MULTIMODAL SENTIMENT ANALYSIS

MINI PROJECT REPORT

Submitted by

ABDUL BASITH M (20EUAI002)

ADITYA S (20EUAI003)

ARAVIND G (20EUAI005)

BHARATHI RAMANAN D (20EUAI006)

in partial fulfillment of the requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

COIMBATORE

(An Autonomous Institution)

(Approved by AICTE and Affiliated to Anna University, Chennai)

ACCREDITED BY NAAC WITH "A" GRADE

NOVEMBER 2022

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution)

(Approved by AICTE and Affiliated to Anna University, Chennai)

ACCREDITED BY NAAC WITH "A" GRADE

NOVEMBER 2022

BONAFIDE CERTIFICATE

Certified that this mini project report titled **"MULTIMODAL SENTIMENT ANALYSIS"** is the bonafide work of **ABDUL BASITH (20EUAI002), ADITYA S (20EUAI003), ARAVIND G (20EUAI005), BHARATHI RAMANAN D (20EUAI006)** who carried out the mini project work under my supervision.

SIGNATURE

Dr. T. SUJATHA

SUPERVISOR,

ASSOCIATE PROFESSOR

SIGNATURE

Dr. S. VENKATA LAKSHMI

HEAD OF THE DEPARTMENT

Department of Artificial Intelligence and Data Science

Sri Krishna College of Engineering and Technology

Kuniyamuthur, Coimbatore.

This Mini Project report is submitted for Autonomous Mini Project Viva-Voce examination held on ……...

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our sincere thanks to the management and **Dr. J. JANET M.E., Ph.D.,** Principal, Sri Krishna College of Engineering and Technology, Coimbatore for providing us the facilities to carry out this mini project work.

We are thankful to **Dr. S. VENKATA LAKSHMI, M.Tech., Ph.D.,** Professor and Head, Department of Artificial Intelligence and Data Science, for her continuous evaluation and comments given during the course of the mini project work.

We express our deep sense of gratitude to our supervisor **Dr. T SUJATHA**, **M.Tech., Ph.D.,** Associate Professor, Department of Artificial Intelligence and Data Science for her valuable advice, guidance and support during the course of our mini project work.

We would also like to thank our mini project coordinator **Mr. G. S. PUGALENDHI, M.E.,** Assistant Professor, Department of Artificial Intelligence and Data Science for helping us in completing our mini project work.

We express our heartfelt sense of gratitude and thanks to our beloved parents, family and friends who have helped during the mini project course.

ABSTRACT

Sentiment Analysis intends to naturally reveal the hidden mentality that we hold towards an entity. At present text-based sentiment analysis depends on the development of word embeddings and Machine Learning models such as Long Short-Term Memory (LSTM) and Transformers model that make conclusion via enormous text collection. As there is a growing demand to automate analysis of user sentiment towards products and services, Opinions are increasingly being available in various media formants such as audio and video too. This led to sentimental analysis of multiple modalities, termed Multimodal Sentiment Analysis which offers good roads for complete sentiment analysis by not only taking textual input but also considering audio and video inputs. Convolutional Neural Networks (CNNs) along with the feature extraction are the methodologies that are used to increase the performance in video analysis. In audio-based sentiment analysis, Mel-Frequency Cepstral Coefficient (MFCC) are calculated to feed into Support Vector Machine (SVM) for increased performance. The results of the multimodal investigation are of great useful in analysing Social media monitoring, Customer support and Market research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

Multimodal sentimental analysis provides methods to carry out opinion analysis based on the combination of video, audio, and text which goes a way beyond the conventional text-based sentimental analysis in understanding human behaviours.

1.1 BACKGROUND INFORMATION

Multimodal Sentiment Analysis is another dimension of the customary text-based assessment investigation, which goes past the test of writings, and incorporates different modalities like sound and visual information. The sentiment is evoked when an individual experiences a particular topic, person, or element. Understanding individuals' position, disposition, or assessment towards a specific feature has numerous applications. The text-based feeling investigation has been the leading figure around here and, as of late, has examined different modalities, like audio and vision, started to be thought of in use.

Liu and Zhang [18] characterized sentiment analysis as an issue of automatic detection of four segments of a notion including, entity, viewpoint, entity holder, viewpoint's feeling. A good sentiment analysis framework ought to have the option to disengage this load of four segments accurately.

A new improvement in multimodal sentiment analysis is visual assumption investigation. Web based media clients regularly share instant messages with pictures/recordings, and these visible sights and sounds are extra direct data in communicating client notions. Mid-level visual supposition portrayals are one valuable development for separating feeling and elements in text-based notion investigation.

Recordings give multimodal information as far as vocal and visual modalities. The vocal balances and looks in the visual information, alongside text information, give significant prompts better to recognize genuine emotional conditions of the assessment holder. Consequently, a mix of text and video information assists with making a better feeling and assumption examination model.

1.2 MOTIVATION

Understanding emotion using text became so common throughout the years. Thus, introducing other models like audio is necessary and provides a broad domain in sentiment analysis. Model will be doing the Text Analysis by using LSTM and Bidirectional LSTM [7]. Audio data will be used to create Spectrograms or MFCC's using the Librosa library, which can predict the label using the spectrograms images with the CNN network or the MFCC values combined with the classification model. Multimodal Sentiment Analysis can be used in chatbots, call centres that can tell the customers' satisfaction after talking to a bot or even an employee.

1.3 OBJECTIVE

To create a Multimodal Sentiment Analysis that will extract the sentiments using the three modes, i.e., Audio, Text, and Video. Evaluating the datasets by checking the loss and accuracy of this model.

1.4 WORK DESCRIPTION

Text Sentiment Analysis is done by filtering the dataset, like reducing every word to its stem and passing the corpus through models with LSTM and Attention Layers to predict the sentiment.

Audio Sentiment Analysis is done by taking the Real-time Audio of a user and then calculating MFCC's to predict the user's emotion [8].

Video Sentiment Analysis is the simple use of detecting human facial expressions in real-timevideo using some Transfer Learning Techniques. A local server website will be created to deploy all these three modes in one.

1.5 REPORT LAYOUT

- Literature Survey describes all the previous works done in this field.
- Proposed System describes the architecture of the project.
- Implementation Details describes the tools that are used and the process that needs tobe followed in each mode.
- Module Description shows the final outputs that are done in the course of this project.
- Conclusion specifies the limitations and future scope of this project.

CHAPTER-2

LITERATURE SURVEY

Multimodal Sentiment Analysis is a dynamic theme in Natural Language Processing (NLP). It automatically removes individuals' perspectives or emotional states from numerous correspondence channels [4] (e.g., text, voice, and facial expressions). Furthermore, it has different applications. The center test displays the complex intra-modular and between modular cooperations, where multimodal highlights are being intertwined. Yanan Jia and Sony SungChu proposed the idea of Multimodal Sentiment Analysis in which they used two modes, i.e., Audio and Text for Sentiment Analysis; model here will add another method, i.e., of video mode that will use facial expressions.

2.1 TEXT SENTIMENT

As an aim to extract evaluative meaning, an alternative to topic detection in the field ofsentiment analysis was started.

Maybe the most encouraging improvement in text sentiment is due to the use of deep learning.Deep learning can leverage massive scope datasets to register word embeddings that are relevant for feeling examination, delivering naturally extended lexical. While the derivation ofword classes dependent on deep learning [12] strategies is accomplishing results exceptionally near those of human annotators, ongoing work found that extrapolating word sentiment consistent factors dependent on word embeddings still requires significant work. Profound Recurrent Neural Networks have been applied to the errand of subjectivity detection, and wordvector representations can join administered and unaided learning when applied to feeling analysis.

For Text Sentiment the authors of implemented SVM in their research, but the thought of using different techniques, and so model will work with Bidirectional LSTM's with the Attention Mechanism. Though specialists have stretched out LSTM cells and doors to learn fleeting collaboration designs among multimodal successions and also Pham-et-al proposedconsideration-based RNNs [3] to learn multimodal portrayals with a cyclic interpretation misfortune among modalities. Still, model give a chance to a Bidirectional LSTM that will helps to upbeat these mechanisms significantly.

2.2 AUDIO SENTIMENT

Targeting opinion unequivocally solely from spoken expressions is an equivalently youthful field. Zeroing in on the acoustic side of communication in language, theline among opinion and feeling investigation is regularly extremely frail, as, e. g. In Mairesseet al. zero in on pitch-related provisions and saw that addition- ally, without text-based signals [3], pitch contains data on feeling. Various further works center around feeling examination solely from the text-based substance as present in the discourse. For example, Costa Pereira etal's proposed approach takes a verbally expressed inquiry and recovers reports whose conclusions look like the question. Likewise, Pérez-Rosas and Mihalcea focus on the semantics of spoken audits in the wake of utilizing discourse acknowledgment. Kaushik et al.and its extension observe that feeling examination on normal unconstrained discourse information canbe acknowledged in any event when confronted with low word acknowledgment rates.

The Audio Sentiment implement by authors of used KNN for their purpose. Model will be calculating the MFCC's for carrying out the work in the Audio Field. Sequence models can be fitted dependent on channel banks, MFCCs, or any other low-level descriptors removed from crude discourse without highlight designing. In any case, this methodology, for the most part,requires exceptionally effective calculation and huge explained sound records. It used an audiodataset with the meantime for calls to be 4 seconds for the sentiment analysis in audio. Still, model will try to increase its mean to 7 seconds to check its progress for large audios as they haven't explored that region [4].

Zadeh et al. planned a multiview gated memory unit that neural organizations constrain. It stores furthermore, predicts fleeting cross-modular collaborations. Tsai et al. used transformer consideration systems to learn both cross-modular arrangements furthermore, collaborations. Albeit neural organizations extraordinarily work on the presentation over conventional techniques, and their unpredictable engineering genuinely influences the model interpretability.

2.3 VIDEO SENTIMENT

While there have been connected lines of examination in vision-based emotion acknowledgment for quite a while, e.g., directing sentiment investigation by computer vision is a somewhat ongoing region of research. The chief examination undertakings in "visual opinion analysis" spin around displaying, distinguishing, and utilizing sentiment expressed through facial or accurate signals or feeling associated with visual sight and sound.

Among the soonest work in visual opinion examination, Wanget al. investigated descriptor affiliations coordinated into 12 adjective–modifier word sets more than 100 pictures commented on by 42subjects. They utilized an assortment of shading high- lights, including lightness, immersion, and sharpness highlights related to support vector relapse to anticipate the presence of these sets like warm-cool, brilliant–gloomy, and vibrant–desolate

Every one of these work in promoting and applying visual feeling examination highlight the potential in the higher precision methods, as with CNNs, just as expanded inclusion, as with multilingual and different substance source methods. Furthermore, with the expanding number of freely accessible PC vision models/libraries and visual feeling datasets, visual opinion examination is ready to see development in both of these bearings. The complex idea of feeling shows that visual feeling investigation alone cannot wholly gauge and additionally portray our experiential attitude and sentiments in interactive media information. For instance, visible substance probably won’t have the option to comprehend the unique circumstance or concentrate the element.

In Video Sentiment model will be using simple face expressions to identify sentiments like Happy, Angry, Disgust, etc. As of late, neural network techniques are well known to demonstrate the perplexing interaction between images. The authors of improved methods for Faster CNN which are well known as to be Transfer Learning Techniques.

CHAPTER-3

PROPOSED SYSTEM

3.1 SYSTEM ARCHITECTURE

The figure 3.1 below explains the System Architecture of this project that deals with the three modes of data, i.e., Text, audio & video and essential working of the Web Application of the Sentiment Analysis. The system is designed to provide sentiments using text, audio, and video. The ends of the system consist of a list of emotions that can be predictedusing any of the three models with probabilities assigned to each one of them individually.

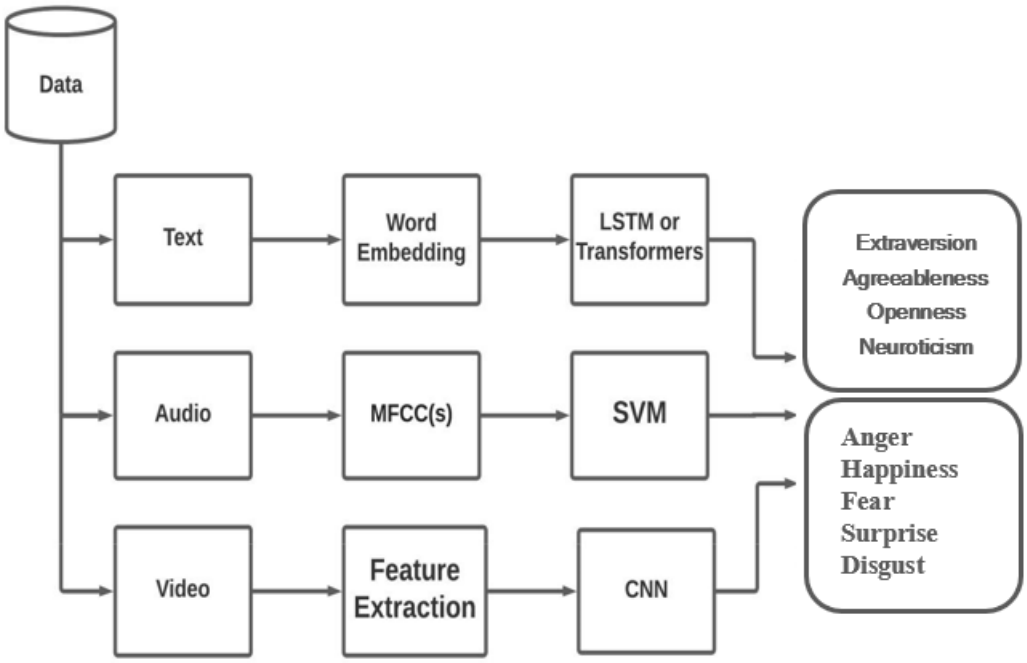


Figure 3.1: System Architecture

The walkthrough of the Architecture is as follows:

- The Text Data is cleaned and pre-processed. This Bag Of Words or Embedding Matrix is created to send it to the LSTM model that will predict the label or the maximum probability of sentiment in the text.
- The Audio Data is cleaned and pre-processed. Using this audio, Model calculated the Spectrograms or MFCC, gave it to Neural Networks of Classification models, respectively, and predicted the label accordingly.
- The Video Data is cleaned and pre-processed. After this,landmark points are extracted, which then is used by the Transfer Learning Techniques to predict the label.

3.2 WORD EMBEDDINGS:

The text data to be predicted for sentimental analysis is provided to the Word Embeddings module, which is equipped for catching the setting of a word in an archive, semantic and syntactic likeness, connection with different words.

3.3 MFCC's :

In solid preparation, the mel-recurrence cepstrum (MFC) is a depiction which shows momentary force span of a sound in light of a direct cosine change of a log powerrange on a nonlinear mel size of recurrence.

Mel-recurrence cepstral coefficients (MFCCs) are basically some coefficients that aggregatelymake up an MFC. They can be obtained from a kind of cepstral display of the brief snippet (anonlinear "range-of-a-range").

The difference between the mel-recurrence cepstrum and cepstrum is that the recurrence groupsare separated similarly on mel scale that resembles the reaction of body hearable frameworks more closely than the recurrence groups that are divided straight utilized in conventional range.For example, in sound pressure the distortion in recurrence can take into account improved portrayal of sound.

MFCCs are decided by means of following:

- For a sign, take Fourier transform.
- Guide the forces of the range got above onto the mel scale, utilizing three-sided covering windows or, on the other hand, cosine surrounding windows.
- For each mel frequencies make a log of the forces.
- Of the rundown of log powers of mel, calculate the discrete cosine, assuming it isanything but a sign.

3.4 TRANSFORMERS :

Sequence-to-Sequence architecture is a neural network which changes a specified succession of components, like grouping words in a sentence, another grouping. These models are admissible for interpretation, in which the grouping in words from one language is changed to a series of different words in some other dialect.

CHAPTER-4
THEORETICAL BACKGROUND

The point is to foster a model which is ready to provide real-time live sentiment with a visual UI developed with HTML,Js and TensorFlow. Consequently, the proposed model have chosen to isolate three kinds of information sources:

1. Textual Information: It has been developed to interview an individual that will helps to determine the Personality Traits of the individual. Model can also get these using a coverletter of an individual and analyze them accordingly.

2. Audio Information: It has been developed to take audio input of about 15 sec and visualize the sentiments like Angry, Happy, Disgust, Sad and Neutral over the period.This can be used in customer satisfaction detection after the call gets ended in the Call Centres.

3. Video Information: It will take an individual's live video feed and helps to identify the sentiment in a live form using a webcam.

4.1 DATASET SOURCES

TEXT:

For the text input, data which was gathered in a study by King and Pennebaker [19]. It has 2,468 daily writing submissions given by 34 psychology [8] scholars (five men and 29 women from 18 to 67 years of age).

AUDIO:

For sound informational collections, model using the "Ryerson Audio-Visual Database of Emotional Speech and Song". RAVDESS contains 7356 voice clips (size: 24.8 GB). These records contain 24 audio clips (12 females, 12 guys), showing two lexically coordinated explanations in a nonbiased North American speech. Discourse incorporates quiet,glad, miserable, sore, unfortunate, shock, and repugnance articulations, and the tune contains quiet, cheerful, dismal, sad feelings.

VIDEO:

For the video informational collections, model utilizing the well-known FER2013 Kaggle Challenge informational index. The information comprises 48x48 pixel grayscale pictures of countenances. The informational collection remains very testing to use since thereare vacant pictures or wrongly ordered pictures.

4.2 DATA PRE-PROCESSING

This comprises two different variety of data namely Audio and Video. Model will discuss the pre-processing of all the data formats.

4.2.1 TEXT PRE-PROCESSING:

The pre-processing is the initial step of this NLP pipeline. This is the place where model convert crude content records to cleaned arrangements of words. To finish this interaction, model first need to tokenize the corpus. This implies that sentences are parted into a rundown of single words, likewise called tokens. Other pre-processing steps remember using standard articulations for a request to erase undesirable characters or reformat comments. At last, there are strategies accessible to supplant words by their linguistic root: the objective of both stemming and lemmatization is to decrease derivationally related types of a comment to a typical base structure.

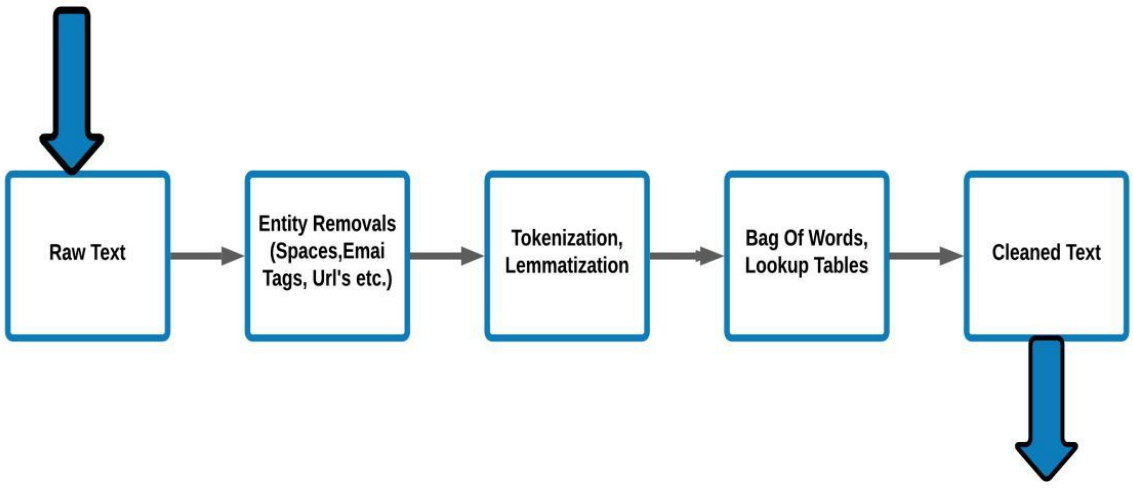


Figure 4.1: Text Cleaning Pipeline

Figure 4.1 explains the Text Cleaning Pipeline and how the text is converted to its basic stem and fedto the model for the training and testing purposes.

4.2.2 AUDIO PRE-PROCESSING:

To begin with, before starting feature extractions, it's fitting to apply a pre-emphasis filter on the sound sign to intensify every one of the significant frequencies. After the pre-emphasis filter, model need to part the sound sign into transient windows called frames. Model duplicate eachcase by a Hamming window work in the wake of parting the movement into different casings.It permits decreasing spectral spillage or any sign discontinuities and working on signal lucidity.

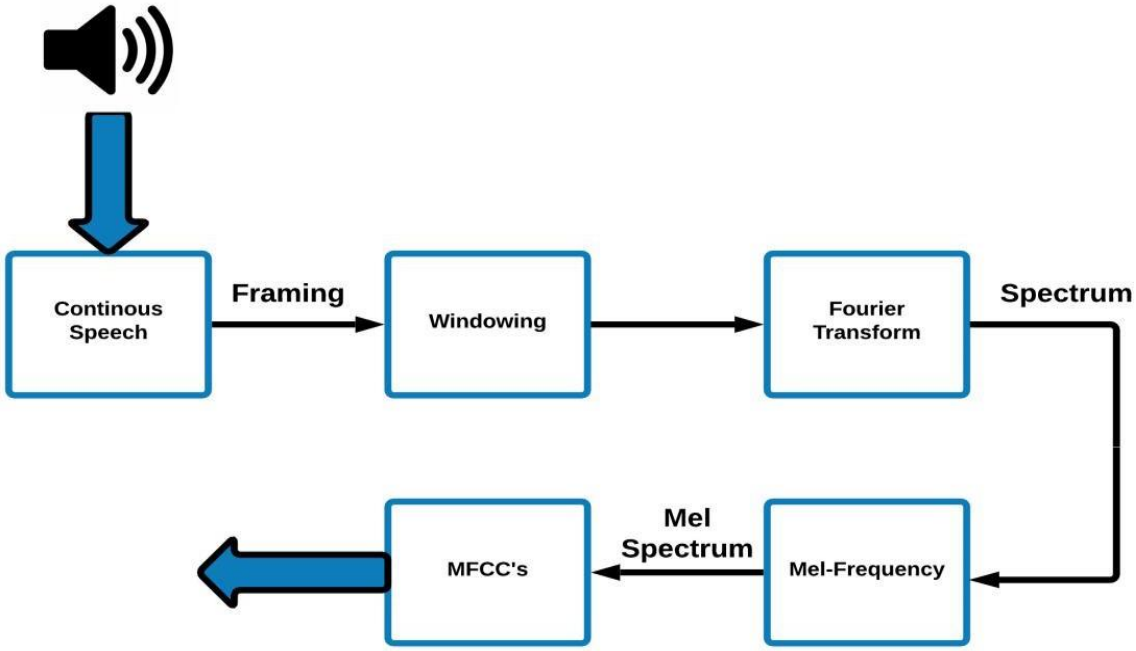


Figure 4.2: Audio Pre-processing

This system explains the Audio Cleaning and conversion of those into the MFCC's that will be used asthe input for the model and is used for training and testing purposes in Figure 4.2

4.2.3 VIDEO PRE-PROCESSING:

Starting by analyzing the video frame by frame, then applying filters using some of the convolution techniques and making fewer inputs to identify the face then and adequately zoomon it, reducing pixel density to the same pixel density as that of the train set. Getting landmarkspoints is a part of feature extraction that is processed during this stage. Model are transforming the input image to a model readable input to predict the emotion of the information.

CHAPTER-5

MODULE DESCRIPTION

5.1 LANGUAGES AND LIBRARIES

- OpenCv : OpenCv is a ML package library and associates ASCII text file laptop vision. It's a library of programming functions chiefly geared toward period laptop vision. Model have used this mostly in my video phase, where most of the work of face detection and augmentation is done by this library.

- HTML: HTML or HyperText Markup Language is a markup language that allows web users to create and structure various parts of a web page such as headers, tables and links using elements, tags, and attributes. It tends to be helped by innovations like Cascading Style Sheets (CSS) and prearranging dialects like JavaScript . This would be based on the language for my website that would create to deploy my threemodels of Sentiment Analysis.

- NumPy : It is a open source Python library. NumPy works with Python objects called multi-dimensional arrays. Arrays are basically collections of values, and they have one or more dimensions. NumPy array data structure is also called *ndarray*, shortfor n-dimensional array. Datasets are usually built as matrices and it is much easier to open those with NumPy instead of working with lists. Numpy here is used for many processes. This library does all the mathematical computation in my project.

- Tensorflow: Tensorflow is an open source framework. It was initially designed to be a neural network library but with advancement it can perform much more functions.It is a machine learning library. It is the base library that model have used to create this model; this is the cover of Keras that helped in model designing and fitting the values.

5.2 SETUP USED

- Flask: It is a small net framework written in Python. It's classified as a microframework. As a result of it doesn't need explicit tools or libraries. It's no information abstraction layer, type validation, or the other parts wherever pre-existingthird- party libraries give standard functions. It has been used to create an interface between the website and models and also is responsible for returning the HTML pagesaccordingly to the output.

- Google Colab :Colab is a free Jupyter notebook climate that runs altogether in the cloud. In particular, it doesn't need an arrangement, and the notebooks that user make can be at the same time altered by user colleagues - how user vary reports in Google Docs. The free GPU of Colab is used in this project for the training purpose of the Video and Audio modes for providing fast results.

- Spyder :It is a free and open-source logical climate written in Python and planned by and for researchers, specialists, and information examiners. It includes a remarkable mix of an exhaustive advancement instrument's high-level altering, examination, troubleshooting, and profiling usefulness. This will work on allmy python files, and all the mathematical work is done over here.

- VS Code :VS Code is a lightweight text editor, one of the best for coding in all most all languages. It provides user to code in any programming language for examplePython, Java, C++, JavaScript, and more. Visual Studio Code is a source code editor, which helps businesses build and debug web applications running on Windows, Linux, and macOS. It is a source-*code* editor text editor program design.

5.3 TEXT SENTIMENT

Text modal used the Pennebaker and King dataset for Text Sentiment Analysis that usually predicts the Personality Traits that model will use to check over an individual that can be used inan interview process. Sentiment Analysis is always a difficult task as the machine cannot understand humor, anger, happiness, and sadness. Day by day, NLP is growing, and model getting many models that are improving and solving this problem. Initially, RNN models were used, but the problem was that it could not see the future data as a word by word inputs were given to the model. Thus, new models came up like the LSTM's, Bidirectional LSTM's, and Transformers. Model used Bidirectional-LSTM's in the process that helped to improve the accuracyand decision by the model.

The steps to go through this module are:

1. First of all, the text is cleaned, and unnecessary words are removed using the Tokenization method, and all symbols are removed, and the whole text is made in lower-case.

2. Then model will create a Bag Of Words that will contain the vocabulary size i.e., most ofthe words used in the data.

3. Embedding Matrix is created which is the strong relationship of words that are nearbylike King and Queen, or Apple and Mango are strongly related.

4. This embedding matrix data is put as an input to the Attention Based Model that model will custom create with Bidirectional LSTM Encoders, Attention Layer, and the Decoders.

5. Many to One LSTM's are used to predict the label using the text.

5.4 AUDIO SENTIMENT

Audio modal used the RAVDESS data for the Audio Sentiment Analysis. It uses 15- second audio provided by the user in the portal; the runtime is less for less computational work as training and handling the audio in small chunks is a significant improvement for the predictions.

Literature is centered on just around six feelings., happy, sad, angry, disgusted,fear, and surprise. Albeit the feeling classifications are more plentiful and complex, in actuality.

The steps that model went through this module were:

1. Extract 15 seconds audio and add some noise to the data so that model can also be used in the real-life process.

2. Signal Pre-processing will be done in the next stage, like amplifying high-frequencyand splitting audio in frames.

3. After all this MFCC' is calculated, which are the input data that will be used for themodel.

4. Classification models can be used to predict one of the six labels of sentiment.

5. Printing a bar plot of the sentiments achieved by using Argmax computation.

5.5 VIDEO SENTIMENT

The work that is done on the Facial Expressions has been trained over FER2013 KaggleChallenge dataset and has obtained a good accuracy while using the Xception transfer learning model.

1. First of all, the video is split into frames, and the analysis is done step by step.

2. Filters are used after getting the frames, and Convolution Operations are performed.

3. Features Extraction is done, and landmark points are located in those frames

4. The image is flattened and fed to the Exception Model for an output.

5.6 DEPLOYMENT

The project's primary purpose is to have a place where model can test all the capabilities of a project. This deployment is the last stage that will helps to do this work.

The website on the local server that will run all the three modules of this project, i.e., Audio, Text, and Video. All the three models that we have created during the training time willbe used up by Flask which will helps to run our project on Local Network so that all the dependencies can be used in one go.

All the steps discussed here were implemented and below are the results of that implementation with the final local web server.

CHAPTER-6
IMPLEMENTATION AND TESTING

6.1. INTERFACE

The Web-App has deployed all three models in a local server and ran it using Flask; each of the Modes' results is present below. Figure 6.1 shows the Home Page of the deployed model in the local server.

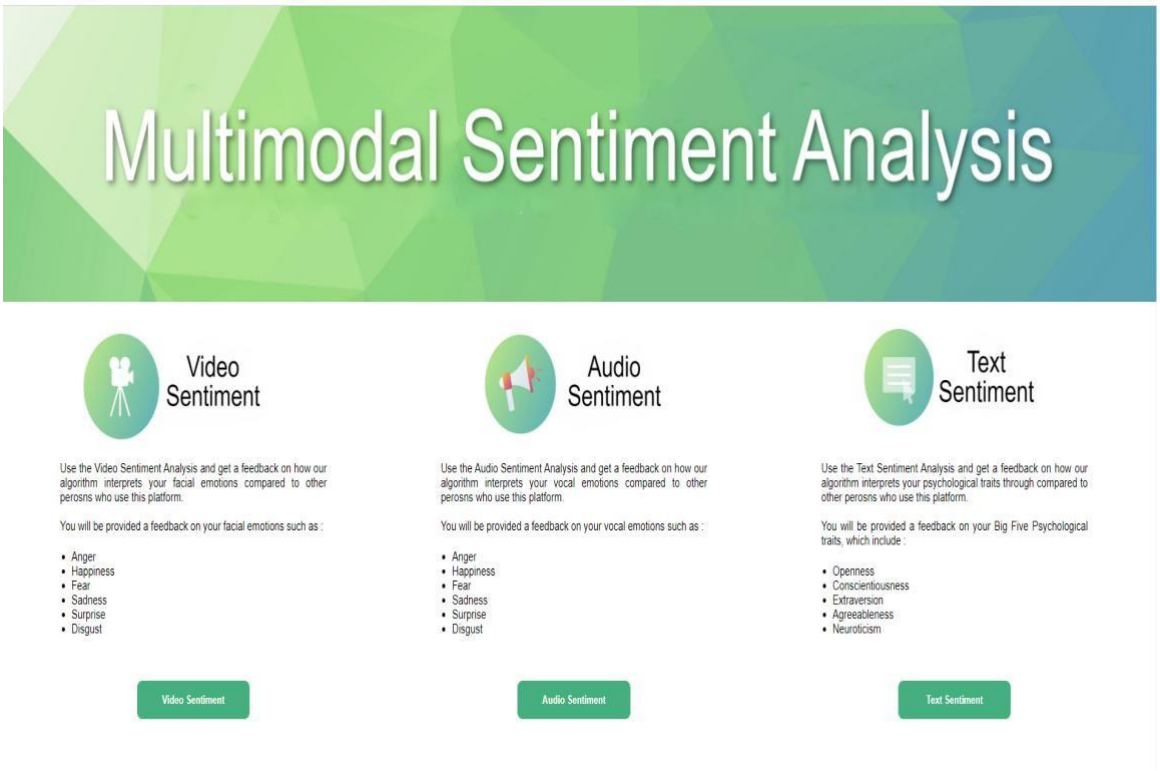


Figure. 6.1: Home Page

The Web-App is to be designed with three sections with Text, Audio, and Video SentimentAnalysis. The user will type in the Text Sentiment Analysis, which will use the LSTM techniques to predict the Sentiment of the data by a particular label that has been defined during the training. The Audio sections take the audio file as input in a .wav file and predictthe Sentiment by calculating the MFCC's and predicting the label used in training. In the video section, real-time camera access is needed for the input of the Sentiment Analysis, and the facial expressions determine the Sentiment.

6.2 SENTIMENT ANALYSIS

6.2.1 TEXT-SENTIMENT

In this Text Modal, model have implemented Text Analysis for predicting the Personality Traits in a human being used for interview simulation [4]. Model can help finalize the candidate in an interview.

The dataset that has been used is by Pennebaker, and King for training and testing purposes. Two options are added one a Dialogue Box and one Pdf upload that will helps to identify the Personality of an individual and compare it with other candidates by plotting a bar graph.

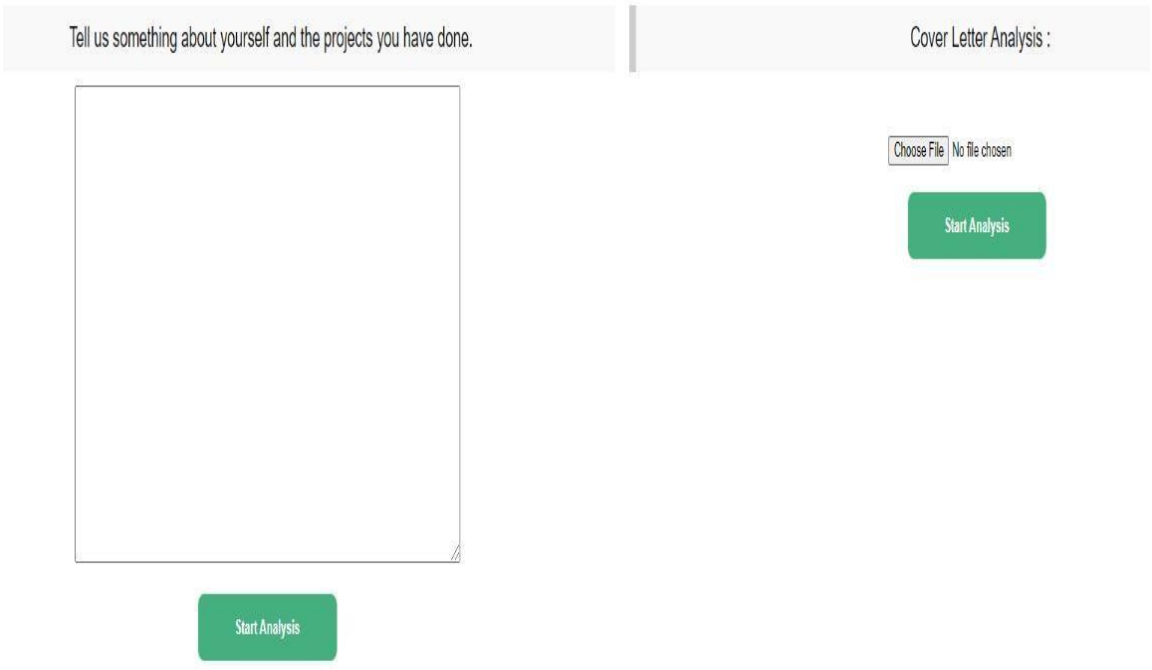


Figure. 6.2: Text Sentiment Home-Page

Figure 6.2 shows the two methods we can use in the Text-Sentiment, i.e., Text [15] and Cover Letter upload. Compared with the other candidates, the output bar plots are displayed, andthe most common words that appear in the text are also shown on the sidelines. The predicted probability percentage is shown beside the bar plots.

The bar plots with probability are shown below:

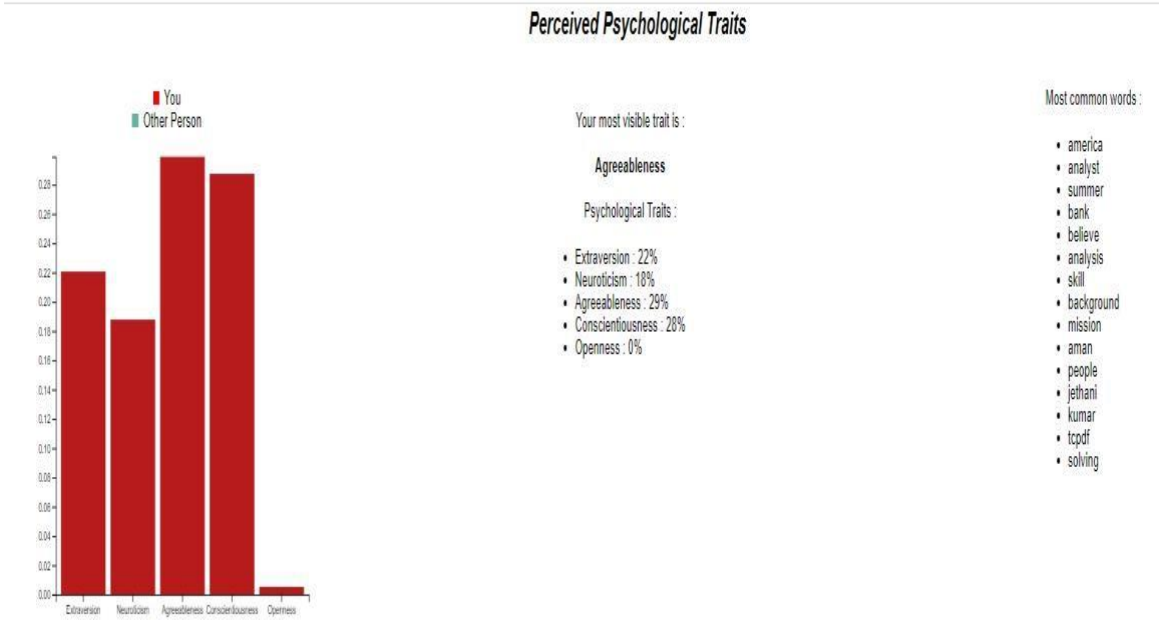


Figure. 6.3: Probability Bar Plot for Our Input Text

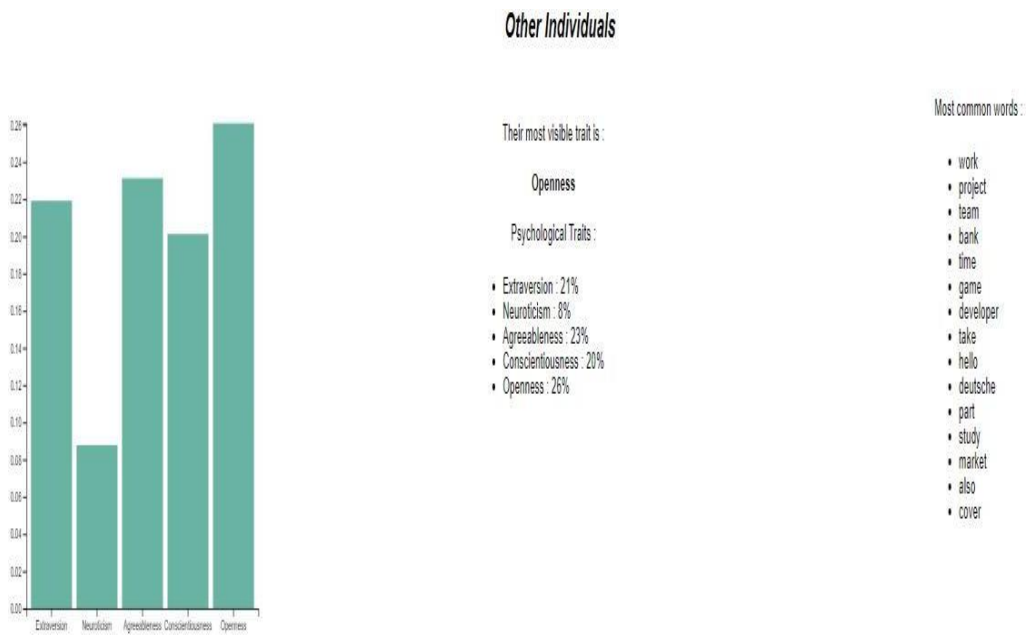


Figure. 6.4: Probability Bar Plot For Other Individuals

Figure 6.3 and 6.4 gives the label prediction, i.e., the emotion with the highest probability of our text input and the comparison with other individuals, respectively.

The accuracy by using different models is shown below. The method that has been used is Word-2-Vec embedding with LSTM and SVM models. Both the accuracy of the test set isshown below.

Model	EXT	NEU	AGR	CON	OPN
Word2Vec + SVM	46.18	48.21	49.65	49.97	50.07
Word2Vec + LSTM	55:07	50:17	54:57	53:23	53:84

Table 6.1: Text Accuracy Confusion Matrix

Table 6.1 shows the accuracy of labels with two different types of models. LSTM helped to increase the accuracy because LSTM is used as a Bidirectional and can see any independence of the current word with the future.

6.2.2 AUDIO-SENTIMENT

In this Audio Modal, model have implemented Audio Analysis to predict the Sentiment that takes the live audio of about 15 seconds and runs its prediction on that limited audio. The MFCC and Power-Spectogram are calculated and used in the Neural Networks or classification models.

The labels that are predicted using the Audio-Sentiment are Angry, Happy, Neutral, Sad, Disgust and Fear and it also plots a bar plot in the results of all the emotions perceived. The dataset that have been used "Ryerson Audio-Visual Database of Emotional Speech andSong"(RAVDESS) dataset for training and testing purposes.

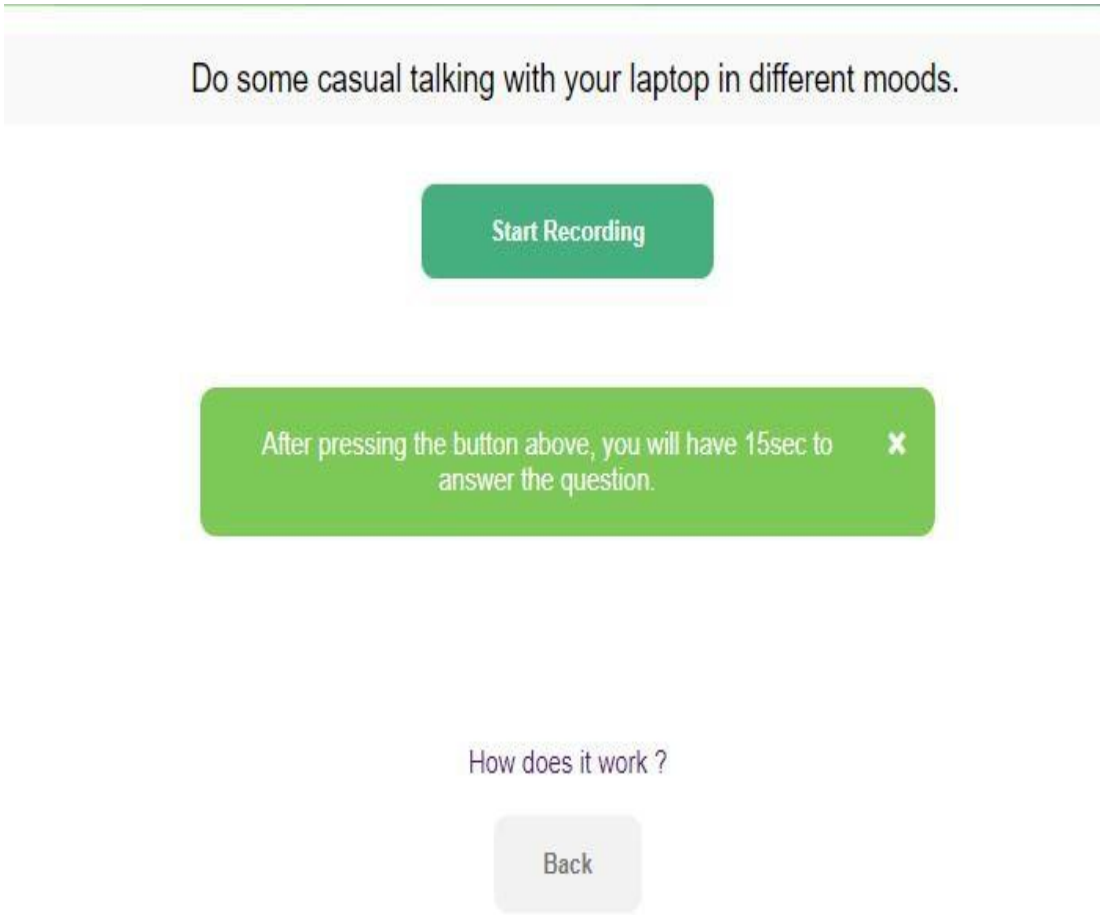


Figure 6.5: Audio Sentiment Home Page

As soon model click Start Recording as seen in Figure 8 in the Audio Home-Page, it starts running for 15 seconds. As the time is completed, it shows a button Get Emotion Analysis for the results.

After clicking on getting Analysis, model can see the output bar plots and compare them to how a particular person shows emotions in the audio.

The predicted probability percentage is shown beside the bar plots. The image below has the emotion analysis for the last two audios that were played while testing the web app.

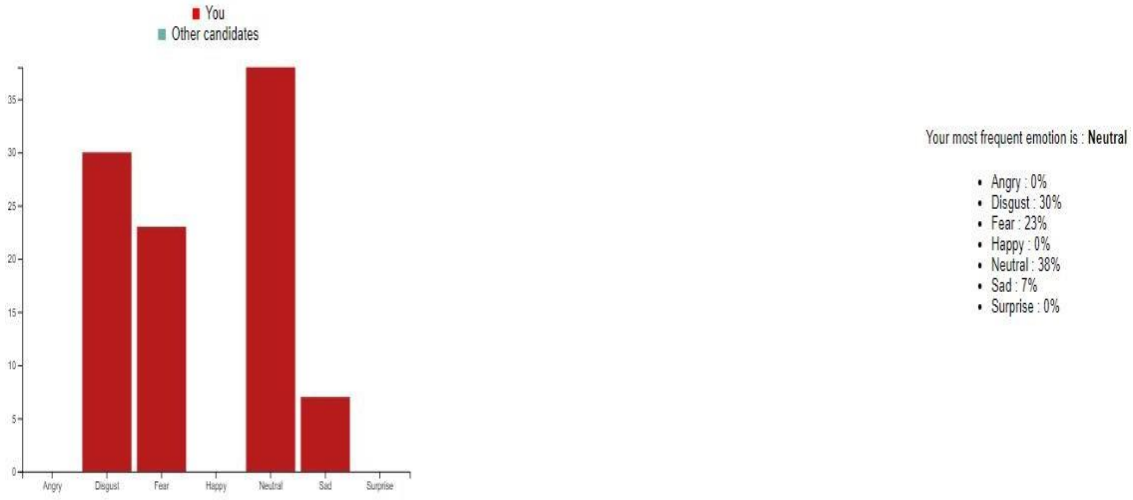


Figure. 6.6: Label Prediction and Bar Plot For Our Audio

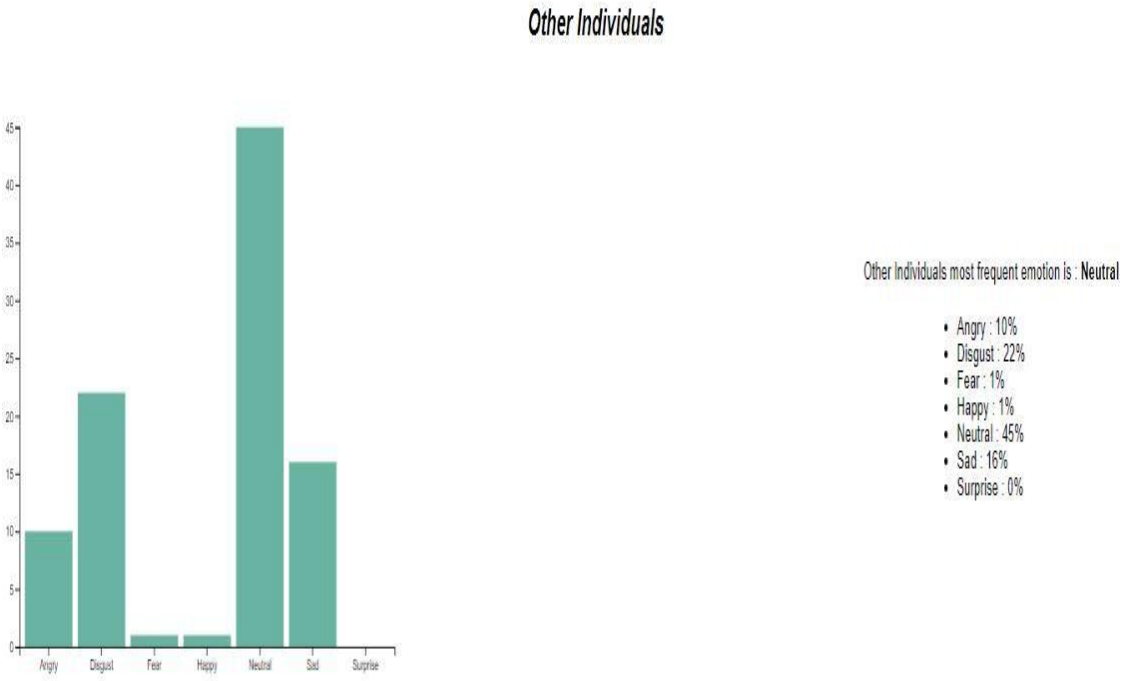


Figure. 6.7: Label Prediction and Bar Plot Of Others Audio

Figure 6.6 and 6.7 gives the label prediction, i.e., the emotion with the highest probability of our audio input and the comparison with other individuals, respectively.

This Audio modal have been implemented with MFCC's calculation and then fed those MFCC's to the Classification Network using the Neural Networks. The confusion matrix accuracy of each label is given below.

		Predicted labels						
		Happy	**Sad**	**Angry**	**Scared**	**Neutral**	**Disgusted**	**Surprised**
Actual labels	**Happy**	80.0%	0.0%	5.7%	5.7%	5.7%	2.9%	3.4%
	Sad	8.1%	81.1%	0.0%	0.0%	2.7%	8.1%	1.5%
	Angry	6.3%	6.3%	75%	0.0%	6.3%	6.3%	0%
	Scared	6.7%	0.0%	4.4%	71.1%	8.9%	8.9%	4.7%
	Neutral	11.1%	5.6%	2.8%	8.3%	66.7%	5.6%	0.3%
	Disgusted	0.0%	8.7%	0.0%	4.3%	2.2%	84.8%	2.9%
	Surprised	0.0%	8.7%	0.0%	4.3%	2.2%	84.8%	67.3%

Table 6.2 : Audio Accuracy Confusion Matrix

Table 6.2 shows the accuracy of all labels using MFCC's fed to some of the classification methods with the use of Neural Networks.

The Audio model's accuracy and loss graph plot is shown below, and the Final Accuracy can be seen from them predicting those six labels.

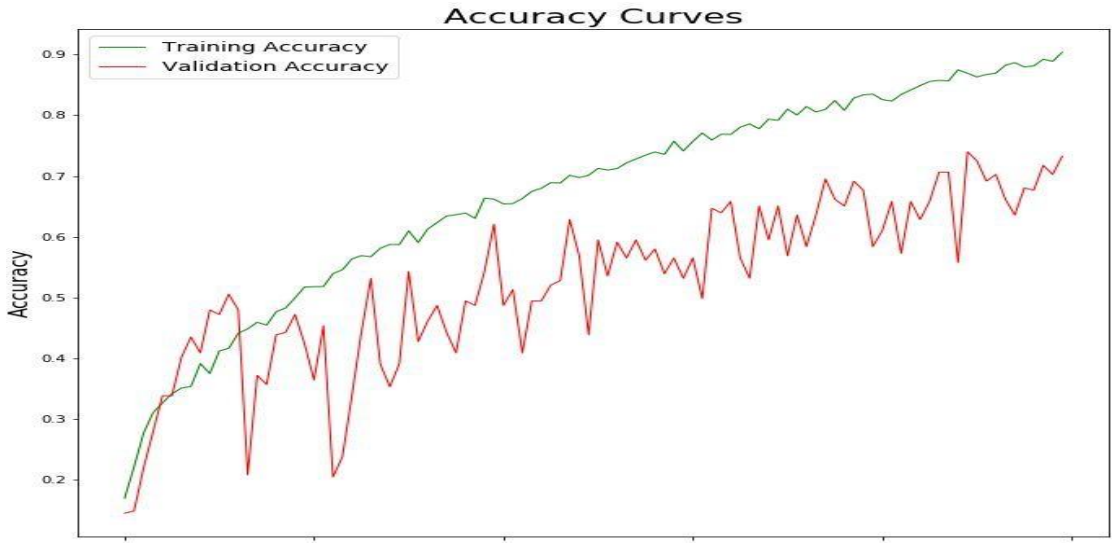


Figure. 6.8: Audio Sentiment Accuracy Curve

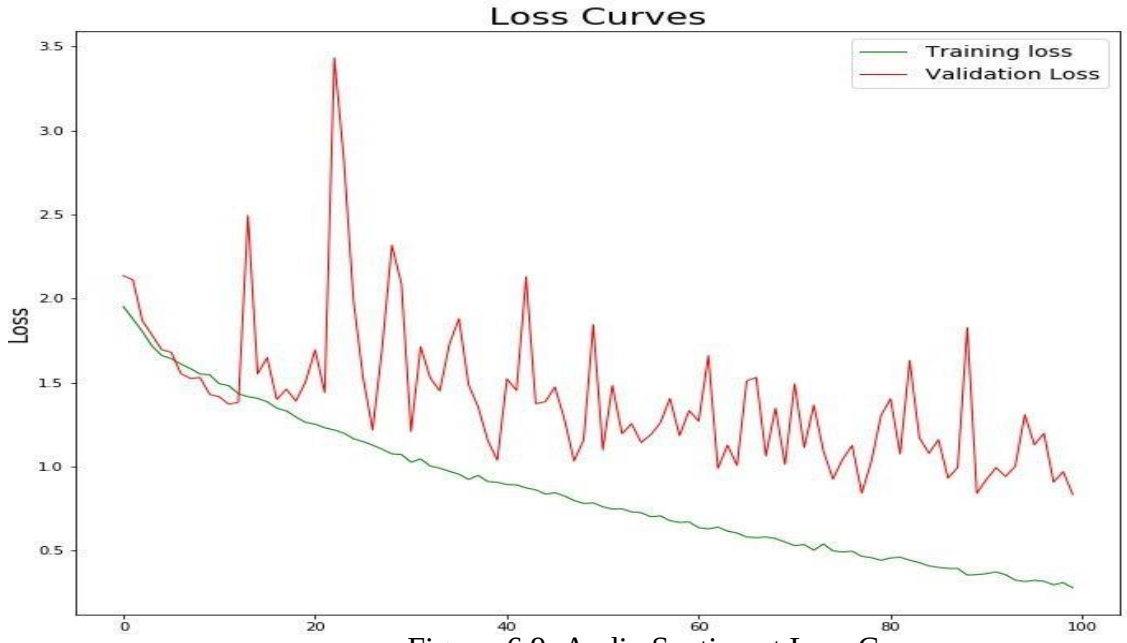


Figure. 6.9: Audio Sentiment Loss Curve

Note: Keras Early Stopping made the graph stops at 103 Epochs as there was no improvement in the accuracy.

This model presents reasonably satisfying results. This prediction recognition rate is around 75% for 7-way (happy, sad, angry, scared, disgust, surprised, neutral) emotions.

6.2.3 VIDEO-SENTIMENT

In this Video Modal, model have implemented Video Analysis for predicting the Sentiment that takes the live webcam feed and runs its prediction on that live video, detects emotions, and identifies the number of faces [19]. The process is simple; the video is brokeninto frames. Each frame is convolved using filters, and landmarks points are obtained usingthat filtered image to predict sentiments.

The labels that are predicted using the Video-Sentiment are Angry, Happy, Neutral, Sad, Disgust and Fear and it also plots a bar plot in the results of all the emotions perceived. It also tells emotions in a line chart throughout 45 sec. The dataset that has been used is FER2013 Kaggle Challenge dataset for training and testing purposes.

Figure 6.10 shows the Home Page for Video Analysis and has a start recording button that takes to the new page where sentiment analysis is done on a live webcam, as shown in Figure 6.11 and Figure 6.12, respectively.

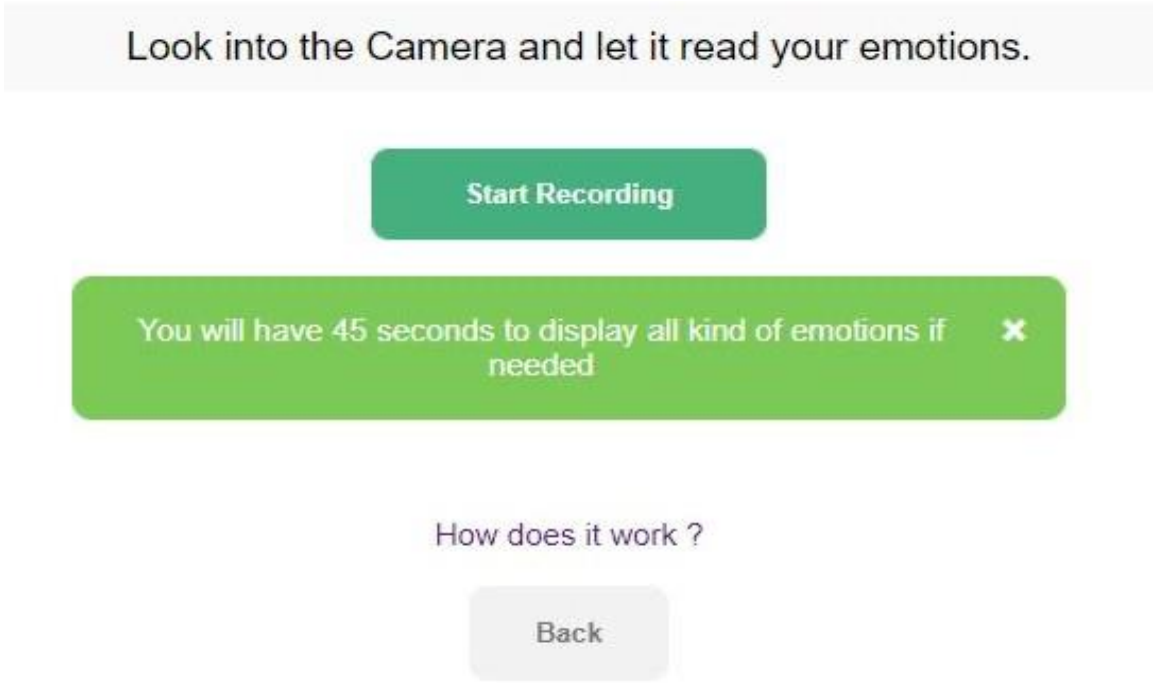


Figure. 6.10: Video Sentiment Home-Page

As soon model click Start Recording in the Video Home-Page it starts running for 45 secondsand moves to a another window where live webcam emotions can be detected. The images of the live emotion detection are shown below.

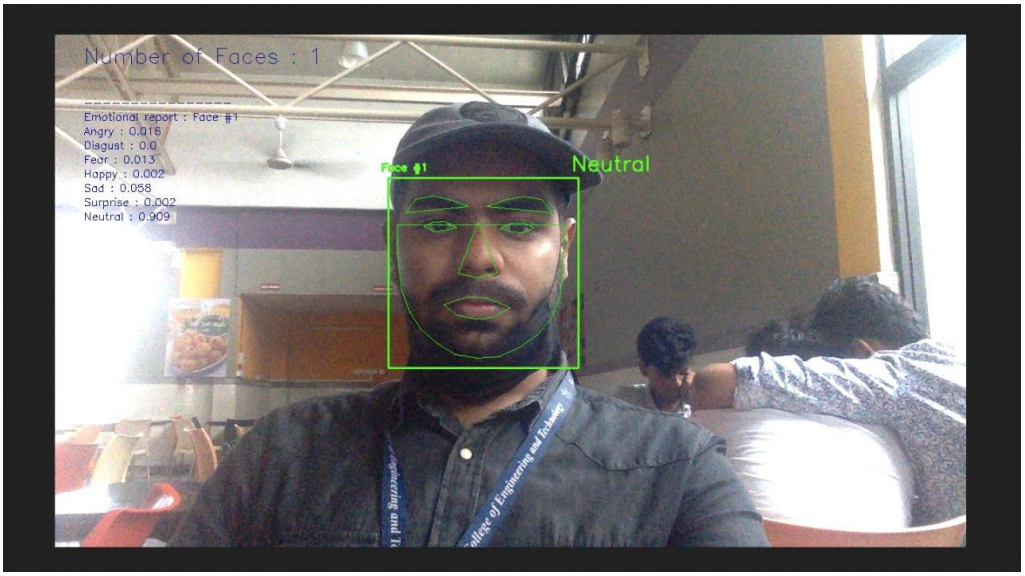


Figure. 6.11: Emotion Detected(Neutral)

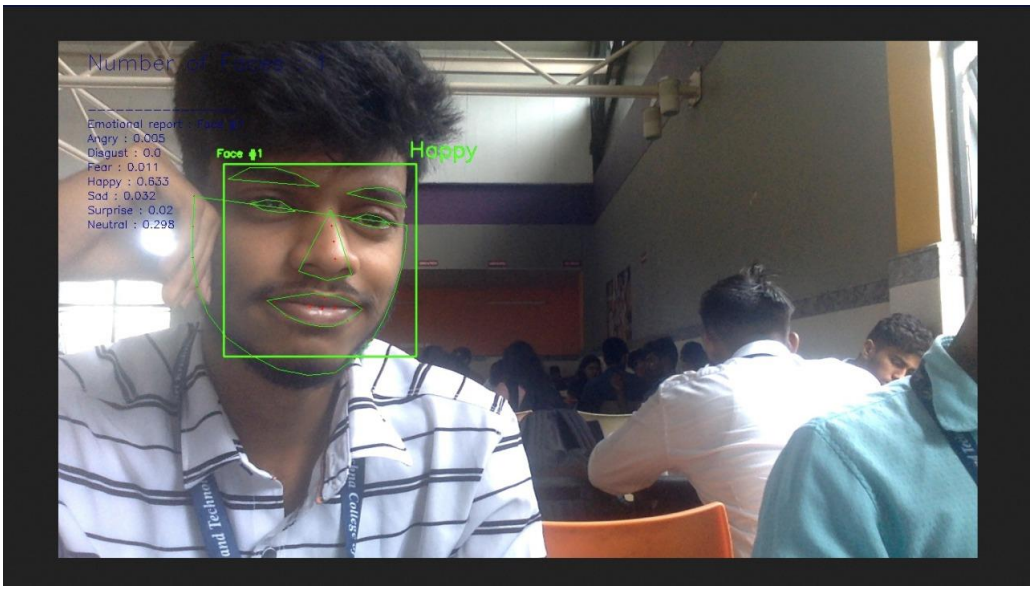


Figure. 6.12: Emotion Detected(happy)

The figure above shows the emotions in the green box by using the positions of the landmarks and thus making of call for an emotion.

After the video is over recording, model move to the next page with the bar plots with the probability of the expressions over the period and a line chart that shows how our emotions have varied.

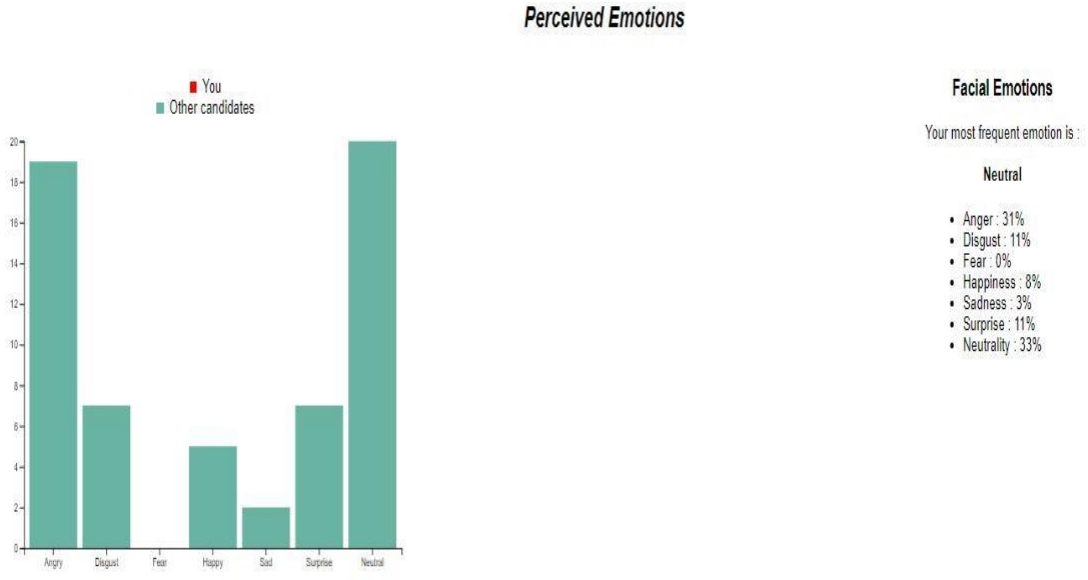


Figure. 6.13: Probability Bar Plot For Our Input Live Video

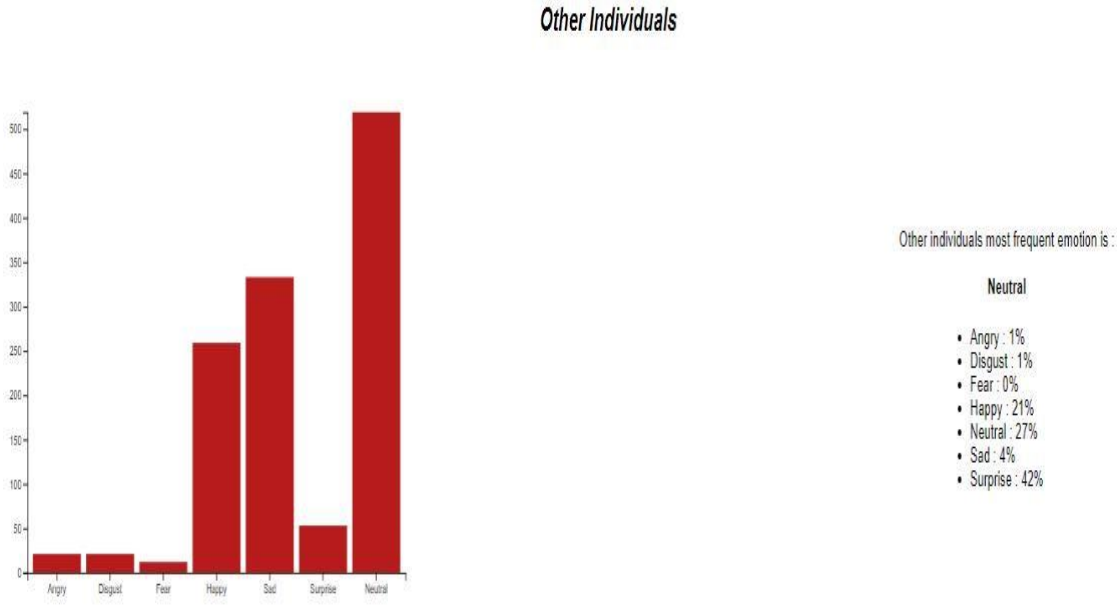


Figure. 6.14: Probability Bar Plot Of Other Individuals

Figure 6.13 and 6.14 gives the label prediction i.e. the emotion with highest probability of our video feed and also comparison with other individuals respectively.

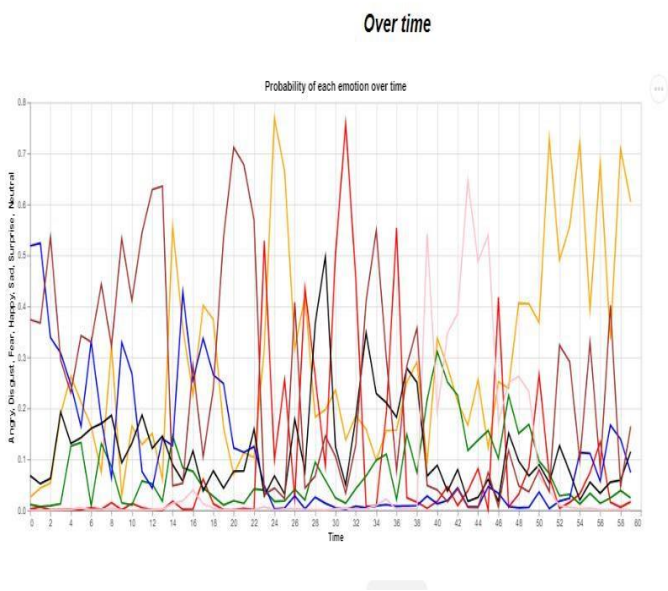


Figure. 6.15: Line Chart for Varying Emotions

Figure 6.15 shows how emotions vary concerning the time using a line chart that can be used in the long run to get the mean Sentiment. Model have used the Xception model that is a Transfer Learning Model and is used in competition for predictions of the 1000 labels.

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 48, 48, 32)	320
max_pooling2d_2 (MaxPooling2	(None, 24, 24, 32)	0
batch_normalization_1 (Batch	(None, 24, 24, 32)	128
conv2d_3 (Conv2D)	(None, 22, 22, 32)	9248
max_pooling2d_3 (MaxPooling2	(None, 11, 11, 32)	0
batch_normalization_2 (Batch	(None, 11, 11, 32)	128
conv2d_4 (Conv2D)	(None, 11, 11, 32)	9248
max_pooling2d_4 (MaxPooling2	(None, 5, 5, 32)	0
conv2d_5 (Conv2D)	(None, 5, 5, 32)	9248
flatten_1 (Flatten)	(None, 800)	0
dense_1 (Dense)	(None, 512)	410112
dense_2 (Dense)	(None, 7)	3591

Total params: 442,023
Trainable params: 441,895
Non-trainable params: 128

Figure. 6.16: Figure for Varying Emotions

Figure 6.16 shows the Keras Xception model summary and all the layers that have been used.

The accuracy and loss graph for that model is shown below.

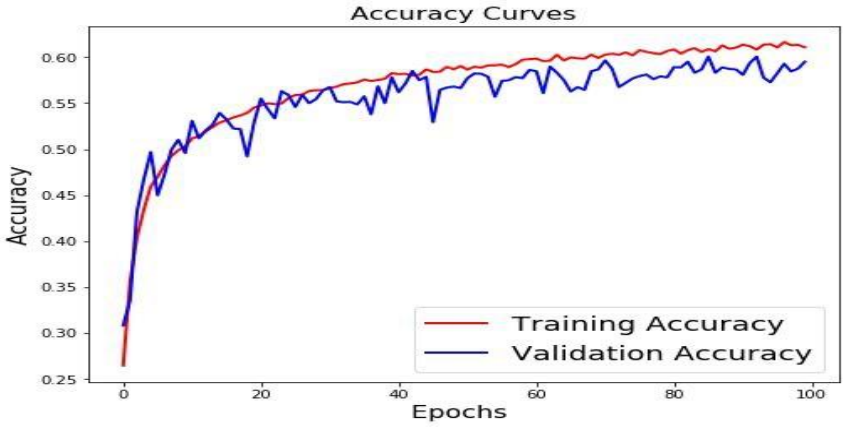


Figure. 6.17: Xception Accuracy Graph

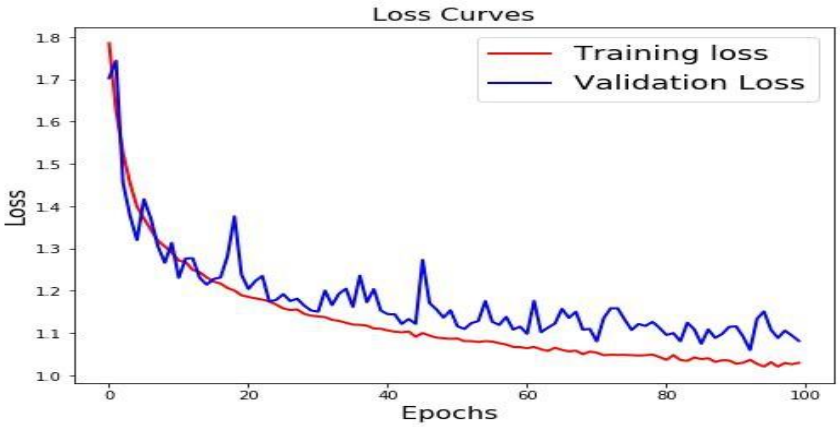


Figure. 6.18: Xception Loss Graph

Note: Keras Early Stopping made the graph stops at 100 Epochs as there was no improvement in the accuracy. Figure 6.17 and Figure 6.18 show the trend of the Accuracy and Loss of the trained and tested model using the Xception Transfer Learning.

This modal was also tried on different lengths of videos like 15sec, 30sec, 40sec, but there was no significant impact on the accuracy, so model only implemented it on 45sec.

6.3 APPLICATIONS

Social media monitoring - Social media posts often contain some of the most honest opinions about user products, services, and businesses because they're unsolicited. With the help of sentiment analysis software, user can wade through all that data in minutes, to analyse individual emotions and overall public sentiment on every social platform.

Customer support ticket analysis - Sentiment analysis with natural language understanding (NLU) reads regular human language for meaning, emotion, tone, and more, to understand customer requests, just as a person would. User can automatically process customer support tickets, online chats, phone calls, and emails by sentiment to prioritize any urgent issues.

Business Intelligence Buildup - Sentiment analysis enables user to determine how user product performs in the market and what else is needed to improve user sales. User can also analyze the responses received from the competitors. Based on the survey generated, the companies can satisfy customers needs in a better way.Immediate decisions that will help the user adjust to the present market situation.

Market Research and Analysis - Business intelligence uses sentiment analysis to understand the subjective reasons why customers are or are not responding to something, whether the product, user experience, or customer support. Sentiment analysis will enable user to have all kinds of market research and competitive analysis. It can make a huge difference whether user are exploring a new market or seeking an edge on the competition.

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

With the rapid development of social media, multimedia data has become an important carrier of human sentiments and opinions. Sentiment analysis for the multimedia content will be a huge game changer. The proposed web based application aids in determining a person's moods. When employed in all forms of communication, including text, audio, and video, it is helpful for researchers It is worth mentioning again that model target real-time sentiment monitoring for retail businesses.

The most prominent methods for visual sentiment analysis, and discussed the prevalent approaches for multimodal sentiment analysis and indicated the proposed model will have a promising impact in the future directions in this area.

FUTURE SCOPE:

The proposed model can improve the mode of Text sentiment analysis by using BERT techniques, and the Audio sentiment field can beimproved by combining multiple techniques like HMM, CNN, and MFCC, together to produce an efficient analysis. The proposed model can give more accurate results if it uses two modes at once like Audio and Video analysis together to get better accuracy for the predicted labels respectively. This analysis should also be augmented to take into account the temporal contingency between these vocal, visual and verbal behaviors. Furthermore, Deep Learning-based multimedia sentiment analysis will still be a hot topic.

CHAPTER-8

REFERENCES

1. "Mfcc(s)," https://en.wikipedia.org/wiki/Mel-frequency_cepstrum.

2. "The facial emotion recognition challenge from kaggle," https://www.kaggle.com/

3. Association for Computing Machinery, 2016, pg: 4647–4657.

4. E. Sariyanidi, H. Gunes, and A. Cavallaro, "Automatic analysis of facial affect: A survey of registration, representation, and recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 6, pg: 1113–1133, 2015.

5. Fast, B. Chen, and M. S. Bernstein, "Empath: Understanding topic signals in large-scale text," in Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ser. CHI '16. New York, NY, USA.

6. M. Chen, S. Wang, P. P. Liang, T. Baltruaitis, A. Zadeh, and L.-P. Morency,"Multimodal sentiment analysis with word-level fusion and reinforcement learning,"Proceedings of the 19th ACM International Conference on Multimodal Interaction, 2017.

7. Mandera, E. Keuleers, and M. Brysbaert, "How useful are corpus-based methods for extrapolating psycholinguistic variables?" Quarterly Journal of Experimental Psychology, vol. 68, no. 8, pg: 1623–1642, 2015

8. N. Pappas, M. Redi, M. Topkara, B. Jou, H. Liu, T. Chen, and S. Chang,"Multilingual visual sentiment concept matching," 06 2016

9. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and Polosukhin, "Attention is all you need," CoRR, vol. abs/1706.03762, 2017. Pham,P. Liang, T. Manzini, L.-P. Morency, and B. Poczos, "Found in translation: Learning robust joint representations by cyclic translations

10. Pennebaker-king," https://sites.google.com/michalkosinski.com/mypersonality

11. Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pg: 6892–6899, 07 2019

12. Q. You, J. Luo, H. Jin, and J. Yang, "Robust image sentiment analysis using progressively trained and domain transferred deep networks," 09 2015
13. The ryerson audio-visual database of emotional speech and song (ravdess)," https://smartlaboratory.org/ravdess/

14. V. Campos, A. Salvador, B. Jou, and X. Giró-i Nieto, "Diving deep into sentiment: Understanding fine-tuned cnns for visual sentiment prediction," 10 2015

15. W. W. Lo, X. Yang, and Y. Wang, "An xception convolutional neural network for malware classification with transfer learning," in 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2019

16. Word embeddings," https://en.wikipedia.org/wiki/Word_embedding.

17. Y-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov,"Multimodal transformer for unaligned multimodal language sequences," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.

18. Zeng, X. Shu, Y. Wang, Y. Wang, L. Zhang, T. Pong, and H. Qu, "Emotioncues: Emotion-oriented visual summarization of classroom videos," vol. 27, pg: 3168–3181, 2021 Y.Jia and S. SungChu, ArXiv, vol. abs/2004.10320, 2020

19. Zeng, X. Wang, A. Wu, Y. Wang, Q. Li, A. Endert, and H. Qu, "Emoco:Visualanalysis of emotion coherence in presentation videos," IEEE Transactions on Visualization and Computer Graphics, pg: 1–1, 2019.

APPENDIX I

SPEECH RECOGNITION:

```
## Basics ##
import time
import os
import numpy as np

## Audio Preprocessing ##
import pyaudio
import wave
import librosa
from scipy.stats import zscore

## Time Distributed CNN ##
import tensorflow as tf
from tensorflow.keras import backend as K
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense, Dropout,
Activation, TimeDistributed
from tensorflow.keras.layers import Conv2D, MaxPooling2D,
BatchNormalization, Flatten
from tensorflow.keras.layers import LSTM


'''
Speech Emotion Recognition
'''
class speechEmotionRecognition:

    '''
    Voice recording function
    '''
    def __init__(self, subdir_model=None):
```

```
        # Load prediction model
        if subdir_model is not None:
            self._model = self.build_model()
            self._model.load_weights(subdir_model)

        # Emotion encoding
        self._emotion = {0:'Angry', 1:'Disgust', 2:'Fear',
3:'Happy', 4:'Neutral', 5:'Sad', 6:'Surprise'}


    '''
    Voice recording function
    '''
    def voice_recording(self, filename, duration=5,
sample_rate=16000, chunk=1024, channels=1):

        # Start the audio recording stream
        p = pyaudio.PyAudio()
        stream = p.open(format=pyaudio.paInt16,
                        channels=channels,
                        rate=sample_rate,
                        input=True,
                        frames_per_buffer=chunk)

        # Create an empty list to store audio recording
        frames = []

        # Determine the timestamp of the start of the response
interval
        print('* Start Recording *')
        stream.start_stream()
        start_time = time.time()
        current_time = time.time()

        # Record audio until timeout
        while (current_time - start_time) < duration:
```

```
            # Record data audio data
            data = stream.read(chunk)

            # Add the data to a buffer (a list of chunks)
            frames.append(data)

            # Get new timestamp
            current_time = time.time()

        # Close the audio recording stream
        stream.stop_stream()
        stream.close()
        p.terminate()
        print('* End Recording * ')

        # Export audio recording to wav format
        wf = wave.open(filename, 'w')
        wf.setnchannels(channels)
        wf.setsampwidth(p.get_sample_size(pyaudio.paInt16))
        wf.setframerate(sample_rate)
        wf.writeframes(b''.join(frames))
        wf.close()


    '''
    Mel-spectogram computation
    '''
    def mel_spectrogram(self, y, sr=16000, n_fft=512,
win_length=256, hop_length=128, window='hamming', n_mels=128,
fmax=4000):

        # Compute spectogram
        mel_spect = np.abs(librosa.stft(y, n_fft=n_fft,
window=window, win_length=win_length, hop_length=hop_length)) **
2

        # Compute mel spectrogram
```

```
        mel_spect = librosa.feature.melspectrogram(S=mel_spect,
sr=sr, n_mels=n_mels, fmax=fmax)

        # Compute log-mel spectrogram
        mel_spect = librosa.power_to_db(mel_spect, ref=np.max)

        return np.asarray(mel_spect)


    '''
    Audio framing
    '''
    def frame(self, y, win_step=64, win_size=128):

        # Number of frames
        nb_frames = 1 + int((y.shape[2] - win_size) / win_step)

        # Framming
        frames = np.zeros((y.shape[0], nb_frames, y.shape[1],
win_size)).astype(np.float16)
        for t in range(nb_frames):
            frames[:,t,:,:] = np.copy(y[:,:,(t * win_step):(t *
win_step + win_size)]).astype(np.float16)

        return frames


    '''
    Time distributed Convolutional Neural Network model
    '''
    def build_model(self):

        # Clear Keras session
        K.clear_session()

        # Define input
```

```
        input_y = Input(shape=(5, 128, 128, 1),
name='Input_MELSPECT')

        # First LFLB (local feature learning block)
        y = TimeDistributed(Conv2D(64, kernel_size=(3, 3),
strides=(1, 1), padding='same'), name='Conv_1_MELSPECT')(input_y)
        y = TimeDistributed(BatchNormalization(),
name='BatchNorm_1_MELSPECT')(y)
        y = TimeDistributed(Activation('elu'),
name='Activ_1_MELSPECT')(y)
        y = TimeDistributed(MaxPooling2D(pool_size=(2, 2),
strides=(2, 2), padding='same'), name='MaxPool_1_MELSPECT')(y)
        y = TimeDistributed(Dropout(0.2),
name='Drop_1_MELSPECT')(y)

        # Second LFLB (local feature learning block)
        y = TimeDistributed(Conv2D(64, kernel_size=(3, 3),
strides=(1, 1), padding='same'), name='Conv_2_MELSPECT')(y)
        y = TimeDistributed(BatchNormalization(),
name='BatchNorm_2_MELSPECT')(y)
        y = TimeDistributed(Activation('elu'),
name='Activ_2_MELSPECT')(y)
        y = TimeDistributed(MaxPooling2D(pool_size=(4, 4),
strides=(4, 4), padding='same'), name='MaxPool_2_MELSPECT')(y)
        y = TimeDistributed(Dropout(0.2),
name='Drop_2_MELSPECT')(y)

        # Third LFLB (local feature learning block)
        y = TimeDistributed(Conv2D(128, kernel_size=(3, 3),
strides=(1, 1), padding='same'), name='Conv_3_MELSPECT')(y)
        y = TimeDistributed(BatchNormalization(),
name='BatchNorm_3_MELSPECT')(y)
        y = TimeDistributed(Activation('elu'),
name='Activ_3_MELSPECT')(y)
        y = TimeDistributed(MaxPooling2D(pool_size=(4, 4),
strides=(4, 4), padding='same'), name='MaxPool_3_MELSPECT')(y)
```

```
        y = TimeDistributed(Dropout(0.2),
name='Drop_3_MELSPECT')(y)

        # Fourth LFLB (local feature learning block)
        y = TimeDistributed(Conv2D(128, kernel_size=(3, 3),
strides=(1, 1), padding='same'), name='Conv_4_MELSPECT')(y)
        y = TimeDistributed(BatchNormalization(),
name='BatchNorm_4_MELSPECT')(y)
        y = TimeDistributed(Activation('elu'),
name='Activ_4_MELSPECT')(y)
        y = TimeDistributed(MaxPooling2D(pool_size=(4, 4),
strides=(4, 4), padding='same'), name='MaxPool_4_MELSPECT')(y)
        y = TimeDistributed(Dropout(0.2),
name='Drop_4_MELSPECT')(y)

        # Flat
        y = TimeDistributed(Flatten(), name='Flat_MELSPECT')(y)

        # LSTM layer
        y = LSTM(256, return_sequences=False, dropout=0.2,
name='LSTM_1')(y)

        # Fully connected
        y = Dense(7, activation='softmax', name='FC')(y)

        # Build final model
        model = Model(inputs=input_y, outputs=y)

        return model


    '''
    Predict speech emotion over time from an audio file
    '''
    def predict_emotion_from_file(self, filename,
chunk_step=16000, chunk_size=49100, predict_proba=False,
sample_rate=16000):
```

```
        # Read audio file
        y, sr = librosa.core.load(filename, sr=sample_rate,
offset=0.5)

        # Split audio signals into chunks
        chunks = self.frame(y.reshape(1, 1, -1), chunk_step,
chunk_size)

        # Reshape chunks
        chunks = chunks.reshape(chunks.shape[1],chunks.shape[-1])

        # Z-normalization
        y = np.asarray(list(map(zscore, chunks)))

        # Compute mel spectrogram
        mel_spect = np.asarray(list(map(self.mel_spectrogram,
y)))

        # Time distributed Framing
        mel_spect_ts = self.frame(mel_spect)

        # Build X for time distributed CNN
        X = mel_spect_ts.reshape(mel_spect_ts.shape[0],
                                 mel_spect_ts.shape[1],
                                 mel_spect_ts.shape[2],
                                 mel_spect_ts.shape[3],
                                 1)

        # Predict emotion
        if predict_proba is True:
            predict = self._model.predict(X)
        else:
            predict = np.argmax(self._model.predict(X), axis=1)
            predict = [self._emotion.get(emotion) for emotion in
predict]
```

```
        # Clear Keras session
        K.clear_session()

        # Predict timestamp
        timestamp = np.concatenate([[chunk_size],
np.ones((len(predict) - 1)) * chunk_step]).cumsum()
        timestamp = np.round(timestamp / sample_rate)

        return [predict, timestamp]

    '''
    Export emotions predicted to csv format
    '''
    def prediction_to_csv(self, predictions, filename, mode='w'):

        # Write emotion in filename
        with open(filename, mode) as f:
            if mode == 'w':
                f.write("EMOTIONS"+'\n')
            for emotion in predictions:
                f.write(str(emotion)+'\n')
            f.close()
```

VIDEO RECOGNITION:

```
### General imports ###
from __future__ import division
import numpy as np
import pandas as pd
import time
from time import sleep
import re
import os
import requests
```

```
import argparse
from collections import OrderedDict

### Image processing ###
import cv2
from scipy.ndimage import zoom
from scipy.spatial import distance
import imutils
from scipy import ndimage
import dlib
from imutils import face_utils

### Model ###
from tensorflow.keras.models import load_model
from tensorflow.keras import backend as K

def gen():
    """
    Video streaming generator function.
    """

    # Start video capute. 0 = Webcam, 1 = Video file, -1 = Webcam
for Web
    video_capture = cv2.VideoCapture(0)

    # Image shape
    shape_x = 48
    shape_y = 48
    input_shape = (shape_x, shape_y, 1)

    # We have 7 emotions
    nClasses = 7

    # Timer until the end of the recording
    end = 0

    # Count number of eye blinks (not used in model prediction)
```

```
    def eye_aspect_ratio(eye):

        A = distance.euclidean(eye[1], eye[5])
        B = distance.euclidean(eye[2], eye[4])
        C = distance.euclidean(eye[0], eye[3])
        ear = (A + B) / (2.0 * C)

        return ear

    # Detect facial landmarks and return coordinates (not used in
model prediction but in visualization)
    def detect_face(frame):

        #Cascade classifier pre-trained model
        cascPath = 'Models/face_landmarks.dat'
        faceCascade = cv2.CascadeClassifier(cascPath)

        #BGR -> Gray conversion
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        #Cascade MultiScale classifier
        detected_faces =
faceCascade.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=6,

minSize=(shape_x, shape_y),

flags=cv2.CASCADE_SCALE_IMAGE)
        coord = []

        for x, y, w, h in detected_faces :
            if w > 100 :
                # Square around the landmarks
                sub_img=frame[y:y+h,x:x+w]
                # Put a rectangle around the face
                cv2.rectangle(frame,(x,y),(x+w,y+h),(0,
255,255),1)
                coord.append([x,y,w,h])
```

```
        return gray, detected_faces, coord

    #  Zoom on the face of the person
    def extract_face_features(faces, offset_coefficients=(0.075,
0.05)):

        # Each face identified
        gray = faces[0]

        # ID of each face identifies
        detected_face = faces[1]

        new_face = []

        for det in detected_face :
            # Region in which the face is detected
            # x, y represent the starting point, w the width
(moving right) and h the height (moving up)
            x, y, w, h = det

            #Offset coefficient (margins), np.floor takes the
lowest integer (delete border of the image)
            horizontal_offset =
np.int(np.floor(offset_coefficients[0] * w))
            vertical_offset =
np.int(np.floor(offset_coefficients[1] * h))

            # Coordinates of the extracted face
            extracted_face = gray[y+vertical_offset:y+h,
x+horizontal_offset:x-horizontal_offset+w]

            #Zoom on the extracted face
            new_extracted_face = zoom(extracted_face, (shape_x /
extracted_face.shape[0],shape_y / extracted_face.shape[1]))

            # Cast type to float
```

```
            new_extracted_face =
new_extracted_face.astype(np.float32)

            # Scale the new image
            new_extracted_face /= float(new_extracted_face.max())

            # Append the face to the list
            new_face.append(new_extracted_face)

        return new_face

    # Initiate Landmarks
    (lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
    (rStart, rEnd) =
face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

    (nStart, nEnd) = face_utils.FACIAL_LANDMARKS_IDXS["nose"]
    (mStart, mEnd) = face_utils.FACIAL_LANDMARKS_IDXS["mouth"]
    (jStart, jEnd) = face_utils.FACIAL_LANDMARKS_IDXS["jaw"]

    (eblStart, eblEnd) =
face_utils.FACIAL_LANDMARKS_IDXS["left_eyebrow"]
    (ebrStart, ebrEnd) =
face_utils.FACIAL_LANDMARKS_IDXS["right_eyebrow"]

    # Load the pre-trained X-Ception model
    model = load_model('Models/video.h5')

    # Load the face detector
    face_detect = dlib.get_frontal_face_detector()

    # Load the facial landmarks predictor
    predictor_landmarks  =
dlib.shape_predictor("Models/face_landmarks.dat")

    # Prediction vector
    predictions = []
```

```
    # Timer
    global k
    k = 0
    max_time = 15
    start = time.time()

    angry_0 = []
    disgust_1 = []
    fear_2 = []
    happy_3 = []
    sad_4 = []
    surprise_5 = []
    neutral_6 = []

    # Record for 45 seconds
    while end - start < max_time :

        k = k+1
        end = time.time()

        # Capture frame-by-frame the video_capture initiated
above
        ret, frame = video_capture.read()

        # Face index, face by face
        face_index = 0

        # Image to gray scale
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        # All faces detected
        rects = face_detect(gray, 1)

        #gray, detected_faces, coord = detect_face(frame)
```

```
        # For each detected face
        for (i, rect) in enumerate(rects):

            # Identify face coordinates
            (x, y, w, h) = face_utils.rect_to_bb(rect)
            face = gray[y:y+h,x:x+w]

            # Identify landmarks and cast to numpy
            shape = predictor_landmarks(gray, rect)
            shape = face_utils.shape_to_np(shape)

            # Zoom on extracted face
            face = zoom(face, (shape_x / face.shape[0],shape_y /
face.shape[1]))

            # Cast type float
            face = face.astype(np.float32)

            # Scale the face
            face /= float(face.max())
            face = np.reshape(face.flatten(), (1, 48, 48, 1))

            # Make Emotion prediction on the face, outputs
probabilities
            prediction = model.predict(face)

            # For plotting purposes with Altair
            angry_0.append(prediction[0][0].astype(float))
            disgust_1.append(prediction[0][1].astype(float))
            fear_2.append(prediction[0][2].astype(float))
            happy_3.append(prediction[0][3].astype(float))
            sad_4.append(prediction[0][4].astype(float))
            surprise_5.append(prediction[0][5].astype(float))
            neutral_6.append(prediction[0][6].astype(float))

            # Most likely emotion
            prediction_result = np.argmax(prediction)
```

```
            # Append the emotion to the final list
            predictions.append(str(prediction_result))

            # Draw rectangle around the face
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255,
0), 2)

            # Top left : Put the ID of the face
            cv2.putText(frame, "Face #{}".format(i + 1), (x - 10,
y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

            # Draw all the landmarks dots
            for (j, k) in shape:
                cv2.circle(frame, (j, k), 1, (0, 0, 255), -1)

            # Add prediction probabilities on the top-left report
            cv2.putText(frame, "----------------",(40,100 +
180*i), cv2.FONT_HERSHEY_SIMPLEX, 0.5, 155, 0)
            cv2.putText(frame, "Emotional report : Face #" +
str(i+1),(40,120 + 180*i), cv2.FONT_HERSHEY_SIMPLEX, 0.5, 155, 0)
            cv2.putText(frame, "Angry : " +
str(round(prediction[0][0],3)),(40,140 + 180*i),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, 155, 0)
            cv2.putText(frame, "Disgust : " +
str(round(prediction[0][1],3)),(40,160 + 180*i),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, 155, 0)
            cv2.putText(frame, "Fear : " +
str(round(prediction[0][2],3)),(40,180 + 180*i),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, 155, 1)
            cv2.putText(frame, "Happy : " +
str(round(prediction[0][3],3)),(40,200 + 180*i),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, 155, 1)
            cv2.putText(frame, "Sad : " +
str(round(prediction[0][4],3)),(40,220 + 180*i),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, 155, 1)
```

```
            cv2.putText(frame, "Surprise : " +
str(round(prediction[0][5],3)),(40,240 + 180*i),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, 155, 1)
            cv2.putText(frame, "Neutral : " +
str(round(prediction[0][6],3)),(40,260 + 180*i),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, 155, 1)

            # Annotate main image with the emotion label
            if prediction_result == 0 :
                cv2.putText(frame, "Angry",(x+w-10,y-10),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
            elif prediction_result == 1 :
                cv2.putText(frame, "Disgust",(x+w-10,y-10),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
            elif prediction_result == 2 :
                cv2.putText(frame, "Fear",(x+w-10,y-10),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
            elif prediction_result == 3 :
                cv2.putText(frame, "Happy",(x+w-10,y-10),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
            elif prediction_result == 4 :
                cv2.putText(frame, "Sad",(x+w-10,y-10),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
            elif prediction_result == 5 :
                cv2.putText(frame, "Surprise",(x+w-10,y-10),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
            else :
                cv2.putText(frame, "Neutral",(x+w-10,y-10),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

            # Eye Detection and Blink Count
            leftEye = shape[lStart:lEnd]
            rightEye = shape[rStart:rEnd]

            # Compute Eye Aspect Ratio
            leftEAR = eye_aspect_ratio(leftEye)
            rightEAR = eye_aspect_ratio(rightEye)
```

```
            ear = (leftEAR + rightEAR) / 2.0

            # And plot its contours
            leftEyeHull = cv2.convexHull(leftEye)
            rightEyeHull = cv2.convexHull(rightEye)
            cv2.drawContours(frame, [leftEyeHull], -1, (0, 255,
0), 1)
            cv2.drawContours(frame, [rightEyeHull], -1, (0, 255,
0), 1)

            # Detect Nose and draw its contours
            nose = shape[nStart:nEnd]
            noseHull = cv2.convexHull(nose)
            cv2.drawContours(frame, [noseHull], -1, (0, 255, 0),
1)

            # Detect Mouth and draw its contours
            mouth = shape[mStart:mEnd]
            mouthHull = cv2.convexHull(mouth)
            cv2.drawContours(frame, [mouthHull], -1, (0, 255, 0),
1)

            # Detect Jaw and draw its contours
            jaw = shape[jStart:jEnd]
            jawHull = cv2.convexHull(jaw)
            cv2.drawContours(frame, [jawHull], -1, (0, 255, 0),
1)

            # Detect Eyebrows and draw its contours
            ebr = shape[ebrStart:ebrEnd]
            ebrHull = cv2.convexHull(ebr)
            cv2.drawContours(frame, [ebrHull], -1, (0, 255, 0),
1)

            ebl = shape[eblStart:eblEnd]
            eblHull = cv2.convexHull(ebl)
```

```
            cv2.drawContours(frame, [eblHull], -1, (0, 255, 0),
1)

        # Show number of faces captured
        cv2.putText(frame,'Number of Faces : ' +
str(len(rects)),(40, 40), cv2.FONT_HERSHEY_SIMPLEX, 1, 155, 1)

        # For flask, save image as t.jpg (rewritten at each step)
        cv2.imwrite('tmp/t.jpg', frame)

        # Yield the image at each step
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' +
open('tmp/t.jpg', 'rb').read() + b'\r\n')

        # Emotion mapping
        #emotion = {0:'Angry', 1:'Disgust', 2:'Fear', 3:'Happy',
4:'Neutral', 5:'Sad', 6:'Surprise'}

        # Once reaching the end, write the results to the
personal file and to the overall file
        if end-start > max_time - 1 :
            with open("static/js/db/histo_perso.txt", "w") as d:
                d.write("density"+'\n')
                for val in predictions :
                    d.write(str(val)+'\n')

            with open("static/js/db/histo.txt", "a") as d:
                for val in predictions :
                    d.write(str(val)+'\n')


            rows =
zip(angry_0,disgust_1,fear_2,happy_3,sad_4,surprise_5,neutral_6)

            import csv
            with open("static/js/db/prob.csv", "w") as d:
```

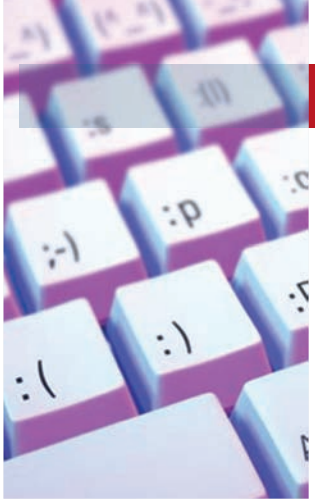
```
                writer = csv.writer(d)
                for row in rows:
                    writer.writerow(row)


            with open("static/js/db/prob_tot.csv", "a") as d:
                writer = csv.writer(d)
                for row in rows:
                    writer.writerow(row)

            K.clear_session()
            break

    video_capture.release()
# Clear session to allow user to do another test afterwards
#K.clear_session()


    # d.write(','.join(str(i) for i in angry_0)+'\n')
    # d.write(','.join(str(i) for i in disgust_1)+'\n')
    #d.write(','.join(str(i) for i in fear_2)+'\n')
    # d.write(','.join(str(i) for i in happy_3)+'\n')
    #  d.write(','.join(str(i) for i in sad_4)+'\n')
    #  d.write(','.join(str(i) for i in surprise_5)+'\n')
#  d.write(','.join(str(i) for i in neutral_6)+'\n')
```



Editor: **Erik Cambria,** Nanyang Technological University, Singapore, cambria@ntu.edu.sg

Multimodal Sentiment Intensity Analysis in Videos: Facial Gestures and Verbal Messages

Amir Zadeh, *Carnegie Mellon University*
Rowan Zellers, *University of Washington*
Eli Pincus, *University of Southern California*
Louis-Philippe Morency, *Carnegie Mellon University*

With the advent of mobile applications and social websites such as YouTube, Vine, and Vimeo, we have observed an increase in the number of online videos shared by people expressing their opinions, stories, and reviews. To give you a better idea how popular these websites are, more than 300 hours of video is uploaded to YouTube every minute. These videos address a large array of topics, such as movies, books, and products. This growth in multimedia sharing has seen increasing attention from many companies, researchers, and consumers interested in building better opinion-mining applications for summarization, question answering, and video retrieval. We highlight three challenges of studying sentiment in these online opinion videos.

The first challenge comes from the volatile and high-tempo nature of these opinion videos, wherein speakers often will switch between topics and opinions. This makes it challenging to identify and segment the different opinions expressed. For example, a speaker can express more than one opinion in the same spoken utterance, as in, "That was a great effect; there is a lot of cheap childish humor everyone can relate to, but I thought it was hilarious."

The second challenge comes with the range and subtlety of sentiment expressed in these opinion videos. We want approaches that can recognize the polarity of a video segment (for example, positive or negative) and also estimate the strength of the expressed sentiment.

The third challenge is a fundamental research question on how to use information more than text for sentiment analysis. In everyday communications, ideas and opinions are expressed through verbal content as well as visual and vocal behaviors, such as facial expressions, head gestures, and voice quality.

In this article, we introduce the Multimodal Opinion-Level Sentiment Intensity (MOSI) dataset, the video corpus with opinion-level sentiment intensity annotations that can be used for sentiment, subjectivity, and multimodal language studies. (For more information on text-based and multimodal sentiment analysis, see the "Background" sidebar.) We focus on psycholinguistic study of coverbal gestures.[1] Using a data-driven approach, we exploit prototypical interaction patterns between facial gestures and spoken words, and we introduce a new representation called *Multimodal Dictionary*. Finally, we evaluate our proposed Multimodal Dictionary on the challenging task of sentiment intensity prediction, using a speaker-independent paradigm (in which the model is tested on a new, unseen set of speakers to reduce the chance of bias introduced by speaker identification).

MOSI Dataset

In this section, we introduce our new MOSI dataset, the first such dataset to enable studies of multimodal sentiment intensity analysis. It can also be reliably used for detailed studies of language and gestures because of the rigorous annotation

1541-1672/16/$33.00 © 2016 IEEE
Published by the IEEE Computer Society

Background

Text-based sentiment analysis research has been an active and extremely successful field.[1] Among the notable efforts are works done in concept-level sentiment analysis,[2] automatic identification of opinion words and their sentiment polarity,[3] studies using *n*-grams and more complex language models,[4] works addressing sentiment compositionality by using polarity shifting rules or careful feature engineering,[5] and works that use deep learning approaches.[6] All these approaches primarily focus on the (spoken or written) text and ignore other communicative modalities.

Multimodal sentiment analysis has gained attention because of recent successes in multimodal analysis of human communications and affect.[7] Similar to our study are works that use support vector machines to classify sentiment polarity based on movie reviews,[8] that study multimodal sentiment analysis in Spanish videos,[9] that use convolutional neural networks and careful feature engineering for sentiment polarity classification,[10] and that use externally extracted word polarity data.[11] All the approaches in previous works use multimodal cues, including visual and acoustic cues. However, they have shortcomings with respect to core language and gestures studies, they present no analysis of sentiment intensity, and their approaches are speaker dependent.

References

1. B. Pang and L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts," *Proc. 42nd Ann. Conf. Assoc. Computational Linguistics*, 2004, article 271.
2. E. Cambria et al., "AffectiveSpace 2: Enabling Affective Intuition for Concept-Level Sentiment Analysis," *Proc. 29th AAAI Conf. Artificial Intelligence*, 2015, pp. 508–514.
3. M. Taboada et al., "Lexicon-Based Methods for Sentiment Analysis," *Computational Linguistics*, vol. 37, no. 2, 2011, pp. 267–307.
4. B. Yang, and C. Cardie, "Extracting Opinion Expressions with Semi-Markov Conditional Random Fields," *Proc. Jt. Conf. Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 1335–1345.
5. T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency Tree-Based Sentiment Classification Using CRFs with Hidden Variables," *Proc. Ann. Conf. North Am. Chapter of the Assoc. Computational Linguistics Human Language Technologies*, 2010, pp. 786–794.
6. R. Socher et al., "Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank," *Proc. Conf. Empirical Methods in Natural Language Processing*, 2013, pp. 1631–1642.
7. E. Cambria, "Affective Computing and Sentiment Analysis," *IEEE Intelligent Systems*, vol. 31, no. 2, 2016, pp. 102–107.
8. L.P. Morency, R. Mihalcea, and P. Doshi, "Towards Multimodal Sentiment Analysis: Harvesting Opinions from the Web," *Proc. 13th Int'l Conf. Multimodal Interfaces*, 2011, pp. 169–176.
9. V.P. Rosas, R. Mihalcea, and L.P. Morency, "Multimodal Sentiment Analysis of Spanish Online Videos," *IEEE Intelligent Systems*, vol. 28, no. 3, 2013, pp. 38–45.
10. S. Poria, E. Cambria, and A. Gelbukh, "Deep Convolutional Neural Network Textual Features and Multiple Kernel Learning for Utterance-Level Multimodal Sentiment Analysis," *Proc. Conf. Empirical Methods in Natural Language Processing*, 2015, pp. 2539–2544.
11. S. Poria et al., "Fusing Audio, Visual and Textual Clues for Sentiment Analysis from Multimodal Content," *Neurocomputing*, vol. 174, 2016, pp. 50–59.

procedure. The dataset annotations contain the following:

- multimodal observations, including transcribed speech and visual gestures (an extensive set of automatically extracted text, audio, and visual features are also available for download with the dataset);
- opinion-level subjectivity segmentation;
- sentiment intensity annotations using unbiased crowdsourcing; and
- alignment between words, visual, and acoustic features.

The following subsections describe the dataset in more details.

Acquisition Methodology

We collected videos from YouTube with a focus on video blogs (vlogs)—popular monologue videos used by many YouTube users to express opinions about different subjects. The videos are recorded in diverse setups; some users have high-tech microphones and cameras, whereas others use less-professional recording devices. Users are in different distances from the camera with different lighting and background. The videos vary in length from 2 to 5 minutes. We selected a total of 93 videos from 89 distinct speakers, including 41 female and 48 male speakers. Most of the speakers were approximately between the ages of 20 and 30 years old. Although the speakers were from different ethnic backgrounds (for example, Caucasian, African American, Hispanic, and Asian), all speakers expressed themselves in English, and the videos originated from either the US or the UK. Figure 1 shows sample snapshots of video in the MOSI dataset.

We manually transcribed all the video clips to extract spoken words and the start time of each spoken utterance. Our transcription methodology had three stages. First, an expert transcriber manually transcribed all the videos, followed by a second transcriber reviewing and correcting all the transcriptions. Our transcription scheme contained details about pause fillers (such as "umm" and "uhh"), stresses, and speech pauses. In the third stage, the text was carefully aligned at word and phoneme levels with the videos using a forced alignment method called P2FA.[2] During the final stage, the results of the alignment were manually checked and, if necessary, corrected using PRAAT.[3]

Subjectivity Annotation

An important requirement of creating a dataset for sentiment analysis is to perform subjectivity segmentation to find opinionated segments of speech.

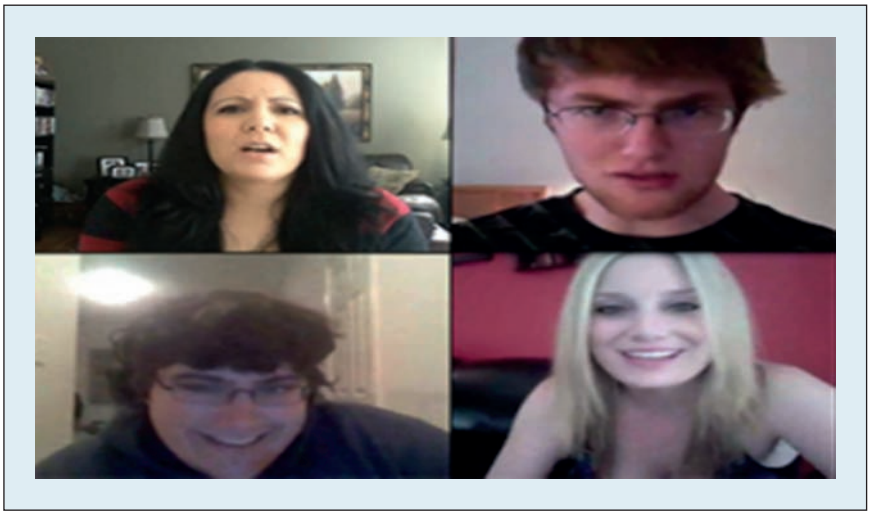


Figure 1. Example snapshots of videos from our new Multimodal Opinion-Level Sentiment Intensity (MOSI) dataset.

Table 1. MOSI dataset statistics.

Statistical measure	Value
Total no. segments	3,702
Total no. opinion segments	2,199
Total no. objective segments	1,503
Total no. videos	93
Total no. distinct speakers	89
Average no. opinion segments in video	23.2
Average length of opinion segments	4.2 seconds
Average word count per opinion segments	12
Total no. words in opinion segments	26,295
Total no. unique words in opinion segments	3,107
Total no. words in opinion segments appearing at least 10 times in the dataset	557

Following the work of Janyce Wiebe and colleagues,[4] *subjective sentences* are defined as expressions of a person's opinions, whereas *objective sentences* express facts and truth. Our annotation scheme expands their work to extract spoken opinion segments, defined to isolate distinct opinions and perform sentiment analysis on them. Therefore, subjective content comprises one or more opinion segments (hereafter, we will use *subjective segment*, *opinion segment*, and *opinion* interchangeably to refer to the same concept).

We define subjectivity as an attempt to express a private state, one that is distinguishable by carrying an opinion, belief, thought, feeling, emotion, goal, evaluation, or judgment. To more accurately annotate the boundaries of each opinion segment, we have defined the following rules. If the text contains an expression of a private state, the following segmentation rules apply (brackets are used to hold segments):

- Segment the subjective content on the basis of the number of private states revealed—for example, "[I love *The Shawshank Redemption*] [and I love *Transformers*]" results in two subjective segments.
- Segment if the utterance contains a modification of a private state while maintaining the subject—for example, "[Well, based on what I saw today, I feel like the movie industry is going crazy][or maybe it's just me being so hard on the poor actors.]"
- Segment if the subjective utterance ends with the start of an objective segment—for example, "[In my opinion, the movie was all about eating healthy food], you could see banners of different organic brands in several shots."

If there is subjective content and it extends beyond the boundary of the utterance while retaining the opinion, we merge the extension with the original utterance—for example, "[I don't like it! It's not a likable movie!]" The extension can be multiple sentences or part of a sentence.

Two trained annotators did the subjectivity annotation. The two annotations resulted in a Krippendorf's alpha of 0.68. The subjectivity annotation resulted in 2,199 subjective segments and 1,503 objective ones. We considered both subjective and objective segments for multimodal subjectivity studies, but for sentiment annotations, we focused on subjective segments. Table 1 gives detailed statistics of the dataset and opinion segments.

Crowdsourced Sentiment Intensity Annotation

Sentiment intensity is defined from strongly negative to strongly positive with a linear scale from –3 to +3. Online workers from Amazon Mechanical Turk performed the intensity annotations. Only master workers with an approval rate of higher than 95 percent were selected to participate. A total of 2,199 short video clips were created from the subjective opinion segments. For each video, the annotators had eight choices: strongly

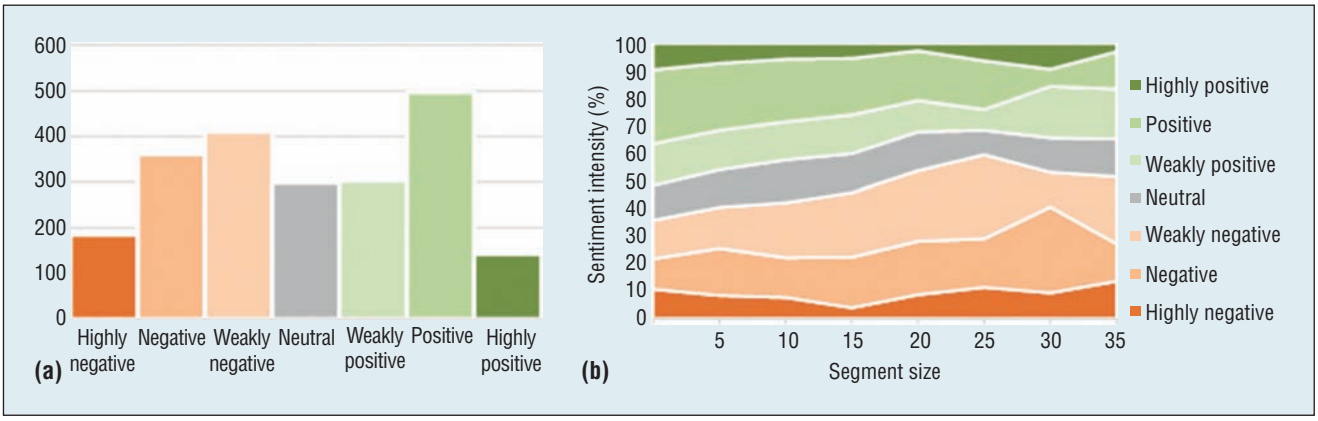


Figure 2. Histograms of sentiment distribution in MOSI dataset. (a) Distribution of sentiment over the entire dataset. (b) The percentage of each sentiment intensity per segment size (number of words in opinion segment).

positive (labeled as +3), positive (+2), weakly positive (+1), neutral (0), weakly negative (–1), negative (–2), strongly negative (–3), and uncertain.

We kept the instructions simple to reduce any training bias. The only tutorial was on how to use the online system (for example, how to submit the form). The task was phrased as follows: "How would you rate the sentiment expressed in this video segment? (Please note that you may or may not agree with what the speaker says. It is imperative that you only rate the sentiment stated by the speaker, not your opinion.)" Each video clip was annotated by five workers. The interannotator agreement between workers was 0.77 in terms of Krippendorf's alpha. The final sentiment intensity of each segment is the average of all five workers. Figure 2a shows the distribution of sentiment intensities for all opinion segments in the MOSI dataset; Figure 2b shows how the sentiment distribution changes as the size of the opinion (that is, the number of words in that opinion) increases. Although Richard Socher and colleagues reported that short (fewer than 10 words) text-only opinions have no significant sentiment and are mostly neutral,[5] short video segments have equal distribution along all scores, which shows that the presence of multimodal information, more than just text, makes it possible for human annotators to deduce sentiment for small opinion segments.

Manual Gesture Annotations

We provided a set of manually annotated gestures to study the relations between words and gestures. Because hands were not always visible in the YouTube videos, only facial gestures are annotated. We selected four gestures and expressions: smile, frown, head nod, and head shake. These are expressive of emotions and regularly happen in MOSI dataset. The annotations were done at the segment level. An expert coder manually annotated all 2,199 video segments, and a second coder annotated a subset of this dataset to compute the agreement between the coders. For all four gestures, the average coder agreement was 0.81.

Multimodal Analysis of Visual Gestures and Verbal Messages

The MOSI dataset enables detailed statistical study of language as a multimodal signal. We conducted a study to find a suitable multimodal representation for sentiment analysis. We wanted to understand the interaction patterns between spoken words and visual gestures. To study these interaction patterns, we studied the changes in the distribution of perceived sentiment intensity when a specific facial gesture is present or not. We performed this analysis at the opinion level, wherein we studied the multimodal interactions of the top 100 spoken words with all four facial gestures (smile, frown, head nod, and head shake).

Interaction Patterns

Figure 3 shows representative examples from our multimodal analysis, in which we identified four types of interaction patterns between spoken words and facial gestures: neutral, emphasizer, positive, and negative. Each subgraph is a histogram that represents the distribution of perceived sentiment intensities per opinion segment.

To help understand the average interaction of facial gestures with spoken words, the first row of Figure 3 shows how the sentiment intensities are distributed for all opinion segments (the top-left histogram of Figure 3 is repeated from Figure 2). It is not surprising to see that opinion segments with a smile or a head nod are perceived as more positive. The opposite effect is observed for frown and head shake gestures.

Neutral interaction pattern. To exemplify this pattern, we selected the most frequent word in our dataset: "the." "The" is considered sentimentally neutral in isolation. The second row of Figure 3 shows the interaction between the facial gestures and the spoken word

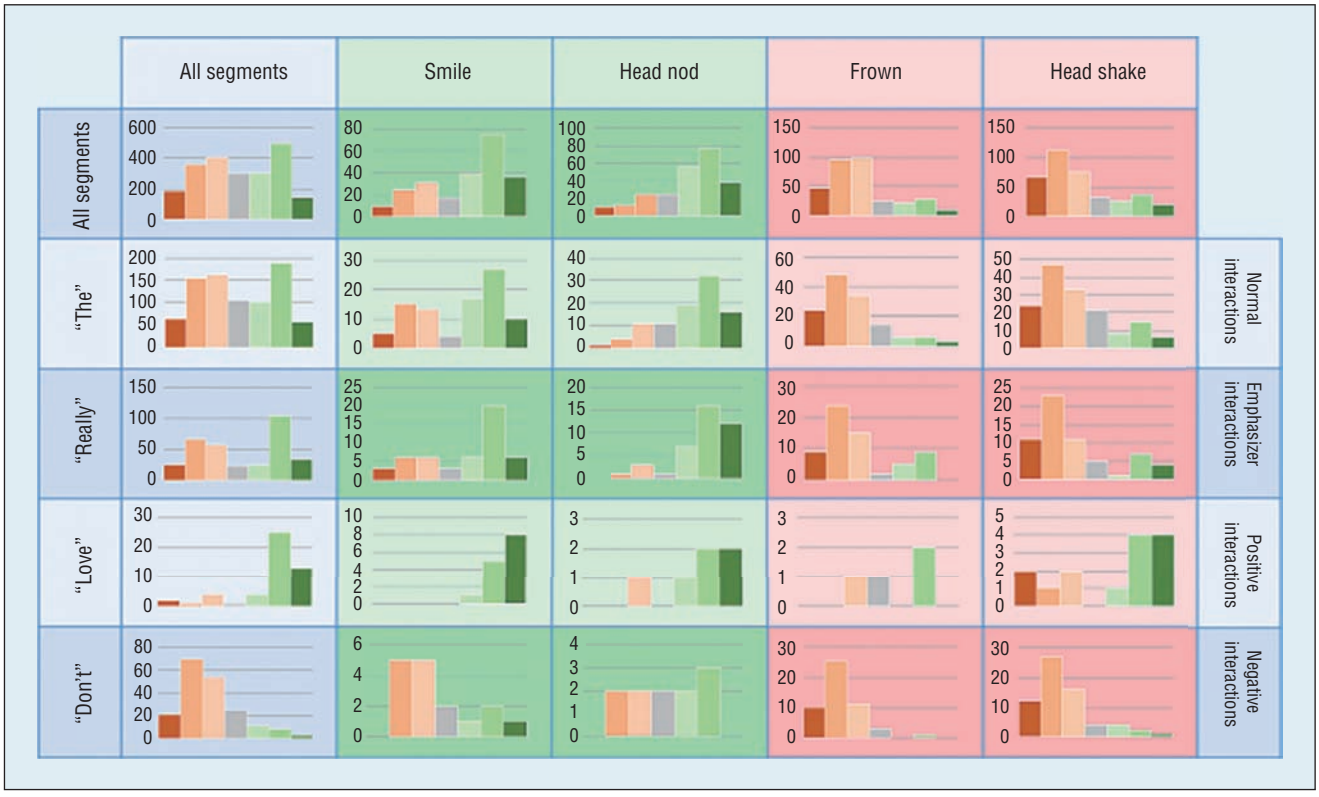


Figure 3. Sentiment intensity histograms for spoken words and visual gestures. In each histogram, the *y*-axis is the number of co-occurrence and the *x*-axis is the sentiment intensity as in Figure 2.

"the." We can observe that the pattern mostly follows the common interaction patterns in the first row.

Emphasizer interaction pattern. We observed a second interaction pattern in our multimodal analysis. To exemplify this pattern, we showed in the third row of Figure 3 how facial gestures interact with the word "really." When accompanied by a smile or head nod, the distribution tends to shift to positive sentiment intensity, with less negative or neutral intensities. The opposite effect happens when "really" is accompanied by a frown or head shake—then, the distribution is biased toward negative sentiment. In other words, this interaction pattern tends to shift the sentiment toward the extremes. We define this type of interaction pattern as an *emphasizer*.

Negative and positive interaction patterns. The third and fourth type of interaction seem to appear when studying sentimentally polarized words, because their sentiment distributions are not affected in the same way as the neutral and emphasizer. For example, the positively polarized word "love" is shown in the fourth row of Figure 3 (we merged the verbs "love" and "loved" in these histograms for simplification). The sentiment distributions do not significantly change polarity when accompanied by a frown or head shake. An opposite trend happens when we study a negatively polarized word such as "don't," shown in the last row of Figure 3 (we merged all instances of "don't," "doesn't," and "didn't" in these histograms). We observe limited changes in the sentiment distributions for the smile and head nod.

Multimodal Dictionary

On the basis of these interaction patterns between words and gestures, we present a simple representation model that jointly accounts for words and gestures in each opinion segment. We define $W = \{w_1, w_2, w_3, \ldots, w_K\}$ as the set of words in our dataset and K as the dictionary size. We observed in our experiments that information about gestures being present or not present is useful; thus, we defined $G = \{$*smile*, *frown*, *head nod*, *head shake*, *~smile*, *~frown*, *~head nod*, *~head shake*$\}$ (the approximation symbol indicates no evidence of that gesture). We then defined the Multimodal Dictionary to be the Cartesian product of sets of words W and gestures G as follows:

$$M = \{(w, g) \mid w \in W, g \in G\}.$$

The Multimodal Dictionary creates a simple joint space of words and gestures. Each element in this multimodal representation is a binary variable similar to the bag-of-words representation for text and captures if a word

and gesture have co-occurred. Using this method yields better results in sentiment intensity analysis compared with common fusion methods.

Experimental Results

All the experiments described in this section were done in a speaker-independent framework. We trained prediction models using nu-SVR[6] and tested them using a fivefold cross-validation methodology. The automatic validation of the hyperparameters was performed with fourfold cross-validation on the training sets. We calculated the regressors' performance based on mean absolute error (MAE) and correlation. We trained the following models:

- *Random*. We included in our experiments a simple baseline model that always predicts a random sentiment intensity between [3, –3]. This baseline gives an overall idea about how random models will work.
- *Verbal*. We trained this model using only verbal features from MOSI. We created a simple bag-of-words feature set from monograms and bigrams created from words in speech segments, including speech pauses and pause fillers. All the features with fewer than 10 instances in the dataset were removed from the bag-of-words set, given their infrequency.
- *Visual*. We trained this model using facial gestures, as described earlier. We assigned a binary feature for each of the four facial gestures: smile, frown, head nod, and head shake.
- *Verbal+Visual*. We trained this model on verbal and visual data combined. The verbal and visual features were simply concatenated for each opinion segment.
- *Multimodal Dictionary*. We trained this model on Multimodal Dictionary representation. Each element in the Multimodal Dictionary is treated as a random variable and denotes joint representation between words and gestures.
- *Human Performance*. Humans are asked to predict the sentiment score of each opinion segment. This will be both a baseline for how well humans can predict sentiment intensity and a future target for machine learning methods.

Table 2. Mean absolute error and correlation for each of the trained baseline models.

Model	Mean absolute error	Correlation
Random	1.88	0.00
Verbal	1.18	0.46
Visual	1.24	0.36
Verbal+Visual	1.14	0.49
Multimodal Dictionary	**1.10**	**0.53**
Human Performance	0.61	0.83

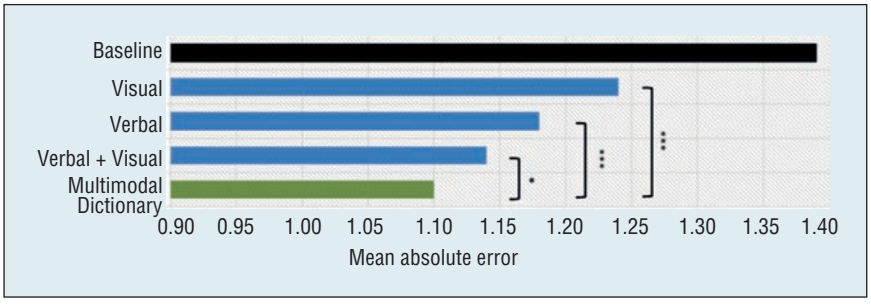


Figure 4. Statistical comparison between Multimodal Dictionary and other trained models. One star shows $p < 0.01$, and three stars show $p < 0.0001$.

Table 2 summarizes our experimental results. We performed a significance test using pairwise *T*-test between the models (see Figure 4; stars indicate the *p*-value range). Our first observation from these results is that the Verbal+Visual model outperforms both the Verbal model and Visual model individually.

A second observation is that the Multimodal Dictionary model outperforms the Verbal+Visual model. The difference is statistically significant ($p < 0.01$). This is well-aligned with multimodal study mentioned earlier and presented in Figure 3, wherein spoken words and facial gestures were shown to have multiple interaction patterns. Our Multimodal Dictionary is designed to explicitly model these interactions. This new representation results in better performance for sentiment intensity prediction.

In this article, we introduced the Multimodal Dictionary to better understand the interaction between facial gestures and spoken words when expressing sentiment. This new computational representation improved prediction performance in speaker-independent multimodal sentiment intensity analysis. The findings we present here open the door to new research directions for studying human communication dynamics. One promising future direction is to analyze the vocal behaviors (such as vocal emphasis or prosodic cues) in the context of multimodal sentiment expressions. This analysis should also be augmented to take into account the temporal contingency between these vocal, visual, and verbal behaviors. These research directions will help us better under-

stand the dynamics of human communications central to many applications such as healthcare and education. ■

Acknowledgments

This material is based on work supported in part by the National Science Foundation under grant no. IIS-1118018 and Yahoo Research. The content does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

References

1. D. McNeill, *Language and Gesture*, vol. 2, Cambridge Univ. Press, 2000.
2. J. Yuan, and M. Liberman, "Speaker Identification on the SCOTUS Corpus," *J. Acoustical Soc. Am.*, vol. 123, no. 5, 2008, pp. 3878.
3. P. Boersma, "Praat, A System for Doing Phonetics by Computer," *GLOT Int'l*, vol. 5, no. 9/10, 2002, pp. 341–345.
4. J. Wiebe, T. Wilson, and C. Cardie, "Annotating Expressions of Opinions and Emotions in Language," *Language Resources and Evaluation*, vol. 39, nos. 2–3, 2013, pp. 165–210.
5. R. Socher et al., "Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank," *Proc. Conf. Empirical Methods in Natural Language Processing*, 2013, pp. 1631–1642.
6. A.J. Smola, and B. Schölkopf, "A Tutorial on Support Vector Regression," *Statistics and Computing*, vol. 14, no. 3, 2004, pp. 199–222.

Amir Zadeh is a PhD student in deep learning at Carnegie Mellon University and a Yahoo InMind Fellow. Contact him at abagherz@cs.cmu.edu.

Rowan Zellers is a PhD student in computer science at the University of Washington. Contact him at rowanz@uw.edu.

Eli Pincus is a PhD student in spoken dialogue at the University of Southern California and a research assistant in the Natural Dialogue Group at USC's Institute for Creative Technologies. Contact him at pincus@ict.usc.edu.

Louis-Philippe Morency is an assistant professor in the Language Technology Institute at Carnegie Mellon University, where he leads the Multimodal Communication and Machine Learning Laboratory. Contact him at morency@cs.cmu.edu.