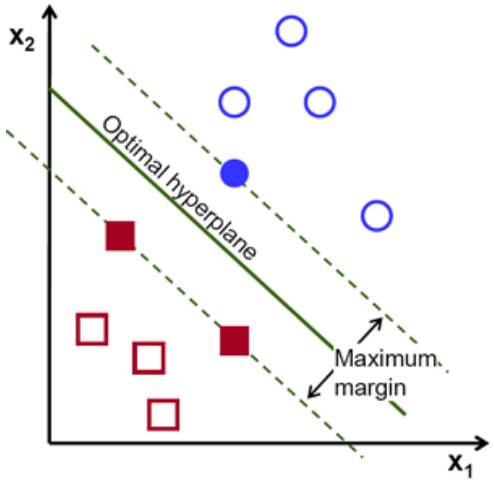


Figure 2.2: Optimal Hyperplane

The hyperplane has to be chosen such that it has the maximum margin i.e. maximum distance between data points of different classes.

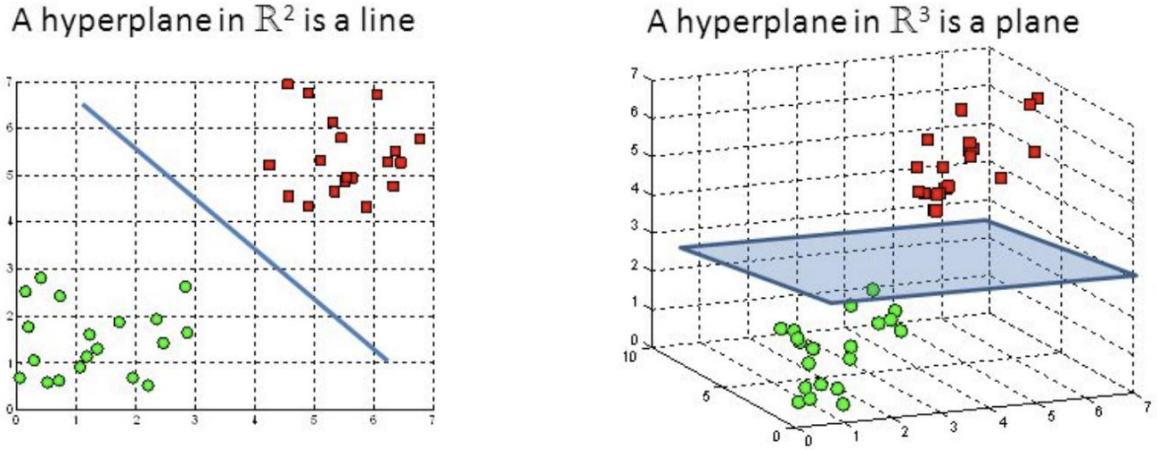


Figure. 2.3: Hyperplanes in 2D and 3D feature space

The loss function which is used to maximize the distance between the data points and hyperplane is Hinge loss, the mathematical representation of which is as follows:

$$c(x, y, f(x)) = (1 - y) * f(x)\ when\ y * f(x) \leq 1\ else\ 0$$

I.e. the cost is zero if the predicted value and actual value are of the same polarity.

Adding Regularization parameter to the cost function, the purpose of which is to balance the loss and margin maximization, the cost function becomes:

$$min_w \lambda ||w||^2 + \sum_{i=1}^{n} (1 - y_i < x_i, w >)_+$$

Using the gradients of the cost function, we can update the weights during the training of the model. Once our model is trained on sufficient dataset, it classifies any new data point.

2.2 Deep Learning Approaches

2.2.1 Convolutional Neural Networks (CNN)

This architecture belongs to the class of feed-forward neural networks that has been proven to be successful for images because of their property of locational invariance in most tasks. Number of weights in a CNN are significantly fewer in comparison to a fully connected layer (because of parameter sharing) which make them very attractive for generic use.

A deep Conv-net is capable of capturing features like translational and rotational invariance. Convolutions are largely used because of the following properties:

Parameter sharing: A feature detector (such as edge detection) that's useful in one part of the image is probably useful in another part of the image.

Sparsity of connections: In each layer, each output value depends only on a small number of inputs.

Fully connected layers are a class of feed forward neural networks that are computation intensive making them impractical to learn in deeper settings or to high dimensional data like images. The convolution operation on contrary has fixed number of parameters, invariant of input image size because of its parameter sharing property. This update plays a significant role in solving the vanishing or exploding gradients problem in training deeper networks using back-propagation.

Kernel Size(K): The kernel size defines the field of view of the convolution. A common choice for 2D is 3, that is 3×3 pixels.

Stride(S): The stride defines the step size of the kernel when traversing the image. While its default is usually 1, we can use a stride of 2 for down sampling an image.

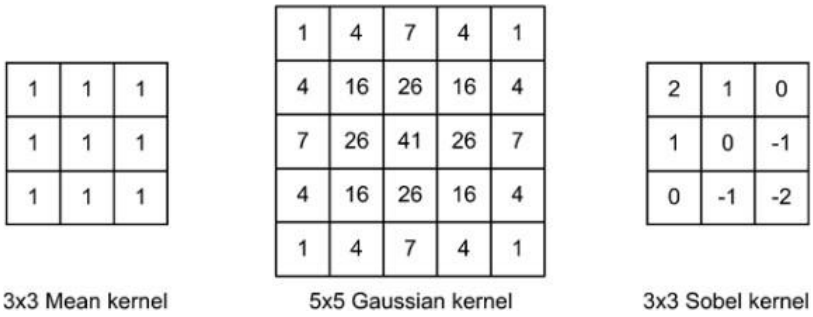


Figure 2.4: Different Kernel Maps

Padding(P): The padding defines how the border of a sample is handled. A padded convolution will keep the spatial output dimensions equal to the input, whereas unpadded convolutions will crop away some of the borders if the kernel is larger than 1.

Input & Output Channels: A convolutional layer takes a certain number of input channels(I) and calculates a specific number of output channels(O).

Input channels can be denoted as (Nh, Nw, Nc). Output channels size depends on the number of filters (Nc), Kernel size(K), Stride(S) and Padding(P).

Output channels can be denoted as

$$N_{hout} = \frac{(N_{hin} + (2 * P) - K)}{S} + 1$$

$$N_{wout} = \frac{(N_{win} + (2 * P) - K)}{S} + 1$$

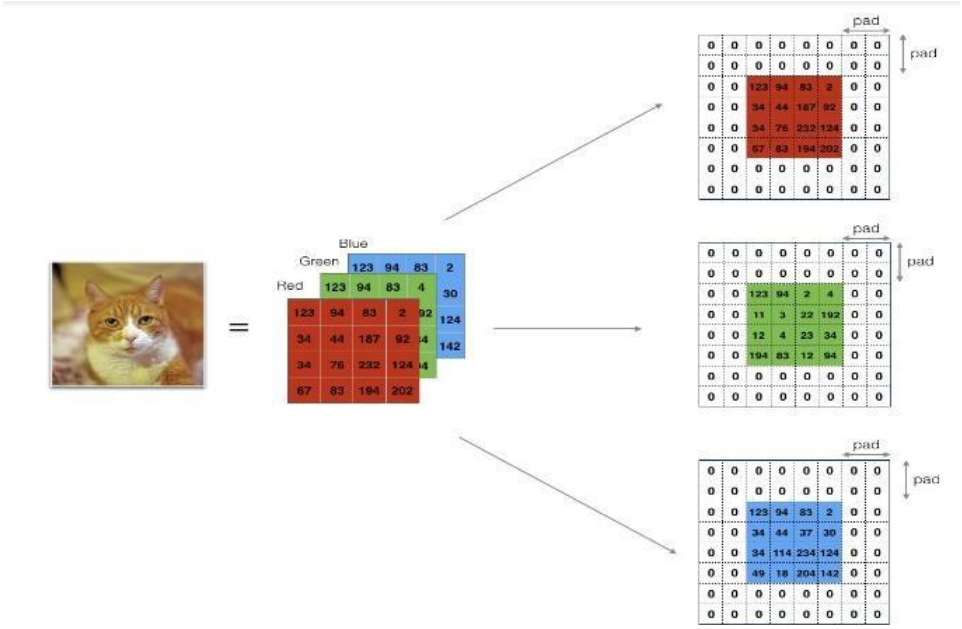


Figure 2.5: Demonstration of padding for three channels of an image

There are few operations and terms that are involved in the Convolutional Neural Network.

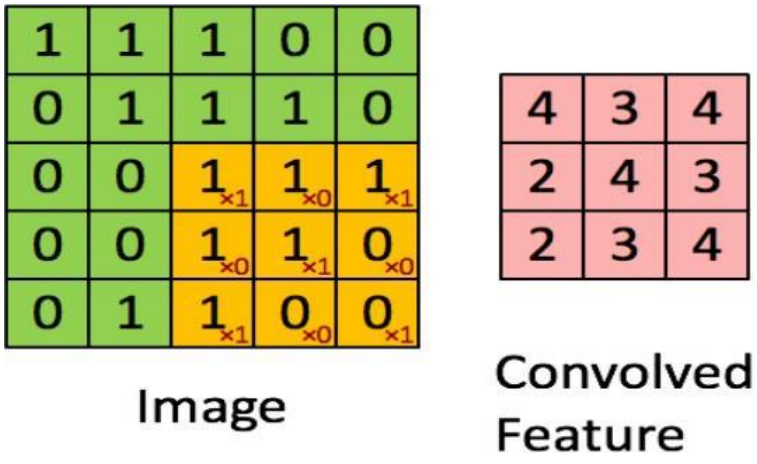


Figure 2.6: Image and it's convolved output with filter known

Pooling: Is an effective method to reduce the dimensions of a layer. It is generally used to keep decrease the dimensionality or prevent it from increasing.

Max pooling: This is the most used pooling technique, that passes the maximum of the neuron inputs over its local receptive field.

Average pooling: Average pooling is a less frequently used pooling method that takes the average over inputs in its receptive field.

Building Blocks:

A CNN architecture is formed by a stack of distinct layers that transform the input volume into an output volume through a differentiable function. A few distinct types of layers are commonly used are discussed below:

ReLU Layer:

ReLU stands for Rectified Linear Units. This layer can be considered as a piecewise defined function of the form, f (x) = max (0, x). Its differential function is bounded and remains 0 for all the negative inputs which is a very desirable property for deep networks over images. This activation function is proven to be faster than others at achieving convergence for images.

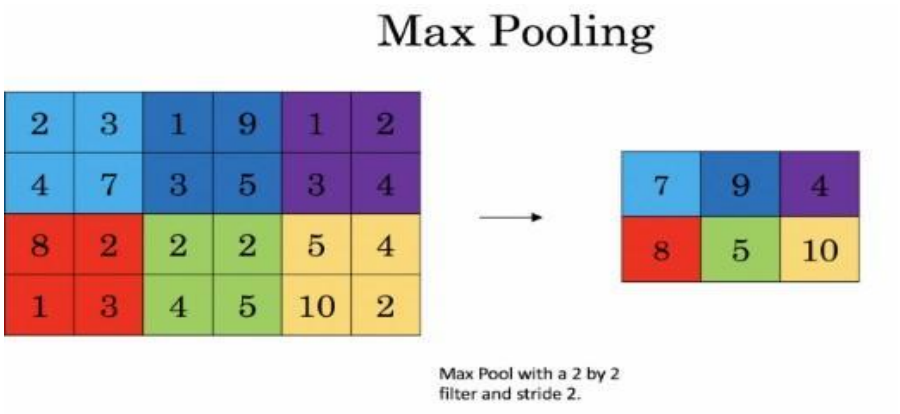


Figure 2.7: Demonstration of max pooling with a demo matrix

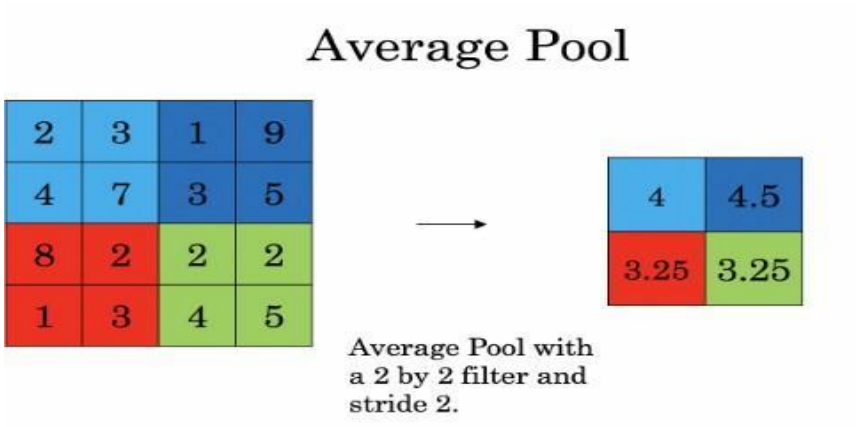


Figure 2.8: Demonstration of average pooling on a demo matrix

Batch Normalization Layer:

As weights, later or deeper in your network more robust to changes to weights in earlier layer of the neural network. It is effected to "Co-variance Shift". So for this problem Batch Normalization is used for activations of each layer.

So this layer just applies Batch Normalization to the previous layer and output the normalized values of same dimensions as previous layer. Generally, Normalization applied on Convolution output and then sent to ReLU layer.

$$\mu = \frac{1}{m}\sum_{i} Z^{[l](i)}, mean\ of\ convolution\ layer\ output$$

$$\sigma^2 = \frac{1}{m}\sum_{i} (Z_i - \mu_i)^2, variance\ of\ convolution\ layer\ output$$

Figure 2.9: ReLU and Leaky ReLU activation functions

$$Z_{norm}^{[l](i)} = \frac{Z^{[l](i)} - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

$$Z^{[l](i)} = \gamma Z_{norm}^{[l](i)} + \beta$$

This Normalized output is sent to the ReLU Layer. Gamma and beta are the hyper parameters.

2.2.2 Recurrent Neural Networks

Traditional Neural Networks do not have the ability to remember previous events and inform it to the next neurons. Recurrent Neural Networks are basically used to address this scenario. RNN networks have loops in them which allow them to persist the information.

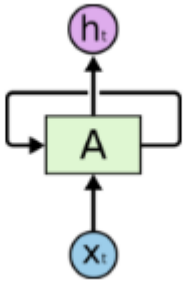


Figure 2.10: RNN loop

Having a loop helps the network to pass the information from one neuron to another. So an RNN can be thought of as multiple copies of the same network, each transferring an information to its immediate next one.

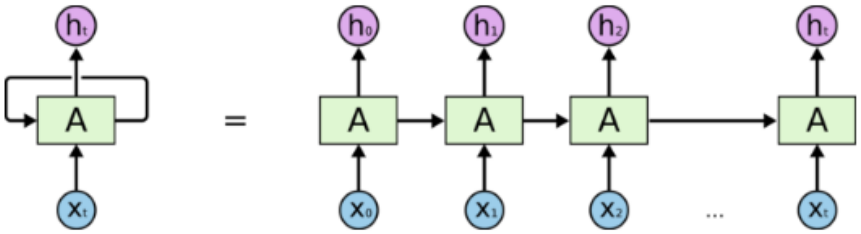


Figure 2.11: Unrolled Recurrent Neural Network

Long Short Term Memory networks, generally known as LSTMs are a special kind of RNN networks which are capable of learning long term dependencies. The repeating module in an LSTM has a different structure than RNN. In the repeating structure of an RNN we find a single neural network whereas in LSTM there are four of them interacting with each other.

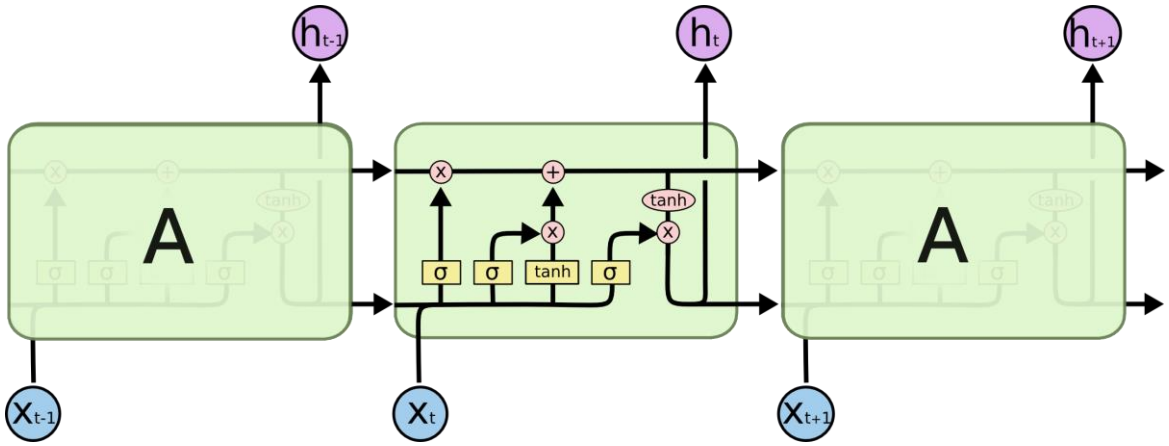


Figure 2.12: Long Short Term Memory network

The first sigmoid layer decides the information that needs to be kept and that which needs to be discarded from the previous layer. Hence this layer is called as the forget gate layer. Output from first sigmoid layer is:

$$f_t = \sigma(W_f.[h_{t-1}, x_t]) + b_f$$

The next sigmoid layer decides the values which have to be updated and the updating is done by tanh which creates a vector of new values $C_t$which can be added to the previous state.

$$i_t = \sigma(W_i.[h_{t-1}, x_t]) + b_i$$

$$\underline{C_t} = tanh(W_c.[h_{t-1}, x_t]) + b_c)$$

Updating the state of the cells:

$$C_t = f_t * C_{t-1} + i_t * \underline{C_t}$$

Finally, the output parts of the neuron are decided by the next sigmoid layer. Various other models have also been developed with a few changes to the standard architecture of LSTMs so as to get specific results.

CHAPTER 3

Methodology

3.1 Methodology of Sentiment Analysis

3.11 Pre-Processing

The data set containing tweets from several users will be noisy since people use language which is often informal to express their views including emoticons and several punctuations. Hence it is necessary to process these tweets into a format which is understandable by the machine using several special characteristics of twitter such as retweets, user mentions, emoticons etc.

Various Pre-processing steps which we used include:

- Urls which users mention in tweets do not provide any information related to the sentiment are the text and are therefore removed and mentioned as URL.
- Converting the entire tweet into lower case.
- Replacing multiple continuous spaces into a single space.
- Dividing the entire emoticon set of twitter and assigning tags of either positive or negative to every emotion which is mentioned in the user's tweet.
- Replacing hashtags used by users to mention trending issues by the hash represented by it.
- Retweets are mentioned by the letters RT at the beginning of the tweet. We remove this from the tweet since it doesn't provide any relevant information.
- We also replaced 2 or more dots with a space.

3.12 Feature Extraction - Unigrams and Bigrams

One of the most commonly used feature in text classification is Unigrams which can later be used as tokens to represent a tweet in vector space. Unigrams are basically single words from the dataset which are extracted along with their frequency distribution in the dataset. Out of these words most of them which occur at the bottom i.e. low frequencies are considered as noise and only the top N words are taken into consideration for classifying a vector in vector space.

Bigrams are pairs of words which occur in the dataset several times. These features can be used to model negation in a tweet - This is *not good.* After extracting the possible bigrams, we selected the top 10000 to create our vocabulary.

3.13 Feature Representation

Sparse Vector Representation uses either unigrams giving a vector size of 15000 or both the unigrams and bigrams i.e. giving the vector a size of 25000 (15000 unigrams + 10000 bigrams). The feature vector for a given tweet has a positive value at the indices of both uni and bigrams. This value is defined by the frequency feature type that we used.

In this feature, a positive value I assigned to uni and bigrams corresponding to their frequency in that particular tweet and zero to all the other uni and bigrams. A matrix for the entire dataset is then constructed using term frequencies which are then scaled up using the inverse document (idf) frequency of the term in order to give relative importance to terms. The idf for a term t is:

$$idf(t) = log(\frac{1 + n_d}{1 + df(dt)}) + 1$$

Where $n_d$and $df(dt)$represent the total documents and the number of documents in which t occurs respectively in the dataset.

When it comes to Dense vector representation, we took the top 90000 unigrams from the dataset and then assign integers to the unigrams with 1 being the most frequent one. After this each tweet is represented by a vector of values corresponding to the assigned integers to the words in the tweet and we considered a 200 dimensional vector assuming that the number of words in a tweet do not exceed 200.

3.14 Convolutional Neural Network Architecture

We use Dense vector representation of a tweet to train the CNN model.

Embedding Layer, which is the first layer is a $(v + 1) * d$ order matrix where d and v represent the dimention of word vector and the vocabulary size respectively. This layer is initialised by random weights from the normalised Gaussian function. Each dense vector is padded with 0's until its length is equal to the maximum length.

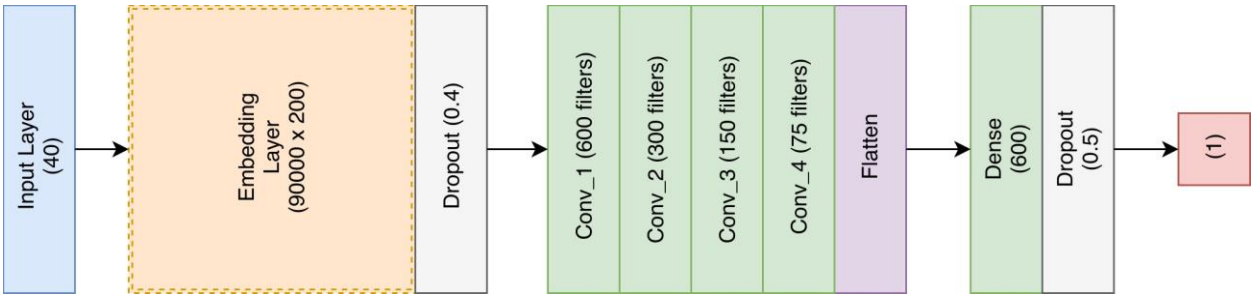


Figure 3.1: Architecture of the Neural Network

In the first convolution layer we performed temporal convolution which has a kernel of size 3. After every convolution layer we apply relu activation function. Dropout layers are also added to the model in order to avoid overfitting of the model.

3.2 Methodology of Facial Expression Recognition:

A Facial Expression System is a two-step process comprising of face detection(bounded) in an image followed by expression detection on the detected bounded face. We have used two techniques for respective tasks mentioned above in facial expression recognition.

1.Haar feature-based cascade classifier: In this implementation we have used Haar-cascade detection in OpenCV. OpenCV has a trainer as well as a detector. OpenCV already consists of many pre-trained models including the face model which is used for frontal face detection in our model. It is real-time and is faster compared to other face detectors.

2.Xception CNN Model: The above detected bounded face is used for facial expression recognition in real-time. The bounded face is used as input for training a classification CNN model architecture and predicts probabilities of 7 emotions in the output layer.

Dataset:

Fer2013 dataset is used for training the xception model to recognize the facial expression. The dataset consists of 35,887 grayscales ,48 x 48 sized face images with 7 different emotions which area all labelled.

Emotion labels in the dataset:

Label	Number of Images	Emotion
0	4593	Anger
1	547	Disgust
2	5121	Fear

3	8989	Happy
4	6077	Sad
5	4002	Surprise
6	6198	Neutral

The method is divided into the three components (i) preprocessing, (ii) Model architecture, and (iii) Model training.

i) Preprocessing:

In the fer2013 dataset, each image is given as a string of size 1x2304 which is 48 x 48 image stored as a row vector. The strings in the csv are first converted into images. These images are then normalized and resized using standard image pre-processing way by scaling then between -1 to 1. Images are first scaled to [0,1] by dividing it by 255 followed by further subtraction by 0.5 and multiplication by 2. This changes the range to [-1,1] which is a better range for neural network models in computer vision.

ii) Model architecture:

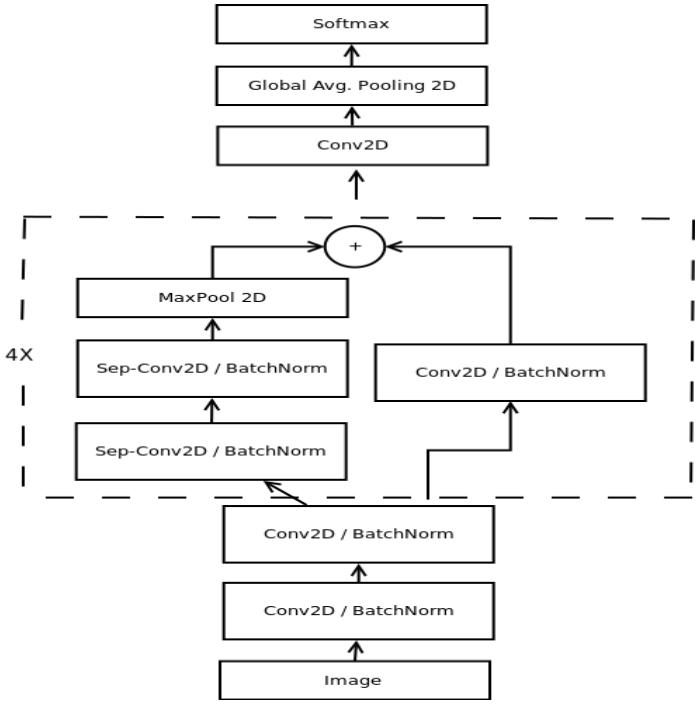


Figure 3.2: Model Architecture

The centre block is repeated 4 times in the architecture which is different from most common CNN architectures. In common architectures which use standard convolutions most of the parameters are present at the end where fully connected layers are present. Modern CNN architectures like Xception use combination of residual modules and depth-wise separable convolutions.

This model is inspired from Xception architecture. Residual modules and depth-wise separable convolutions are combined for this architecture. The desired mapping between two subsequent layers is modified by Residual modules. This makes the learned features equal to the difference of original feature map and the desired features. Consequently, the desired features H(x) are modified in order to solve an easier learning problem F(x) such that H(x)=F(x)+x.

The amount of parameters is further reduced by eliminating them from the convolution layers. This was done through the use of depth-wise separable convolutions. Depth-wise separable convolutions are broken into two operations depth-wise convolutions and point-wise convolutions.

- **Depth-wise convolution:** spatial convolution performed independently over each channel of an input.
- **Point-wise convolution:** A 1x1 convolution projects the channels output by depth-wise convolution into a new channel space.

Separable convolutions first perform channel-wise spatial convolution followed by 1x1 convolution.

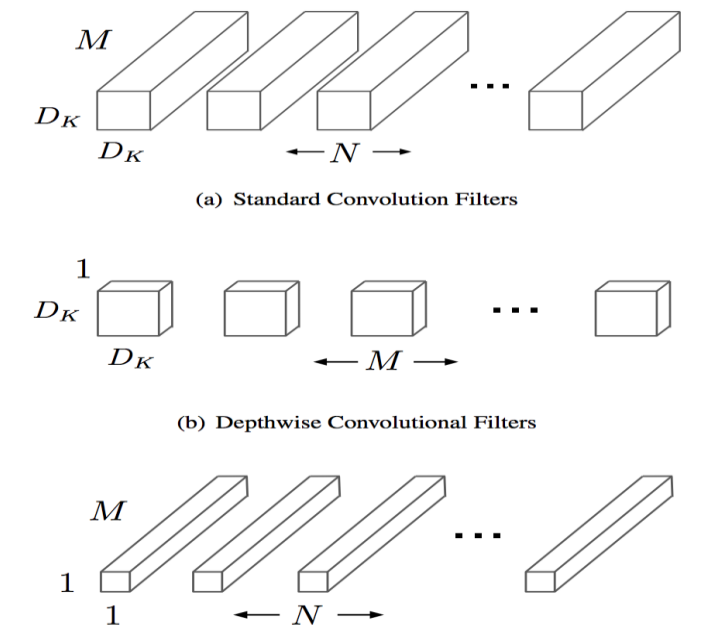


Figure 3.3: Different Convolution filters

iii) Model training:

There are various techniques used while building a deep neural network which are also applicable in most of computer vision problems. The following techniques are used while training the above architecture.

1.Data Augmentation: It is increasing the number of images or generating more data in the dataset by applying transformations. When the training set is not sufficient enough to learn representation data augmentation is required. More image data is generated by transforming the training images by rotations, crop, zoom, shifts, shear, reflection, normalization, translation etc.

2.Batch Normalization: Batch normalization is a method used to normalize the inputs of each layer i.e. applies a transformation that makes the mean activation close to 0 and activation standard deviation close to 1, this solves the problem of covariance shift. It acts as a regularizer and also helps in speeding up the training process.

3.Kernel_regularizer: Any function that takes in a weight matrix and returns a loss contribution tensor can be used as a regularizer. Regularizers allow to apply penalties on layer parameters or layer activity during optimization. These penalties are incorporated in the loss function that the network optimizes.

4.Global Average Pooling(GAP): Global Average Pooling layers are used to minimize overfitting by reducing the total number of parameters in the model. Each feature map is reduced into a scalar value by taking the average over all elements in the feature map. The global features are extracted from the input image using average operation.

5.Depth-wise Separable Convolution: This was thoroughly discussed in the model architecture above.

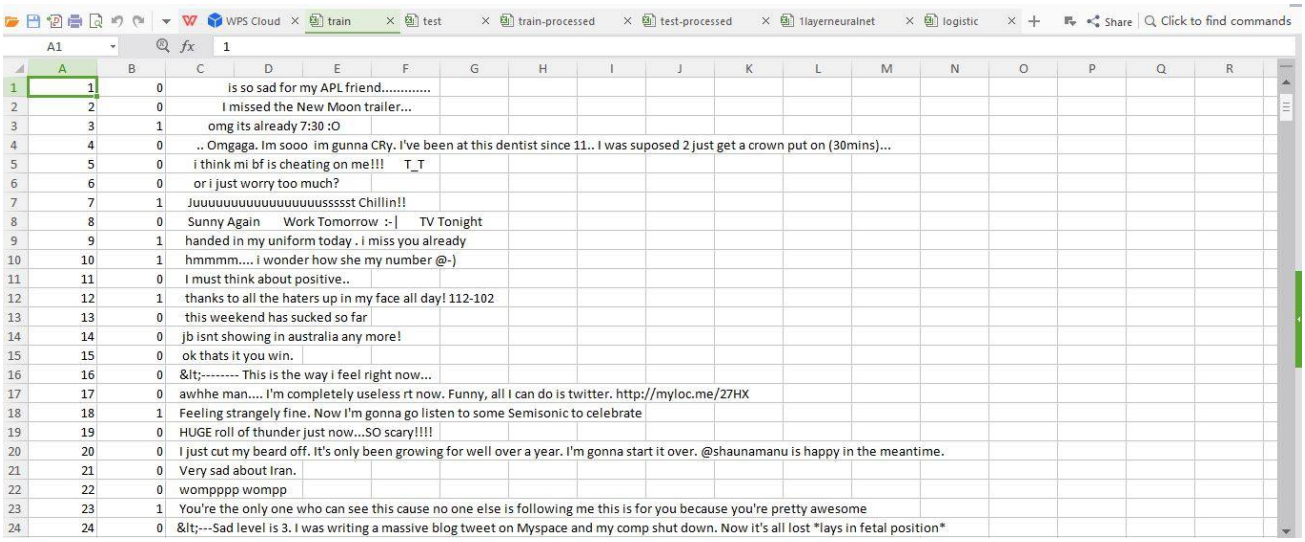
CHAPTER 4

Experiments and Discussion

4.1 Sentimental Analysis Model

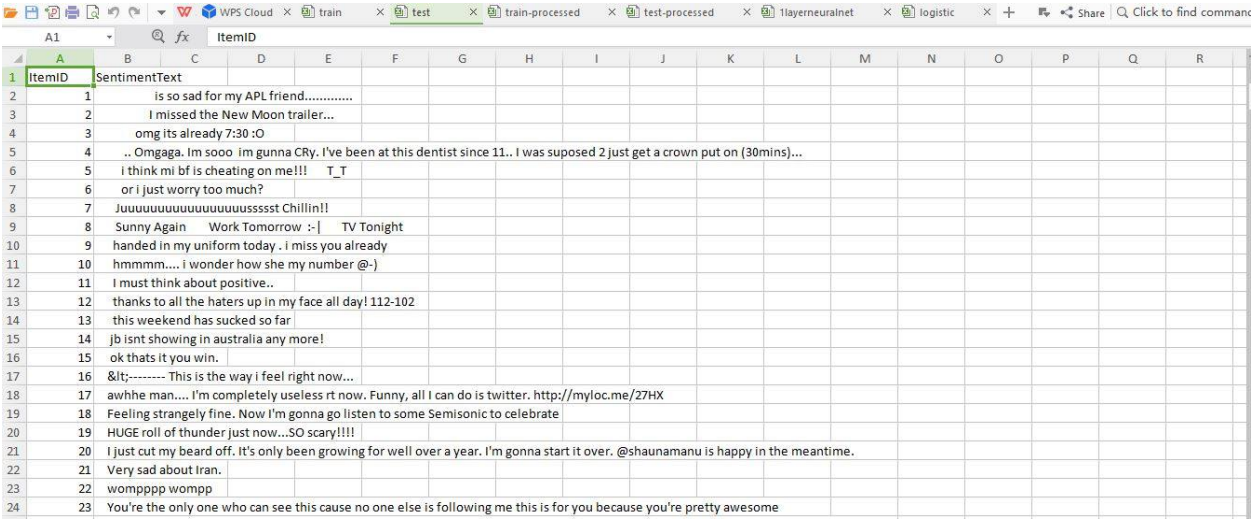
4.1.1 Dataset & Features

The twitter dataset for our models is divided into two parts called Training dataset and testing dataset. The difference between both of them is that the training dataset also contains the sentiment of the text and helps to train the various models and the trained model is then used to predict the sentiment of the test dataset. The format of the data in both of these parts before processing is shown below:



	A	B	C
1	1	0	is so sad for my APL friend.....
2	2	0	I missed the New Moon trailer...
3	3	1	omg its already 7:30 :O
4	4	0	.. Omgaga. Im sooo im gunna CRy. I've been at this dentist since 11.. I was suposed 2 just get a crown put on (30mins)...
5	5	0	i think mi bf is cheating on me!!! T_T
6	6	0	or i just worry too much?
7	7	1	Juuuuuuuuuuuuuuuussssst Chillin!!
8	8	0	Sunny Again Work Tomorrow :-| TV Tonight
9	9	1	handed in my uniform today . i miss you already
10	10	1	hmmmm.... i wonder how she my number @-)
11	11	0	I must think about positive..
12	12	1	thanks to all the haters up in my face all day! 112-102
13	13	0	this weekend has sucked so far
14	14	0	jb isnt showing in australia any more!
15	15	0	ok thats it you win.
16	16	0	<-------- This is the way i feel right now...
17	17	0	awhhe man.... I'm completely useless rt now. Funny, all I can do is twitter. http://myloc.me/27HX
18	18	1	Feeling strangely fine. Now I'm gonna go listen to some Semisonic to celebrate
19	19	0	HUGE roll of thunder just now...SO scary!!!!
20	20	0	I just cut my beard off. It's only been growing for well over a year. I'm gonna start it over. @shaunamanu is happy in the meantime.
21	21	0	Very sad about Iran.
22	22	0	wompppp wompp
23	23	1	You're the only one who can see this cause no one else is following me this is for you because you're pretty awesome
24	24	0	<---Sad level is 3. I was writing a massive blog tweet on Myspace and my comp shut down. Now it's all lost *lays in fetal position*

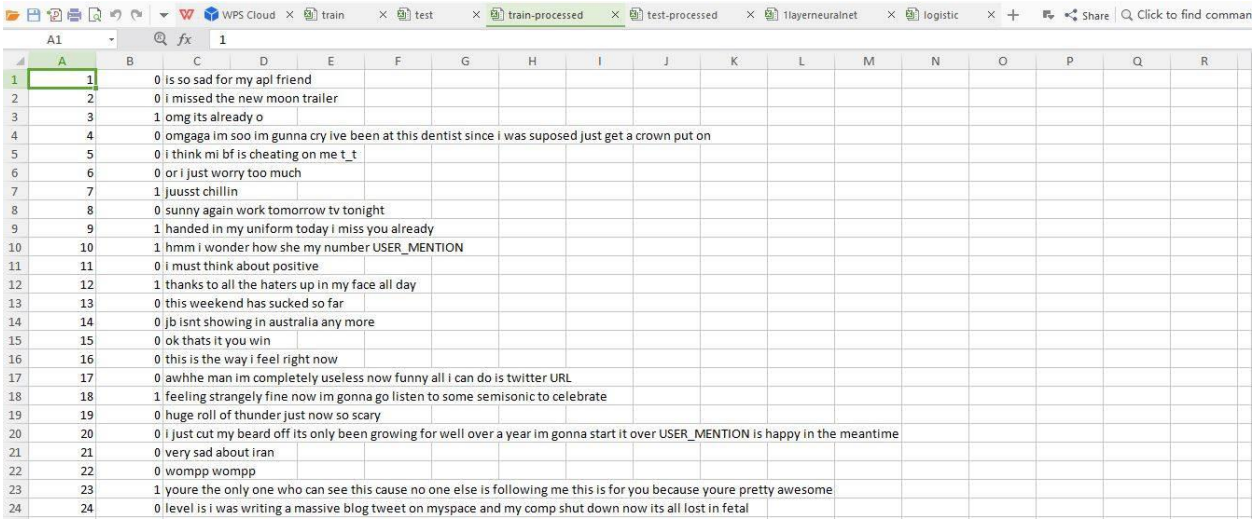
Figure 4.1: Training dataset before processing containing the sentiment of the tweet



ItemID	SentimentText
1	is so sad for my APL friend.....
2	I missed the New Moon trailer...
3	omg its already 7:30 :O
4	.. Omgaga. Im sooo im gunna CRy. I've been at this dentist since 11.. I was suposed 2 just get a crown put on (30mins)...
5	i think mi bf is cheating on me!!! T_T
6	or i just worry too much?
7	Juuuuuuuuuuuuuuuussssst Chillin!!
8	Sunny Again Work Tomorrow :-| TV Tonight
9	handed in my uniform today . i miss you already
10	hmmmm.... i wonder how she my number @-)
11	I must think about positive..
12	thanks to all the haters up in my face all day! 112-102
13	this weekend has sucked so far
14	jb isnt showing in australia any more!
15	ok thats it you win.
16	<-------- This is the way i feel right now...
17	awhhe man.... I'm completely useless rt now. Funny, all I can do is twitter. http://myloc.me/27HX
18	Feeling strangely fine. Now I'm gonna go listen to some Semisonic to celebrate
19	HUGE roll of thunder just now...SO scary!!!!
20	I just cut my beard off. It's only been growing for well over a year. I'm gonna start it over. @shaunamanu is happy in the meantime.
21	Very sad about Iran.
22	womppppp wompp
23	You're the only one who can see this cause no one else is following me this is for you because you're pretty awesome

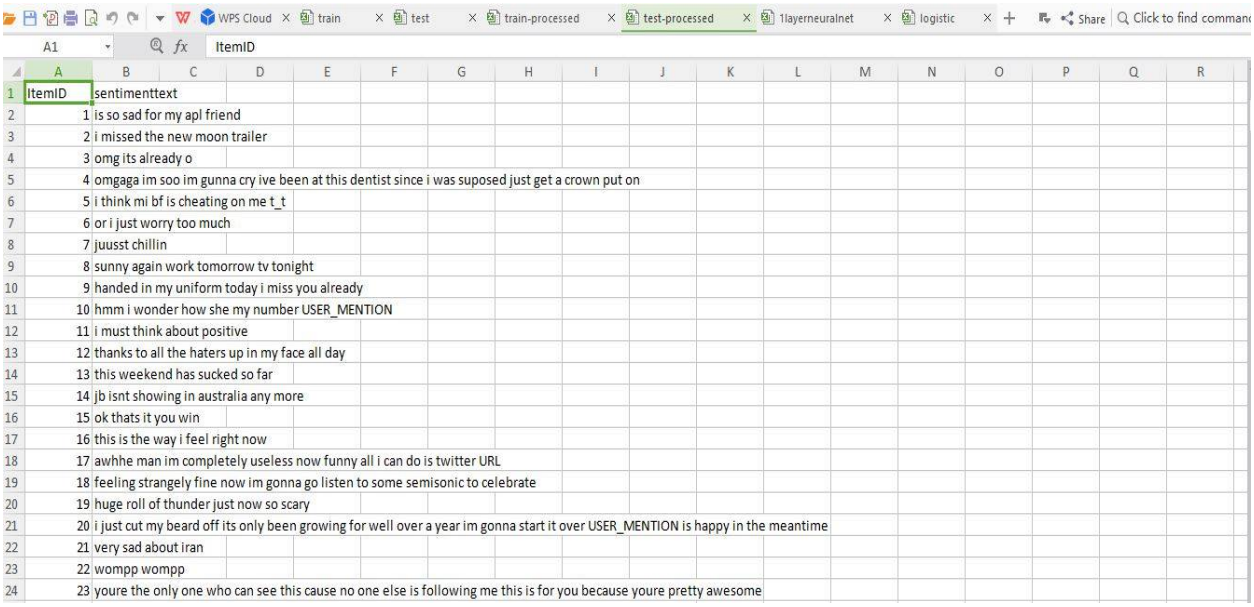
Figure 4.2: Test dataset before processing

After applying the various processing techniques to the test and train datasets we have the following:



1	0	is so sad for my apl friend
2	0	i missed the new moon trailer
3	1	omg its already o
4	0	omgaga im soo im gunna cry ive been at this dentist since i was suposed just get a crown put on
5	0	i think mi bf is cheating on me t_t
6	0	or i just worry too much
7	1	juusst chillin
8	0	sunny again work tomorrow tv tonight
9	1	handed in my uniform today i miss you already
10	1	hmm i wonder how she my number USER_MENTION
11	0	i must think about positive
12	1	thanks to all the haters up in my face all day
13	0	this weekend has sucked so far
14	0	jb isnt showing in australia any more
15	0	ok thats it you win
16	0	this is the way i feel right now
17	0	awhhe man im completely useless now funny all i can do is twitter URL
18	1	feeling strangely fine now im gonna go listen to some semisonic to celebrate
19	0	huge roll of thunder just now so scary
20	0	i just cut my beard off its only been growing for well over a year im gonna start it over USER_MENTION is happy in the meantime
21	0	very sad about iran
22	0	wompp wompp
23	1	youre the only one who can see this cause no one else is following me this is for you because youre pretty awesome
24	0	level is i was writing a massive blog tweet on myspace and my comp shut down now its all lost in fetal

Figure 4.3: Processed Training dataset



ItemID	sentimenttext
1	is so sad for my apl friend
2	i missed the new moon trailer
3	omg its already o
4	omgaga im soo im gunna cry ive been at this dentist since i was suposed just get a crown put on
5	i think mi bf is cheating on me t_t
6	or i just worry too much
7	juusst chillin
8	sunny again work tomorrow tv tonight
9	handed in my uniform today i miss you already
10	hmm i wonder how she my number USER_MENTION
11	i must think about positive
12	thanks to all the haters up in my face all day
13	this weekend has sucked so far
14	jb isnt showing in australia any more
15	ok thats it you win
16	this is the way i feel right now
17	awhhe man im completely useless now funny all i can do is twitter URL
18	feeling strangely fine now im gonna go listen to some semisonic to celebrate
19	huge roll of thunder just now so scary
20	i just cut my beard off its only been growing for well over a year im gonna start it over USER_MENTION is happy in the meantime
21	very sad about iran
22	wompp wompp
23	youre the only one who can see this cause no one else is following me this is for you because youre pretty awesome

Figure 4.4: Processed Testing Dataset

Once the datasets are processed, we now extract features such as unigrams and bigrams from them.

```
In [3]: with open(FREQ_DIST_FILE, 'rb') as pkl_file:
            freq_dist = pickle.load(pkl_file)
        unigrams = freq_dist.most_common(20)
        unigrams

Out[3]: [('i', 50081),
         ('the', 29777),
         ('to', 29563),
         ('you', 26869),
         ('a', 22065),
         ('it', 17440),
         ('and', 16353),
         ('my', 13779),
         ('for', 12548),
         ('is', 12055),
         ('in', 11807),
         ('that', 11411),
         ('im', 11186),
         ('me', 10980),
         ('of', 10602),
         ('have', 9954),
         ('on', 9621),
         ('so', 9144),
         ('but', 9026),
         ('be', 7435)]
```

Figure 4.5: Snippet of the Code for producing Unigrams

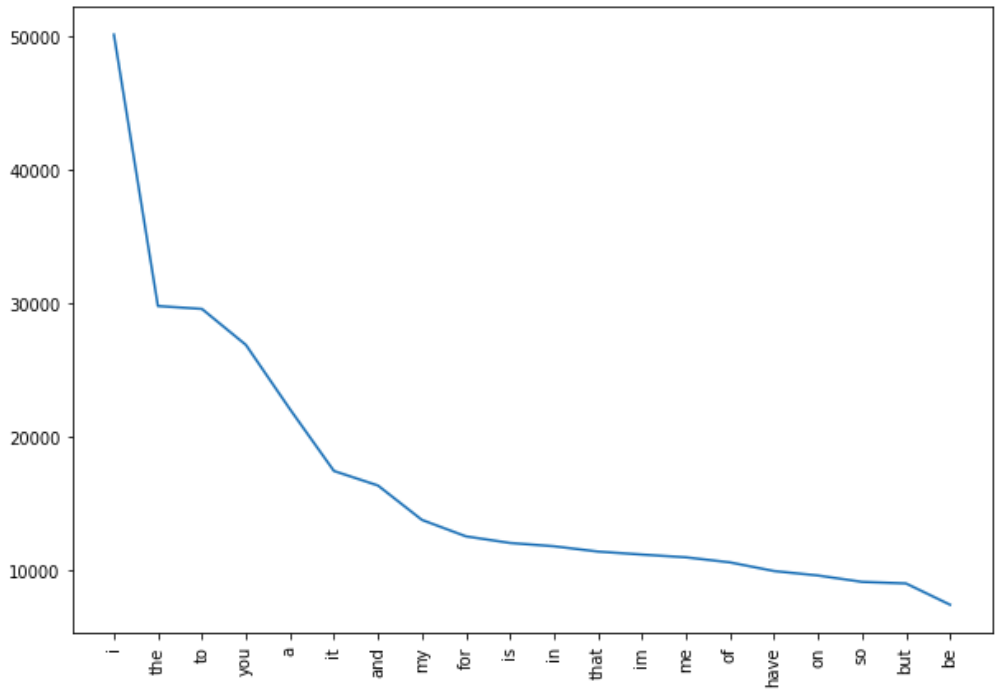


Figure 4.6: Unigram Frequency Distribution Graph

```
In [5]: with open(BI_FREQ_DIST_FILE, 'rb') as pkl_file:
            freq_dist = pickle.load(pkl_file)
        bigrams = freq_dist.most_common(20)
        bigrams

Out[5]: [(('i', 'have'), 2444),
         (('in', 'the'), 2376),
         (('i', 'was'), 2272),
         (('for', 'the'), 2167),
         (('i', 'dont'), 2075),
         (('i', 'am'), 2051),
         (('have', 'a'), 1924),
         (('but', 'i'), 1904),
         (('i', 'love'), 1880),
         (('i', 'know'), 1871),
         (('on', 'the'), 1743),
         (('to', 'be'), 1696),
         (('have', 'to'), 1684),
         (('i', 'think'), 1683),
         (('and', 'i'), 1617),
         (('going', 'to'), 1588),
         (('i', 'cant'), 1587),
         (('it', 'was'), 1536),
         (('of', 'the'), 1458),
         (('thanks', 'for'), 1380)]
```

Figure 4.7: Code Snippet for producing Bigrams

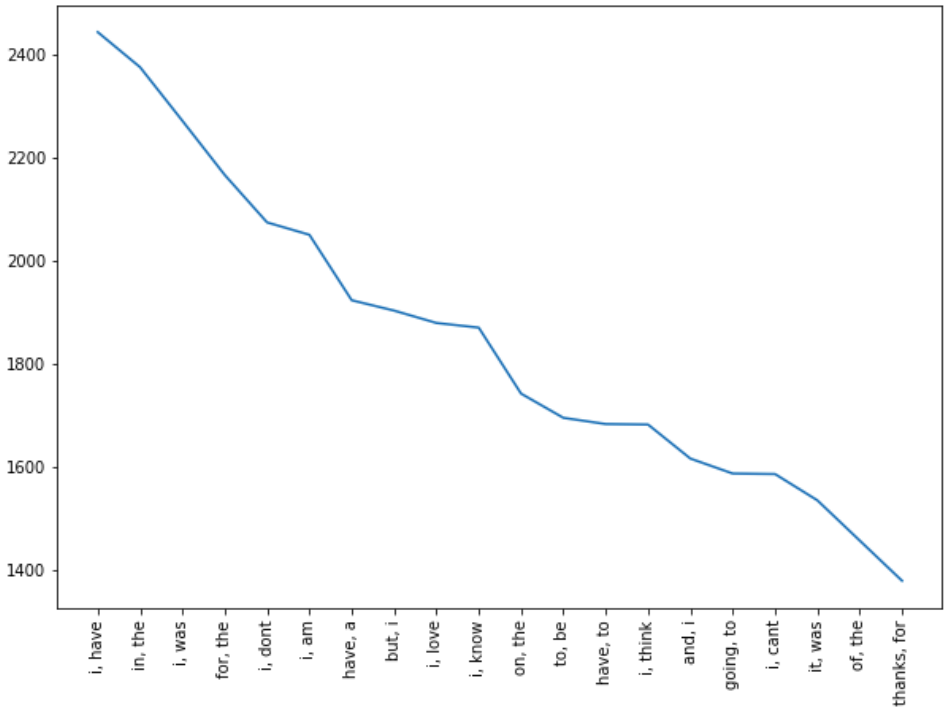


Figure 4.8: Frequency Distribution of Bigrams

4.1.2 Implementing Various Models:

After processing the dataset, extracting the required features and creating spatial vectors, the datasets are trained on several models as mentioned in the literature survey and their accuracies have been noted.

```
(newenvt) C:\Users\HP\Desktop\BTP\Trials2\twitter-sentiment-analysis-master\code>python decisiontree.py
Generating feature vectors
Processing 100000/100000

Extracting features & training batches
Processing 1/1

Testing
Processing 1/1
Correct: 6875/10000 = 68.7500 %
```

Figure 4.9: Decision Tree Classifier

```
(newenvt) C:\Users\HP\Desktop\BTP\Trials2\twitter-sentiment-analysis-master\code>python naivebayes.py
Generating feature vectors
Processing 100000/100000

Extracting features & training batches
Processing 1/1

Testing
Processing 1/1
Correct: 7713/10000 = 77.1300 %
```

Figure 4.10: Naive Bayes Classifier

```
Iteration 180/180, loss:0.5264, acc:0.7520
Epoch: 1, val_acc:0.7648
Accuracy improved from 0.0000 to 0.7648, saving model
Iteration 180/180, loss:0.4442, acc:0.7900
Epoch: 2, val_acc:0.7716
Accuracy improved from 0.7648 to 0.7716, saving model
Iteration 180/180, loss:0.4424, acc:0.8120
Epoch: 3, val_acc:0.7719
Accuracy improved from 0.7716 to 0.7719, saving model
Iteration 180/180, loss:0.4494, acc:0.8020
Epoch: 4, val_acc:0.7680
Iteration 180/180, loss:0.4515, acc:0.8020
Epoch: 5, val_acc:0.7673
Testing
Generating feature vectors
Processing 300001/300001

Predicting batches
Processing 601/601
Saved to 1layerneuralnet.csv
```

Figure 4.11: Basic Neural Network

```
(newenvt) C:\Users\HP\Desktop\BTP\Trials2\twitter-sentiment-analysis-master\code>python svm.py
Generating feature vectors
Processing 100000/100000

Extracting features & training batches
Processing 1/1

Testing
Processing 1/1
Correct: 7794/10000 = 77.9400 %
```

Figure 4.12: Support Vector Machine Classifier

```
(newenvt) C:\Users\HP\Desktop\BTP\Trials2\twitter-sentiment-analysis-master\code>python randomforest.py
C:\ProgramData\Anaconda3\envs\newenvt\lib\site-packages\sklearn\ensemble\weight_boosting.py:29: DeprecationWarning: nump
y.core.umath_tests is an internal NumPy module and should not be imported. It will be removed in a future NumPy release.

  from numpy.core.umath_tests import inner1d
Generating feature vectors
Processing 100000/100000

Extracting features & training batches
Processing 1/1

Testing
Processing 1/1
Correct: 7272/10000 = 72.7200 %
```

Figure 4.13: Random Forest Classifier

```
(newenvt) C:\Users\HP\Desktop\BTP\Trials2\twitter-sentiment-analysis-master\code>python baseline.py
Correct = 65.35%
```

Figure 4.14: Baseline Classifier

As our baseline model, we use a word counter which simply counts the number of positive or negative words in a tweet and classify the tweet as positive if it has higher number of positive words and vice versa. The various accuracies of the models which we used and their comparison can be seen in the graph below:

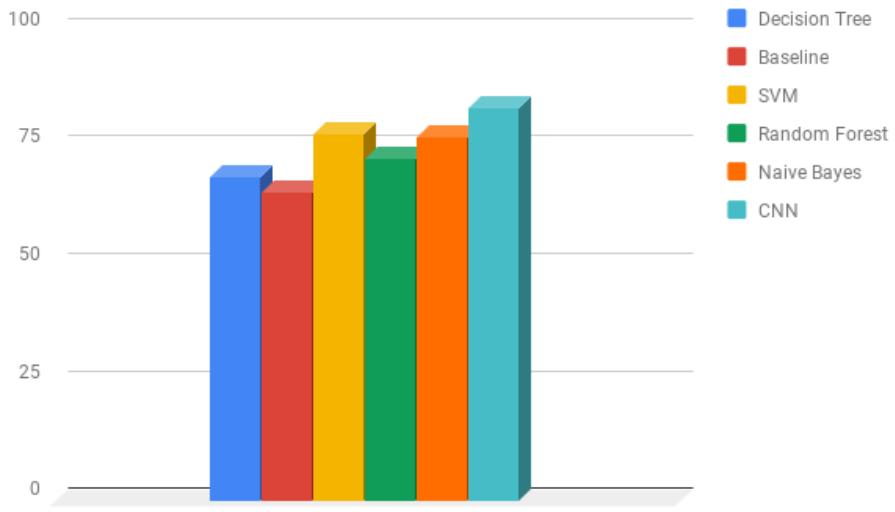


Figure 4.15: Accuracy plot

4.2 Facial Expression Detection

The CNN architecture which was described above was implemented in both real time and on images as input and various emotions have been identified by the system which can be seen below:

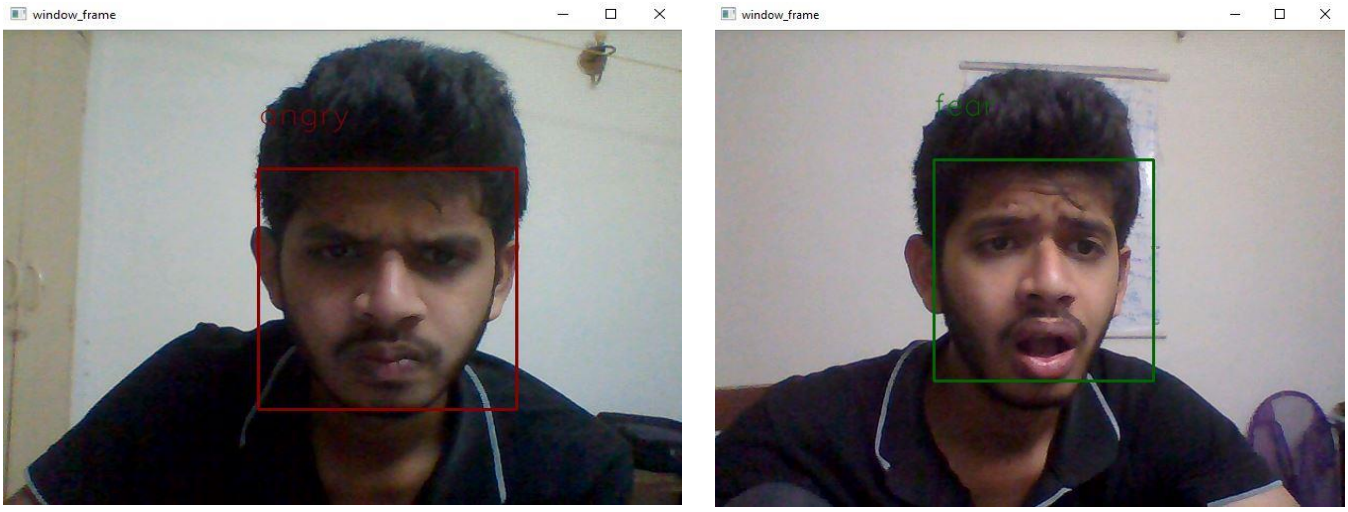


Figure 4.16: Real Time images showing Anger and Fear

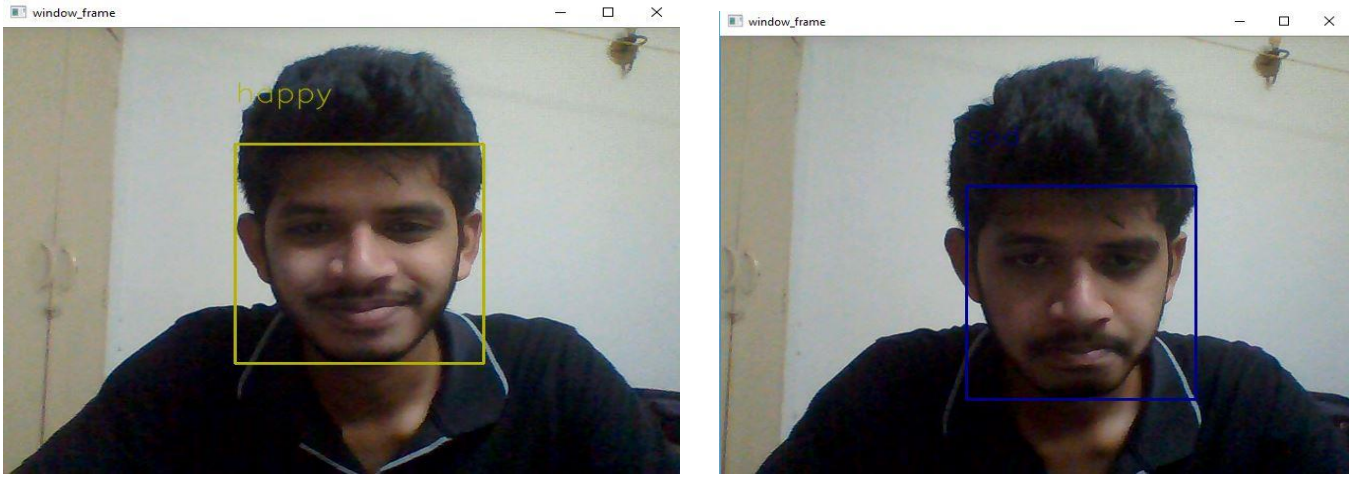


Figure 4.17: Real Time images showing Happy and Sad Emotions

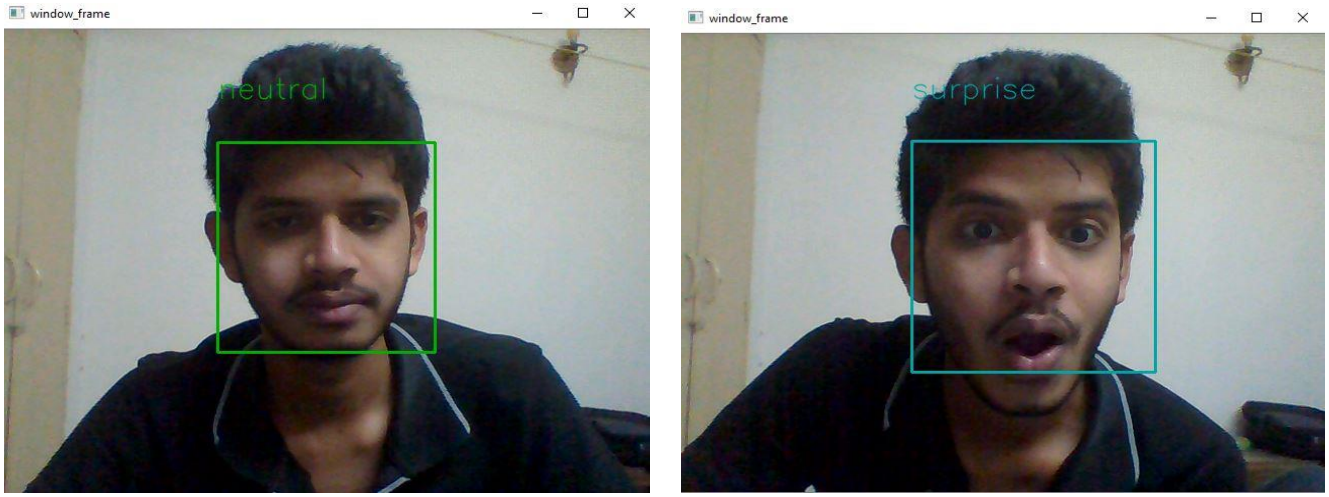


Figure 4.18: Real Time images showing Neutral and Surprise Emotions

Inference on Single random images:

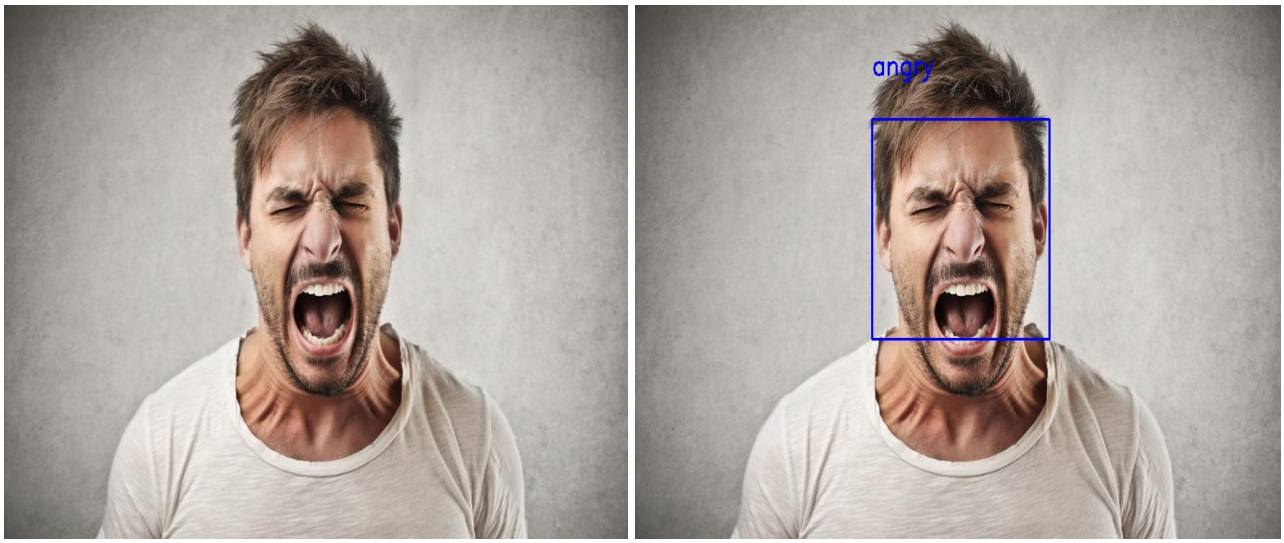


Figure 4.19: Inference of image representing Anger Emotion

Figure 4.20: Inference on image representing Happy emotion

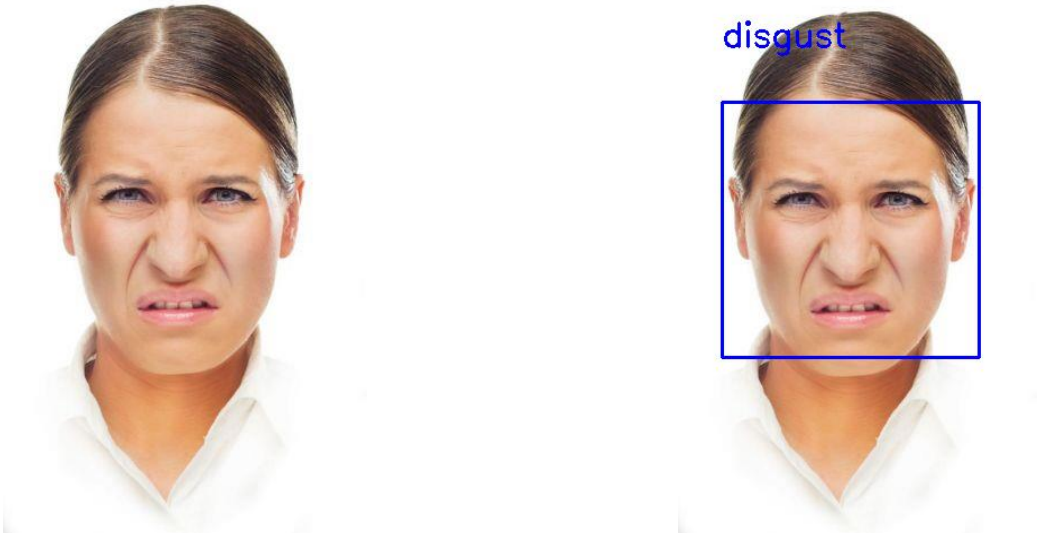


Figure 4.21: Inference on image representing Disgust emotion

Figure 4.22: Inference on image representing Surprise Emotion

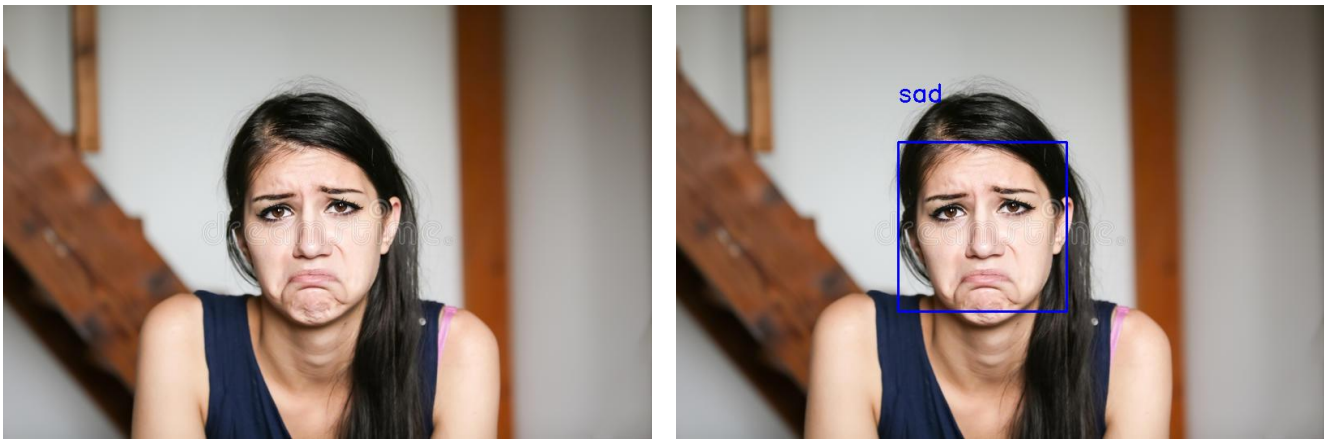


Figure 4.23: Inference on image representing Sad emotion

Figure 4.24: Inference on image representing Neutral emotion

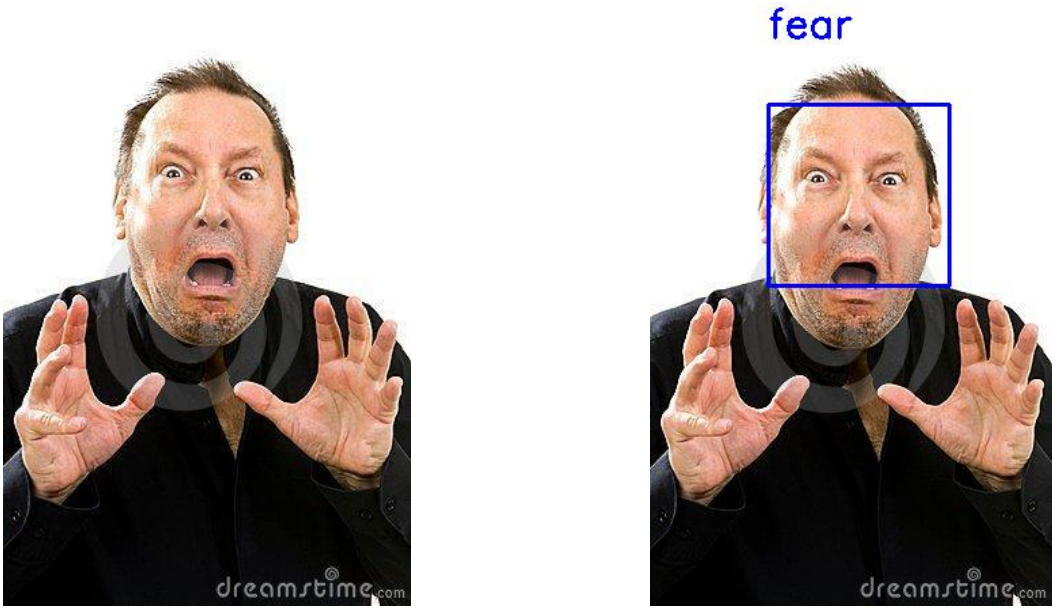


Figure 4.25: Inference on image representing Fear emotion

4.2.1 Discussion

It can be observed that the accuracy obtained by applying the proposed CNN architecture is higher when compared to other classical approaches for the analysis of text. This is comparable to the state-of-the art methods used in this area. In the Facial Expression Recognition, the architecture can be used in real time without the requirement of heavy hardware with an acceptable accuracy. Using the above mentioned architecture on fer2013 dataset we achieved an accuracy of 66 % on all the seven emotion labels which is close to the best model in this domain. We observed that the expressions anger, sad and happiness are observed with greater accuracy when compared to expressions such as disgust and fear. This is because disgust and fear closely resemble anger and sad respectively due to which these misclassifications take place. Further, the use of glasses also effects the classification by interfering with the features.

CHAPTER 5

Conclusion and Future Work

In this work, we discussed several ground-breaking works in Sentimental Analysis and Emotion Detection that has revolutionized several fields especially the field of computer vision. We used a new architecture to implement Convolutional Neural Networks for facial emotion recognition in real time once trained and show that the results are comparable to the state of the art technologies. This model is found to be very efficient and a decent accuracy to detect facial emotions. We feel that the lack of a rich dataset and training acted as a hindrance to completely test how this architecture performs in real time.

The drawback of the sentiment analysis model is that as we increase the number of sentiments to classify a given statement the accuracy decreases substantially which makes it difficult to use, especially in real time. A lot of research is still going on in this field in order to overcome these problems. Most of it is currently focused to increase the accuracy of sentiment and facial recognition systems and to simultaneously decrease the hardware required in order for the system to interact with humans in real time. We can further develop the Sentimental Analysis model to finish unfinished sentences using the context of it which is generally used in search engines like google.

Bibliography

[1] B. Yang, J. Cao, R. Ni and Y. Zhang, "Facial Expression Recognition Using Weighted Mixture Deep Neural Network Based on Double-Channel Facial Images," in *IEEE Access*, vol. 6, pp. 4630-4640, 2018.

[2] S. Xie and H. Hu, "Facial Expression Recognition Using Hierarchical Features with Deep Comprehensive Multipatches Aggregation Convolutional Neural Networks," in *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 211-220, Jan. 2019.

[3] H. Li, J. Sun, Z. Xu and L. Chen, "Multimodal 2D+3D Facial Expression Recognition with Deep Fusion Convolutional Neural Network," in *IEEE Transactions on Multimedia*, vol. 19, no. 12, pp. 2816-2831, Dec. 2017.

[4] Francois Chollet. Xception: Deep learning with depth-wise separable convolutions. CoRR, abs/1610.02357, 2016.

[5] S. Poria, I. Chaturvedi, E. Cambria and A. Hussain, "Convolutional MKL Based Multimodal Emotion Recognition and Sentiment Analysis," 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, 2016, pp. 439-448.

[6] Walaa Medhat, Ahmed Hassan, Hoda Korashy, “Sentiment analysis algorithms and applications: A survey,” Ain Shams Engineering Journal, Volume 5, Issue 4,2014, Pages 1093-1113, ISSN 2090-4479,

[7] X. Fu, J. Yang, J. Li, M. Fang and H. Wang, "Lexicon-Enhanced LSTM with Attention for General Sentiment Analysis," in *IEEE Access*, vol. 6, pp. 71884-71891, 2018.

[8] C. Clavel and Z. Callejas, "Sentiment Analysis: From Opinion Mining to Human-Agent Interaction," in *IEEE Transactions on Affective Computing*, vol. 7, no. 1, pp. 74-93, 1 Jan.-March 2016.

[9] ThoughtWorks. (2019). Recognizing human facial expressions with machine learning. [online] Available at: https://www.thoughtworks.com/insights/blog/recognizing-human-facial-expressions-machine-learning [Accessed 5 May 2019].

[10] Cs231n.stanford.edu. (2019). Stanford University CS231n: Convolutional Neural Networks for Visual Recognition. [online] Available at: http://cs231n.stanford.edu/2018/ [Accessed 5 May 2019].

[11] Pramerdorfer, C., & Kampel, M. (2016). Facial Expression Recognition using Convolutional Neural Networks: State of the Art. *CoRR, abs/1612.02903*.

[12] Keras.io. (2019). Home - Keras Documentation. [online] Available at: https://keras.io/ [Accessed 5 May 2019].