

SOLENT UNIVERSITY, SOUTHAMPTON  
FACULTY OF BUSINESS, LAW AND DIGITAL TECHNOLOGY

Masters in Applied AI and Data Science

2022

*“Multi-modal Emotional Recognition Model”*

Student Name: Sony Abraham  
Student ID : Q15716376

Supervisor: Dr Shadi Eltanani  
Date of submission: September 2022

## Acknowledgements

Thanks to almighty Lord for showering His blessings for the successful completion of my thesis.

It is a great pleasure to acknowledge my deepest thanks and gratitude to Solent University, Southampton for the opportunity to be part of 2021-2022 Masters in Applied AI and Data Science Batch. I am humbled to have this opportunity and had a wonderful time learning concepts of AI and to conduct this research work.

My sincere thanks to project supervisor, Dr Shadi Eltanani for his constant support and encouragement throughout the research work and for his honest feedback and kind supervision.

I would like to express my deepest thanks and sincere appreciation to Prof. Femi Isiaq, Module Leader for his guidance and encouragement throughout this Masters programme.

My heartfelt thanks to all my batchmates for their love, support, and encouragement especially Ms Shany Stephan for supporting me during testing of the current study.

Above all, I would like to express my thanks and gratitude to my husband and my lovely son for their constant motivation, understanding and immense love during the course of study.

Sony Abraham

## Abstract

We are all exposed to stress in our day to day lives due to our busy schedule, in attempt to achieving work life balance, performing daily household chores, managing kids, at workplace etc. Sometimes, we tend to neglect the development of negative emotions in us which might cause damage to our physical as well as mental wellbeing. The facial expressions are nonverbal way of communication, unless and until, it is forcefully suppressed, in usual cases, human faces exhibit their emotion through expressions and with the use of computer vision enabled application, it would help every individual to identify stress by analysing one's facial expression. In addition to the facial emotion recognition, if speech emotion and text emotion can also be recognized, it would be more accurate and beneficial to individual.

The motivation behind choosing this research study is the increasing stressful events in a common man's life which could be identified and managed to an extent by developing an application with emotion recognition capabilities. Current research study helped to develop a CNN model with accuracy of is 98.4% and 73.67% for Speech Emotion Recognition system and 71.78% for Facial Emotion Recognition System.

Keywords – CNN Model, Facial Emotion Recognition, Speech Emotion Recognition, Text to Emotion, Real-time

## Table of Contents

Acknowledgements.....	ii
Abstract.....	iii
1. Introduction.....	6
1.1. The Research Topic - Introduction .....	7
1.2. Significance of the Research Area .....	8
1.3. Problem Statement .....	10
1.4. Research Question .....	10
1.5. Research Aim and Objectives .....	11
1.5.1. Aim .....	11
1.5.2. Objectives .....	11
1.6. The Social Impact of Emotion Recognition.....	12
1.7. Research Approach .....	12
1.8. Project Outline .....	13
1.9. Project Plan and Implementation .....	13
2. LITERATURE REVIEW .....	15
2.1 Introduction.....	15
2.2. Research Material .....	16
.....	20
.....	20
.....	20
.....	20
.....	20
.....	20
2.3. Conclusion .....	26
3. Conceptual Framework.....	27
3.1. Facial Emotion Recognition .....	28
3.1.1 Viola-jones Object Detection Algorithm.....	28
3.1.1.a. Haar-like feature .....	29
3.1.2. Convolutional Neural Network (CNN) Model Architecture .....	32
2.1.2.a. CNN Workflow.....	32
4. Methodology.....	34

4.1. Introduction.....	34
4.2. Facial Emotion Recognition Model.....	34
4.2.1. Importing libraries and packages .....	35
4.2.2. Dataset To mount the contents of Google Drive where the dataset is stored 37	
4.2.3. Data Pre-processing .....	38
4.2.4. Creating model structure .....	39
A Sequential model is created with a plain stack of layers where each layer has exactly one input tensor and one output tensor.....	39
4.2.5 Training the CNN model .....	40
4.2.6. Saving the model structure & saving trained model.....	41
.....	41
.....	42
4.2.8 Model Evaluation.....	43
4.3. Speech Emotion Recognition Model The development of Speech Emotion Recognition model includes the below steps.....	45
1. Import libraries and packages .....	45
2. Data Preparation .....	45
3. Data Augmentation.....	45
4. Audio Feature Extraction.....	45
4.3.1. Import libraries and packages.....	47
4.3.2. Data Preparation .....	48
4.3.3. Data Augmentation.....	50
4.3.4. Feature Extraction.....	51
4.4. Text to Emotion .....	57
4.5. Audio-Video/Audio Input and File Conversion .....	58
4.5.1 Audio Input from Microphone .....	58
5. Results .....	62
6. Discussion.....	63
6.1 Facial Emotion Recognition Model.....	63
7. Conclusion .....	69
8. Limitations & Future Works.....	70
8. Reference list / Bibliography .....	71
9. Appendices .....	A
9.1 Appendix A: Ethics Application.....	A



## List of Tables

Table 1 Dissertation Study Outline.....	13
Table 2 Dataset and Classifier model information for the audio emotion recognition model (source - Ooi et al., 2021) .....	21

## List of Figures

<i>Figure 1</i> The 3 Key Elements of Emotion (source - Verywell / Emily Roberts <a href="https://www.verywellmind.com/what-are-emotions-2795178#citation-2">https://www.verywellmind.com/what-are-emotions-2795178#citation-2</a> )	3
Figure 2 Emotion Detection and Recognition Market Dynamics (source - <a href="https://www.marketsandmarkets.com/Market-Reports/emotion-detection-recognition-market-23376176.html">https://www.marketsandmarkets.com/Market-Reports/emotion-detection-recognition-market-23376176.html</a> )	7
Figure 3 Gantt Chart	13
Figure 4 Single Action Units in the Facial Action Code (source - (Ekman and Friesen, 1976,p.65).	17
Figure 5 Example of information given in the FAC for each Action Unit (source -Ekman and Friesen, 1976, p.66)	18
Figure 6 More grossly defined AUs in the Facial Action Code (source - (Ekman and Friesen, 1976,p.69).	18
Figure 6 A) Principal component analysis for positive and negative emotional vocalizations.	
Figure 7 Venn diagram showing classes of acoustic information that are used to predict participants' ratings for each of the emotional scales. (Source- K Scott et al.)	19
Figure 8 Block Diagram of the proposed audio-visual emotion recognition system (source - Hossain and Muhammad, 2019)	21
Figure 9 Overall Data Processing of the proposed audio-visual emotion recognition system (source - Hossain and Muhammad, 2019)	
Figure 38 Venn Diagram which classes of acoustic information are used to predict participants' ratings for each of the emotional scales.	
Figure 39 which classes of acoustic information are used to predict participants' ratings for each of the emotional scales.	21



## 1. Introduction

The demand for automated solutions and predictive analytics is growing at a higher pace due to the incredible inventions in the fields of artificial intelligence and computer vision technology. Among the diverse technological advancements in the field of AI (Artificial Intelligence), with the use of machine learning capabilities and deep learning techniques, emotion recognition applications have gained importance due to its varied application in different areas such as Education, Healthcare, Human Resource Management, Security & Policing and Marketing Research.

This study is an attempt to gain insights into the machine learning concepts and deep learning architecture to develop a multi-modal emotion recognition system comprising of a facial emotion recognition model, a speech emotion recognition model, and a text-to-emotion conversion technique which is combined to predict with maximum accuracy, the emotion identified from the face, speech signals and spoken words.

This research study was motivated by the idea of an existing relationship between stress and emotion, with the goal of managing the emotion by identifying and guiding the emotion at the right time to reduce the risks of stress and anxiety.

## 1.1. The Research Topic - Introduction

Emotions are reactions that humans have in response to events or situations. When a person experiences an emotion, the circumstances that trigger it determine what kind of emotion is experienced. Human daily lives are significantly impacted by their emotions. Decisions are made depending on emotions like happiness, anger, sadness, boredom, or frustration. According to Hockenbury and Sandra E. Hockenbury, an emotion is a psychological state that consists of three key elements: a subjective experience, a physiological response, and a behavioural or expressive response.

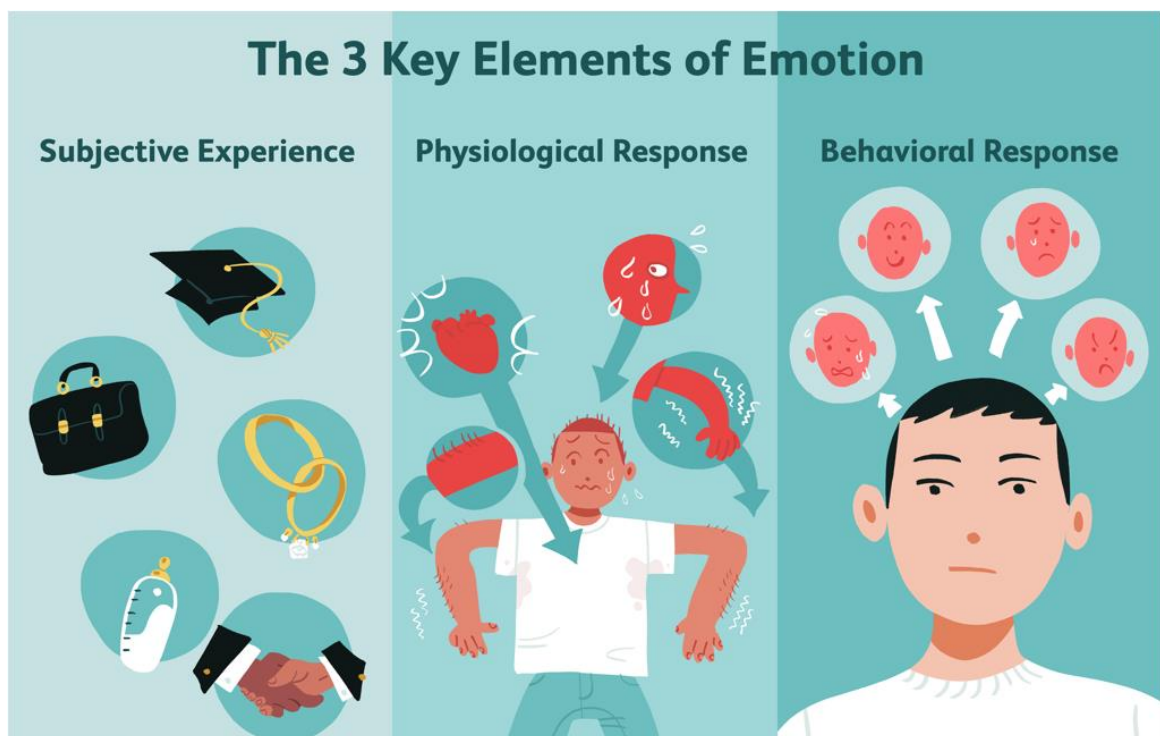


Figure 1 The 3 Key Elements of Emotion (source - Verywell / Emily Roberts <https://www.verywellmind.com/what-are-emotions-2795178#citation-2>)

Although emotions have names, the experience of those emotions varies from person to person and is highly subjective in nature, resulting in different physiological responses such as sweaty palms, racing heartbeats, and so on, which are regulated by the sympathetic nervous system of the human brain and are responsible for controlling the body's fight or flight reactions. The behavioural response or expression of emotion is the most important aspect of human emotion. Humans are the only creatures capable of

identifying and comprehending emotional expression, which is referred to as emotional intelligence.

The study of human emotion began from 4th BC when Aristotle attempted to extract the number of human emotions, and this has been continued by many researchers and psychologists to identify and categorize human emotions. In 1872, Charles Darwin defined a shorter list of basic emotions like fear, anger, sadness, happiness, and love. With the introduction of psychotherapy, the number of emotions increased considerably. Psychologists defined 90 different emotions to describe and differentiate human emotions. In 1972, psychologist Paul Ekman suggested in his book "Book of Emotions" that there are six basic emotions that are universal throughout human cultures: fear, disgust, anger, surprise, happiness, and sadness. In recent times, there have been a collective effort by researchers to categorize emotions that are considered universal. One of the most prominent theories in the field of psychology is Professor Robert Plutchick's wheel of emotions where he proposed eight basic emotions- joy, sadness, trust, disgust, fear, anger, surprise, and anticipation.

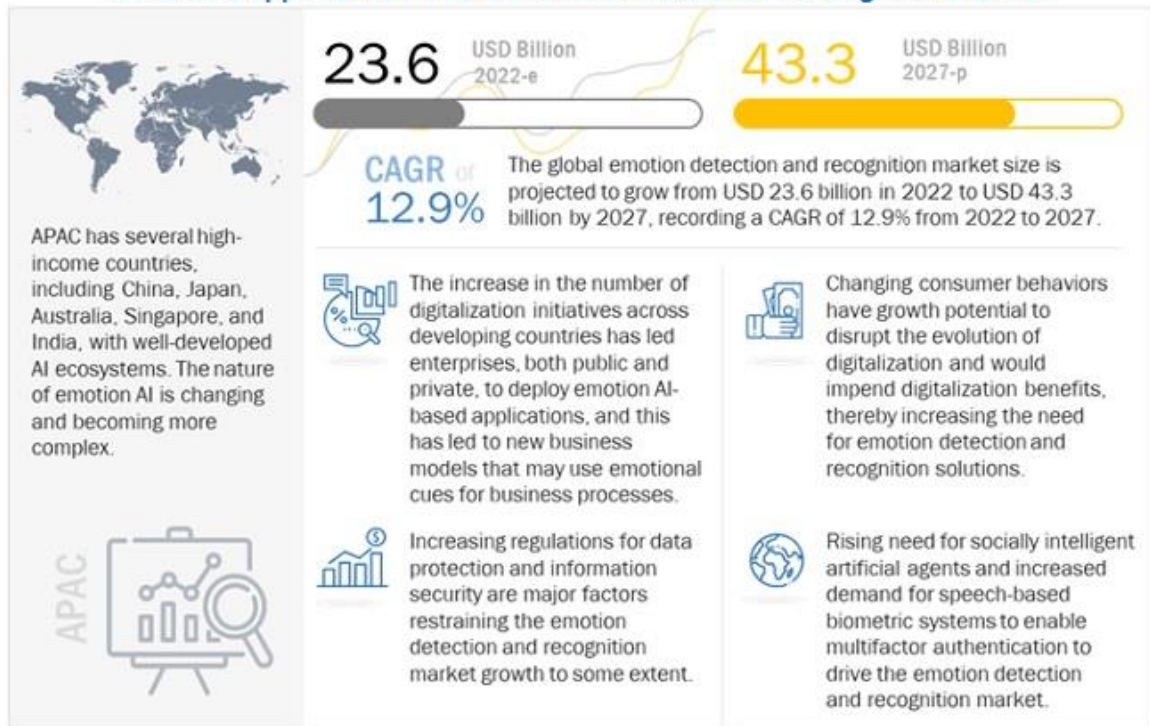
## 1.2. Significance of the Research Area

Many studies have found a link between stress and negative emotions. Stress is the body's reaction to feeling threatened; it can affect our mood, make us anxious and angry, and it can also affect our self-esteem. According to a group of medical experts and researchers in the United Kingdom, more than 37% of British residents experience stress for at least one full day per week. According to their survey of a 2000 study group from the UK, it was discovered that more than 85% of adults in the UK are stressed. Money is the most common source of stress, followed by work, health concerns, a lack of quality sleep, and household chores. It has also been discovered that women are more stressed than men. The stress levels and causes vary based on age, where the primary concern of experiencing stress among 25-34 years are financial problems, whereas 35-44 are both money and work, people aged 45-54 also feel work is stressful. But people above 55 years of age are mainly concerned about their health and related issues. COVID crisis, stress among healthcare workers and children, family problems are other significant stress types

which are prominent since 2020. It is also interesting to know that different age groups deal with stress differently. Various studies have been conducted to identify the impact of stress on the body and it has been revealed that stress can have many effects on the body and emotional effect is often ignored by individuals which might cause serious problems such as feeling angry, anxious, depressed, isolated, and moody resulting in poorer health outcomes and increased complications, decreased quality of life, and a greater need for health care services in the future.

Negative emotions are a result of stress but understanding the mechanism of stress coupled with negative emotions may help those suffering from stress reduce their negative emotions. According to published reports from the World Health Organization (WHO), major depressive disorder is anticipated to be a major reason for disability in the world by 2030. On the other hand, the global emotion detection and recognition market is expected to grow at a compound annual growth rate (CAGR) of 12.9% from US Dollar (USD) 23.6 billion in 2022 to USD 43.3 billion by 2027. The rising need for emotion detection systems to analyse emotional states, global adoption of Internet of Things (IoT), Artificial Intelligence (AI), Machine Learning (ML), and deep learning technologies, rising demand in the Automotive AI industry, rising need for high operational excellence, and rising need for socially intelligent artificial agents are the major factors driving the market growth.

## Attractive Opportunities in the Emotion Detection and Recognition Market



Source: Secondary Research, Expert Interviews, and MarketsandMarkets Analysis

Figure 2 Emotion Detection and Recognition Market Dynamics (source - <https://www.marketsandmarkets.com/Market-Reports/emotion-detection-recognition-market-23376176.html>)

### 1.3. Problem Statement

It is important to recognise stress at the right time to seek support and manage stress in the right way to avoid affecting the daily lives of individuals. Stress can harm both physical and mental wellbeing and make people feel anxious and angry, which can result in the development of exhaustion and strained relationships. In this research study, the emphasis is to help determine whether a person is under stress by identifying emotion from facial expression, speech audio signals and text analysis to develop strategies to manage stress effectively and effortlessly.

### 1.4. Research Question

The current research study for the dissertation included an extensive literature review to

identify similar significant studies conducted in emotion detection and recognition, attempted to understand the key technologies used to develop the system by carefully understanding the underlying concepts, identifying the limitations of study, and choosing relevant methods that would contribute to the successful implementation of this system.

**“Can a multi-modal emotion prediction model accurately identify and precisely predict the emotion of an individual from facial expressions, audio speech signals and spoken words to support manage stress effectively?”**

## 1.5. Research Aim and Objectives

### 1.5.1. Aim

The aim of this study is to develop an emotion recognition system from facial expression and speech audio signals using deep learning algorithms for computer vision technology and built-in text-to-emotion package in Python programming language based on multimodal input data to detect emotion.

### 1.5.2. Objectives

1. To extract facial features from images dataset, to label them based on Ekman’s Facial Action Coding system according to the corresponding facial features identified. Also, to extract audio features from audio dataset and to label them corresponding to the identified emotion classes.
2. To perform statistical analysis for standardization (to ensure the extracted facial features and audio features are resized to a predetermined standard where, dimensionality is reduced while retaining important features)
3. To evaluate the performance accuracy of the classifier system by using test data or validation data.

## 1.6. The Social Impact of Emotion Recognition

Emotion Recognition systems has already captured attention and is improvised to be implemented in Education field where emotion recognition can be used as innovative tool for improving students' performance and learning approaches. (Bouhlal et al.). In Healthcare, there is a much greater than just leveraging technology for security. Real-time emotion detection can be used to recognise a variety of emotions patients display during the time of their stay at facility and analyse the data to ascertain how they are feeling. The analysis' findings can point up areas where patients require additional care if they're in pain or depressed. (Sightcorp)

Although, emotion recognition is widely popular due to the potential for implementation in various industries, it is still growing and is yet to achieve higher accuracy, precision, and reliability. There is always a risk and drawback of security and privacy constraints along with socio-cultural issues associated with it since it uses facial expressions and/or audio signals of a person or a group of individuals. On a positive note, this can be achieved by incorporating and regulating policies and by implementing security controls.

## 1.7. Research Approach

A variety of nonverbal information can be exhibited on our face when we are stressed, which helps to interpret the underlying emotion. For example, raised and arched eyebrows shows surprise, lowered eyebrows often mean anger, sadness, or fear. Dilated eyes show interest whereas intense staring shows anger. Other signals include biting one's lip which could be sign of anxiety, open mouth showing fear, these facial features can be extracted by a computer vision algorithm and can be used to learn, train a machine learning model, and predict emotion of individuals real-time. This research study aims to look at multi-modal emotion recognition by using facial expressions, speech audio signals and text.

The same concept applies to process speech audio signals. Future of this application would assist in identifying emotion thereby enabling individuals to manage stress and omit negative emotion.

## 1.8. Project Outline

The current study is divided into six chapters and below is outline of the study.

Chapter 1	Introduction	This chapter introduces the research question, the background of the study and develops aims and objectives
Chapter 2	Literature Review	An analysis of currently published academic literature that formulates the secondary research of the current study
Chapter 3	Conceptual Framework	A discussion of models and concepts that influence the current study are mentioned
Chapter 4	Methodology	A detailed discussion of architecture to fulfil the aim and objectives are mentioned here. This includes Data collection, Dataset description, Data analysis and model training
Chapter 5	Results	The results obtained from model evaluation and statistical analysis are mentioned in this chapter
Chapter 6	Discussion	The findings obtained from Chapter 5 are discussed in detail
Chapter 7	Conclusion	This chapter summarises the findings of the study
Chapter 8	Limitation and Future Works	This chapter list down the limitations and further developments based on the findings of the current study

*Table 1Dissertation Study Outline*

## 1.9. Project Plan and Implementation

The project plan and execution timelines are conducted as mentioned below in the Gantt Chart approved and reviewed as part of pilot study. This research study is based on pre-existing data samples and is implemented based on extensive research and literature review.



Task	24-Jun	08-Jul	29-Jul	05-Aug	12-Aug	19-Aug	26-Aug	02-Sep	05-Sep	08-Sep	09-Sep
Pilot Study	█	█									
Dataset & Preprocessing		█	█								
Developing Framework				█							
GUI Development					█						
Report writing Draft v1						█					
Model Evaluation							█				
Performance Evaluation								█			
Report Writing Draft v2									█		
User Testing										█	
Report Draft Preparation											█
Report Submission											█

Figure 3 Gantt Chart

## 2. LITERATURE REVIEW

The Literature review of this research topic, Multi-modal emotion recognition includes the detailed analysis of previously published literature on the use and combination of facial emotion recognition, speech emotion recognition and text to emotion recognition techniques and machine learning models. It has also been attempted to do literature review on Facial Emotion Recognition and Speech Emotion Recognition separately due the scarce resources for multi-modal research papers relevant to the current study. The study is segmented into following categories to attain critical insights into the focus areas identified through this study which are relevant to answer the research question.

1. Emotion classes used in the multi-modal emotion recognition papers
2. Data sources and data pre-processing activities used in the literature
3. Data analysis tools and visualization techniques used
4. Machine Learning models used and comparison of their results
5. Limitations of studies and recommendations for future

The papers used for the Literature study are taken from various databases like Google Scholar, IEE, Researchgate.net, Science Direct, APA PsycNet and Semantic Scholar.

### 2.1 Introduction

Understanding facial expressions is a crucial aspect of nonverbal communication and speech is one of the most important modes of verbal communication. A combination of both verbal and nonverbal communication make communication effective and hence, emotion recognition from both facial expressions and speech audio signals would result in more accurate prediction of emotion.

‘The human affective state is an indispensable component of human-human communication. Some human actions are activated by emotional state, while in other cases it enriches human communication. Thus, emotions play an important role by

allowing people to express themselves beyond the verbal domain.’

(Jackson, 2011, p.1)

According to Hess and Thibault, ‘emotions are relatively short duration intentional states that entrain changes in motor behaviour, physiological changes, and cognitions’ (2009, p.120). To analyse in detail, the emotion recognition models, it is important to understand the basic emotions and how they are mapped and categorized from emotion-specific response.

## 2.2. Research Material

‘Ekman & Friesen (1967,1969a) have hypothesized that the universals are to be found in the relationship between distinctive movements of facial muscles and particular emotions (such as happiness, sadness, anger, fear, surprise, disgust, interest). They suggested that cultural differences in facial behaviour would be seen because some of the stimuli which through learning become established as elicitors of emotions will vary across cultures, because the rules for controlling facial behaviour in particular social settings will vary across cultures, and because many of the consequences of emotional arouse will also vary with culture.’ (Ekman, 1970).

The six basic emotions obtained as conclusive evidence from their piece of research are the emotions of happiness, sadness, fear, anger, surprise, and disgust as described by Ekman and Friesen (1975) in their article “Universal Facial Expressions of Emotions”.

The article published by Paul Ekman and Wallace V Friesen in the year 1976 “Measuring Facial Movement” reports a new method of describing facial movement based on an anatomical analysis of facial action. ‘Since every facial movement is the result of muscular action, a comprehensive system could be obtained by discovering how each muscle of the face acts to change visible appearance. With that knowledge, it would be possible to analyse any facial movement into anatomically based minimal action units.’ (Ekman and Friesen, 1976, p 63).

This led to developing Facial Action Code (FAC) in which a list of muscles and how each muscle changes facial appearance were noted. In the next step Ekman and Friesen examined photographs of faces to determine if all muscular actions could be accurately distinguished. The figures below are snippets from Ekman and Friesen article with single action units in the Facial Action Code, instructions on how to make the facial movements and less precise account of 19 additional single action units.

65

PAUL EKMAN, WALLACE V. FRIESEN

**TABLE 1**  
Single Action Units (AU) in the Facial Action Code\*

<i>AU Number</i>	<i>FAC Name</i>	<i>Muscular Basis</i>
1.	Inner Brow Raiser	Frontalis, Pars Medialis
2.	Outer Brow Raiser	Frontalis, Pars Lateralis
4.	Brow Lowerer	Depressor Glabellae; Depressor Supercilli; Corrugator
5.	Upper Lid Raiser	Levator Palpebrae Superioris
6.	Cheek Raiser	Orbicularis Oculi, Pars Orbitalis
7.	Lid Tightener	Orbicularis Oculi, Pars Palebralis
9.	Nose Wrinkler	Levator Labii Superioris, Alaeque Nasi
10.	Upper Lid Raiser	Levator Labii Superioris, Caput Infraorbitalis
11.	Masolabial Fold Deepener	Zygomatic Minor
12.	Lip Corner Puller	Zygomatic Major
13.	Cheek Puffer	Caninus
14.	Dimpler	Buccinator
15.	Lip Corner Depressor	Triangularis
16.	Lower Lip Depressor	Depressor Labii
17.	Chin Raiser	Mentalis
18.	Lip Puckerer	Incisivii Labii Superioris; Incisive Labii Inferioris
20.	Lip Stretcher	Risorius
22.	Lip Funneler	Orbicularis Oris
23.	Lip Tightner	Orbicularis Oris
24.	Lip Pressor	Orbicularis Oris
25.	Lips Part	Depressor Labii, or Relaxation of Mentalis or Orbicularis Oris
26.	Jaw Drop	Maseter; Temporal and Internal Pterygoid Relaxed
27.	Mouth Stretch	Pterygoids; Digastric
28.	Lip Suck	Orbicularis Oris

\*The numbers are arbitrary and do not have any significance except that 1-7 refers to brows, forehead or eyelids.

Figure 4 Single Action Units in the Facial Action Code (source - (Ekman and Friesen, 1976,p.65).

TABLE 2  
An Example of the Information Given in the FAC for Each Action Unit

ACTION UNIT 15—Lip Corner Depressor	
The muscle underlying AU 15 emerges from the side of the chin and runs upwards attaching to a point near the corner of the lip. In AU 15 the corners of the lips are pulled down. Study the anatomical drawings which show the location of the muscle underlying this AU.	
<ol style="list-style-type: none"> <li>(1) Pulls the corners of the lips down.</li> <li>(2) Changes the shape of the lips so they are angled down at the corner, and usually somewhat stretched horizontally.</li> <li>(3) Produces some pouching, bagging, or wrinkling of skin below the lips' corners, which may not be apparent unless the action is strong.</li> <li>(4) May flatten or cause bulges to appear on the chin boss, may produce depression medially under the lower lip.</li> <li>(5) If the nasolabial furrow* is permanently etched, it will deepen and may appear pulled down or lengthened.</li> </ol>	
The photographs in FAC show both slight and strong versions of this Action Unit. Note that appearance change (3) is most apparent in the stronger versions. The photograph of 6+15 shows how the appearance changes due to 6 can add to those of 15. Study the film of AU 15.	
<i>How To Do 15</i>	
Pull your lip corners downwards. Be careful not to raise your lower lip at the same time—do not use AU 17. If you are unable to do this, place your fingers above the lip corners and push downwards, noting the changes in appearance. Now, try to hold this appearance when you take your fingers away.	
<i>When To Score Slight Versions of 15</i>	
Elongating the mouth is irrelevant, as it may be due to AU 20, AU 15, or AU 15+20.	
<ol style="list-style-type: none"> <li>(1) If the lip line is straight or slightly up in neutral face, then the lip corners must be pulled down at least slightly to score 15.</li> <li>or (2) If lip line is slightly or barely down in neutral face, then the lip corners must be pulled down slightly more than neutral and not the result of AU 17 or AU 20.</li> </ol>	

\*A wrinkle extending from beyond the nostril wings down to beyond the lip corners.

Copyright © Ekman & Friesen, 1976

Figure 5 Example of information given in the FAC for each Action Unit (source - (Ekman and Friesen, 1976, p.66)

TABLE 3  
More Grossly Defined AUs in the Facial Action Code

AU Number	FAC Name
19.	Tongue Out
21.	Neck Tightener
29.	Jaw Thrust
30.	Jaw Sideways
31.	Jaw Clencher
32.	Lip Bite
33.	Cheek Blow
34.	Cheek Puff
35.	Cheek Suck
36.	Tongue Bulge
37.	Lip Wipe
38.	Nostril Dilator
39.	Nostril Compressor
41.	Lid Droop
42.	Slit
43.	Eyes Closed
44.	Squint
45.	Blink
46.	Wink

Figure 6 More grossly defined AUs in the Facial Action Code (source - (Ekman and Friesen, 1976,p.69).

‘Work on facial expressions of emotions (Calder, Burton, Miller, Young, & Akamatsu, 2001) and emotionally inflected speech (Banse & Scherer, 1996) has successfully delineated some of the physical properties that underlie emotion recognition. To identify the acoustic cues used in the perception of nonverbal emotional expressions like laughter and screams, an investigation was conducted into vocal expressions of emotion, using nonverbal vocal analogues of the "basic" emotions (anger, fear, disgust, sadness, and surprise; (Ekman & Friesen, 1971), (Scott et al., 1997), and of positive affective states (Ekman, 1992, 2003); (Sauter & Scott, 2007) (K Scott et al.2010).

(K Scott et al. 2010), In this research study, firstly an emotional stimulus was categorized and scored to ensure that listeners could identify and rate the sounds to create confusion matrices. A principal components analysis of the rating data yielded two underlying dimensions, correlating with the perceived valence and arousal of the sounds. Secondly, acoustic properties of the amplitude, pitch, and spectral profile of the stimuli were measured. A discriminant analysis procedure established that these acoustic measures provided sufficient discrimination between expressions of emotional categories to permit accurate statistical classification. Multiple linear regressions with participants' subjective ratings of the acoustic stimuli showed that all classes of emotional ratings could be predicted by some combination of acoustic measures and that most emotion ratings were predicted by different constellations of acoustic features. The results demonstrate that, similarly to affective signals in facial expressions and emotionally inflected speech, the perceived emotional character of affective vocalizations can be predicted based on their physical features.

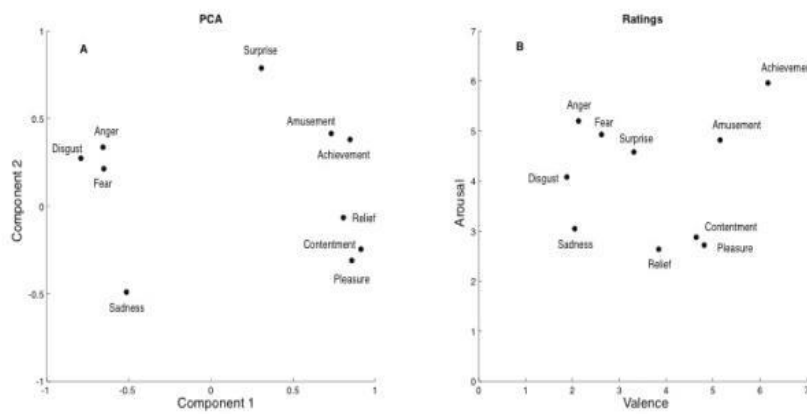


Figure 6 A) Principal component analysis for positive and negative emotional vocalizations. B) Average ratings on the dimensions arousal and valence for each category of emotional sounds (n=20). (Source- K Scott et al.)

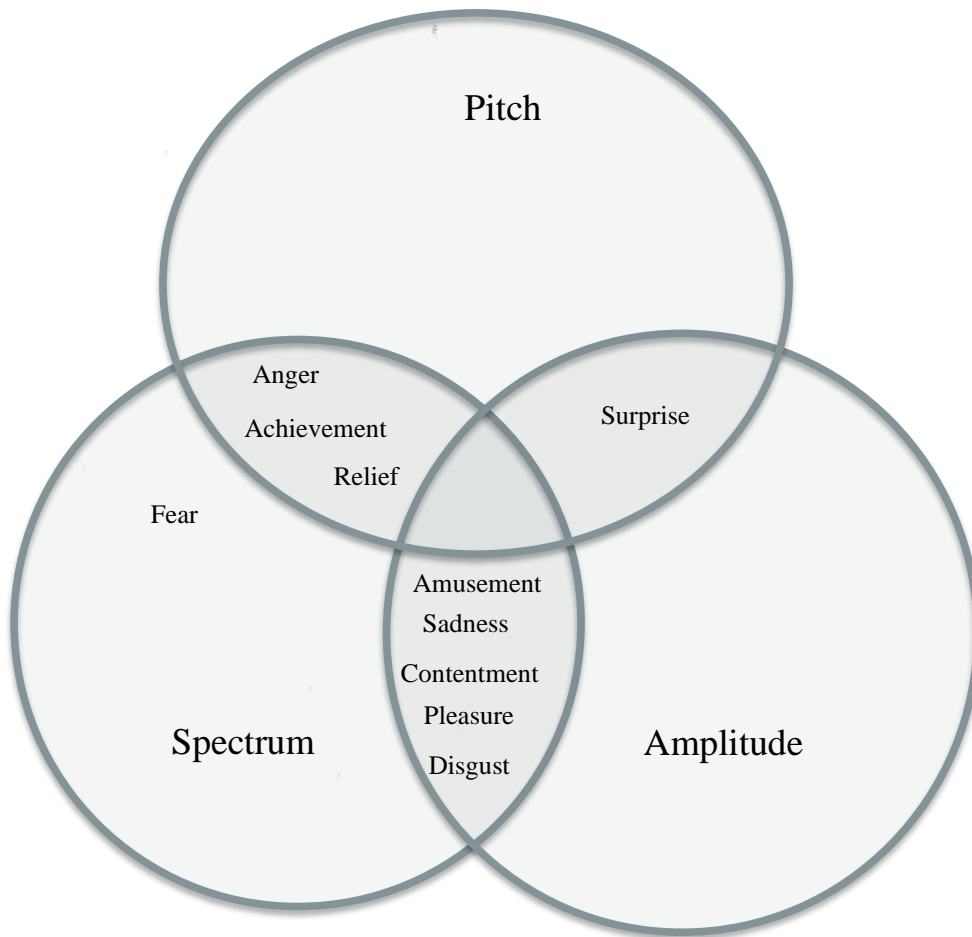


Figure 7 Venn diagram showing classes of acoustic information that are used to predict participants' ratings for each of the emotional scales. (Source- K Scott et al.)

The various aspects on which an audio emotion recognition is conducted are discussed here. An important aspect which was investigated recently was the emotion representation of audio features. Audio features such as pitch (Busso, Lee, & Narayanan, 2009; Devillers, Vidrascu, & Layachi, 2010), log energy, zero crossing rate (Chien Hung, Ping Tsung, & Chen, 2010; Chih-Chang, Chien-Hung, Ping-Tsung, & Chen, 2010), spectral features (Wong & Sridharan, 2001), voice quality (Lugger & Bin, 2007), jitter(Xi et al., 2007), etc. have been discovered useful in emotion recognition.

Latest trends in research of audio emotion recognition emphasized the use of combination of distinctive features to achieve improvement in the recognition performance. A researcher (Yeh, Pao, Lin, Tsai, and Chen, 2011) developed a system to recognize five emotions using up to 128 audio features. The databases used are IEMOCAP and AIBO to build a model of multiple layers and 384 features were extracted such as zero crossing rate, root-mean-square energy, voice quality, pitch, MFCC.' (Ooi et al.).

Speech features can be classified into three groups: vocal tract system, prosodic, and excitation source features. Vocal tract system feature like Log Frequency Power Coefficients (LFPC), MFCC and LPCC when used in combination can recognize up to 6 emotions on SAVEE Dataset. Standard databases such as Emo-DB, eNTER-FACE'05, and RML emotion database are frequently used by researchers. HMM, Support Vector Machine (SVM) and Neural Network classifiers are used in this research study to model sequential data. The Neural Network can be divided into three categories, Recurrent Neural Network (RNN) (Wei & Guanglai, 2009), Multi-Layer Perception (MLP) Neural Network (Lu & Wei, 2004), and RBF Neural Network (Chen, Cowan, & Grant, 1991). The dataset, emotions used, models and their accuracy percentage from this study is tabled below.

Dataset	eNTERFACE'05 database	RML database
---------	-----------------------	--------------



Classifier Model	Recognition Rate	
Hidden Markov Model (HMM)	-	52%
RBF Neural Network (proposed audio emotion recognition system)	<b>75.8%</b>	<b>68.57%</b>
Support Vector Machine(SVM)	62.8%	-

*Table 2 Dataset and Classifier model information for the audio emotion recognition model (source - Ooi et al., 2014)*

SVM was used on eNTERFACE'05 database and HMM was used on RML Database, RBF neural network outperformed on both datasets.

(Hossain and Muhammad, 2019)'' Emotion recognition using deep learning approach from audio–visual emotional big data'' proposes an audio-visual emotion recognition system by using one deep neural network to extract features and another deep network to fuse features and Support Vector Machine (SVM) classifier is used to perform final classification. Below block diagram with the proposed emotion recognition system and detailed pre-processing steps of speech and video in the proposed system clearly covers the entire process.

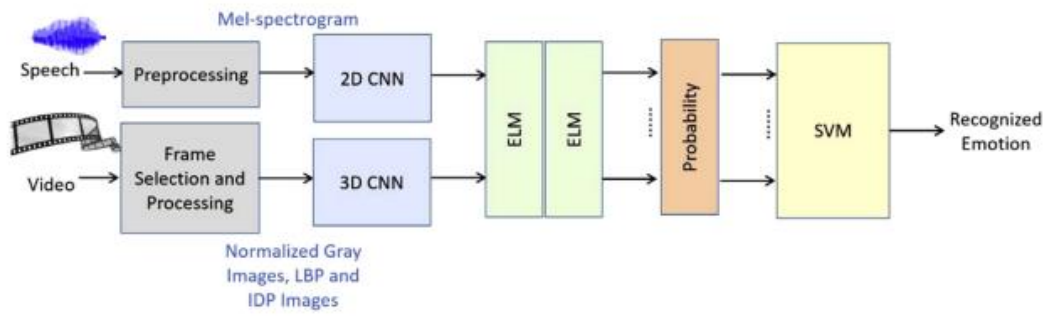


Figure 8 Block Diagram of the proposed audio-visual emotion recognition system (source - Hossain and Muhammad, 2019)

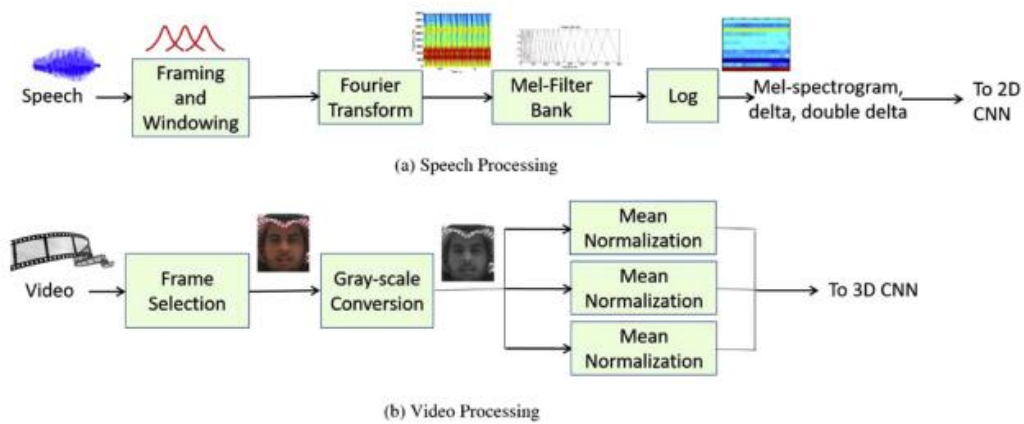


Figure 9 Overall Data Processing of the proposed audio-visual emotion recognition system (source - Hossain and Muhammad, 2019)

In the proposed system, Mel spectrogram is obtained from speech signal converted to grey image is inputted to the 2D CNN model. For the video signal, key frames are selected to calculate histograms and a face detection algorithm, Viola-Jones algorithm is applied to crop the face. 16 key frames are selected from one video segment which are converted to grey image and sampled to  $227 \times 227$  and is inputted to the 3D CNN model. Two ELM functions based on a single hidden layer feed-forward network is used to fuse the scores from two CNN models and is inputted to SVM classifier which achieved 99.9% accuracy. Big Data of emotion and the eNTERFACE database were used for the evaluation.

(Hassan et al., 2019) proposed a physiological signal based human emotion recognition algorithm which is different in context from the methodology of the current study.

However, an analysis of different concepts would give another perspective to approach the research problem. The subjects are asked to watch a set of videos and physiological data is collected using EDA, PPG and zEMG sensors. A deep neural network model is used. The down-sampled fused sensor observations are inputted to the deep neural network which predicts 5 classes of emotion with 89.53% accuracy

(Singh and Fang, 2020) uses IEMOCAP dataset of audio-visual data of 5 men and 5 women in total, where each sentence is labelled with one emotion. Spectrograms were generated in two segments with and without noise clean up and were data augmented by cropping and rotation. Video frames were chosen in accordance to the speech spectrogram images and were cropped and resized. CNN+RNN Model was used for audio spectrograms and 3D CNN was used for video frames. Different model architecture combinations were used in audio (CNN, CNN+RNN & CNN+LSTM) and audio video signals (CNN+RNN+3DCNN.) to compare accuracy and the latter performed better with 71.75% accuracy and predicted happy, sad, angry and neutral emotion classes.

(Chang and Skarbek, 2021) proposes a multi-modal emotion recognition system that uses a novel end to end Deep Neural Network where data pre-processing includes Spatial data augmentation and Time dependent data augmentation were performed on visual frames and vocal frames. RAVDESS dataset and Crema-d dataset were used and 10-folder inter validation concepts were used for splitting train and test data set. Results achieved an average accuracy of 91.4% on the RAVDESS dataset and 83.15% on the Crema-d dataset.

(Dobrišek et al., 2013) The multi-modal emotion recognition system introduced in this paper consists of an audio and a video sub-system where each subsystem processes their input and produce a matching score. The two scores are then fused using sum rule and product rule fusion schemes which is then inputted to a SVM classifier. eNTERFACE'05 corpus dataset with 6 emotion classes were used. Product rule fusion scheme with SVM achieved maximum accuracy of 77.5%.

(Busso et al., 2004) The proposed system pre-processes audio and video features and performs two different approaches to fuse audio and video features. Feature-level fusion, in which a single classifier with features of both modalities are used and, decision level fusion, in which a separate classifier is used for each modality, and the outputs are combined using different criteria such as weight combining rule, product combining rule, averaging combining rule and product combining rule. Overall accuracy is almost same for the proposed system with different rules applied and the maximum among all is product combining with 88.9% accuracy. The database used was recorded by an actor with markers attached to capture visual information. Four classes of emotions, Happiness, Anger, Sadness and Neutral were recognized.

(Ebrahimi Kahou et al., 2015) The proposed system uses 3 different structures of CNN architecture namely deep structure with 3x3 filter size, three-layer CNN with 5x5 filters and another three-layer CNN with 9x9 filter size. Different combinations of dataset were used such as Toronto Face Database (TFD) with 4,178 images and the Facial Expression Recognition dataset (FER2013) containing 35,887 images, both with seven basic expressions: angry, disgust, fear, happy, sad, surprise and neutral. The pre-processing process detected five facial key points for all images using CNN cascade method. A mean shape was computed by averaging the coordinates of key points for each dataset. For video, Recurrent Neural Network (RNN) was used to aggregate frame features due to its ability to deal with variable frames. Although the focus of this research study was to perform emotion recognition on video, audio features were also extracted from the video clips using open-source Emotion and Affect Recognition(openEAR) toolkit. The resulting test performance was only 49.907% and the researcher assumes it could be due to overfitting.

(Li et al., 2020) In this paper, researcher proposed a multimodal attention based BLSTM network architecture for efficient multimodal emotion recognition from spectrograms. Attention-based BLSTM-RNNs is capable of learning feature representations and modelling temporal dependencies between their activation. The experiments show that the proposed model performed competitive results on the AMIGOS dataset. In this model,

spectrogram is used as the input of the LSTM framework, then Attention-based BLSTM-RNN layers will extract sequential features from spectrogram considering it as an image sequence channel. These features are then fed into a DNN to predict the probability of emotional output. A decision level fusion strategy based DNN is finally used to recognize the final emotion state. Different weights scheme was used to evaluate the performance. This model achieved 82% and outperformed over Naïve Bayes model.2.3. Conclusion The Literature Review for this research study has been extremely helpful to gain a clear understanding of the academic research works carried out in the field of Artificial Intelligence.

### 2.3. Conclusion

(RSIS, 2019) This research work attempt to find out the real time applications of Human Emotion Recognition and clearly stresses the importance of further studies in the field of human emotion recognition due to the relevance of facial emotion recognition, audio emotion recognition , text to emotion, study of physiological response in the areas of Education, Medical Science, Mental Health.

### 3. Conceptual Framework

The proposed system constitutes two Convolutional Neural Network models to perform Facial Emotion Recognition and Speech Emotion Recognition. Below architecture diagram depicts the process and the underlying concepts which is discussed in detail.

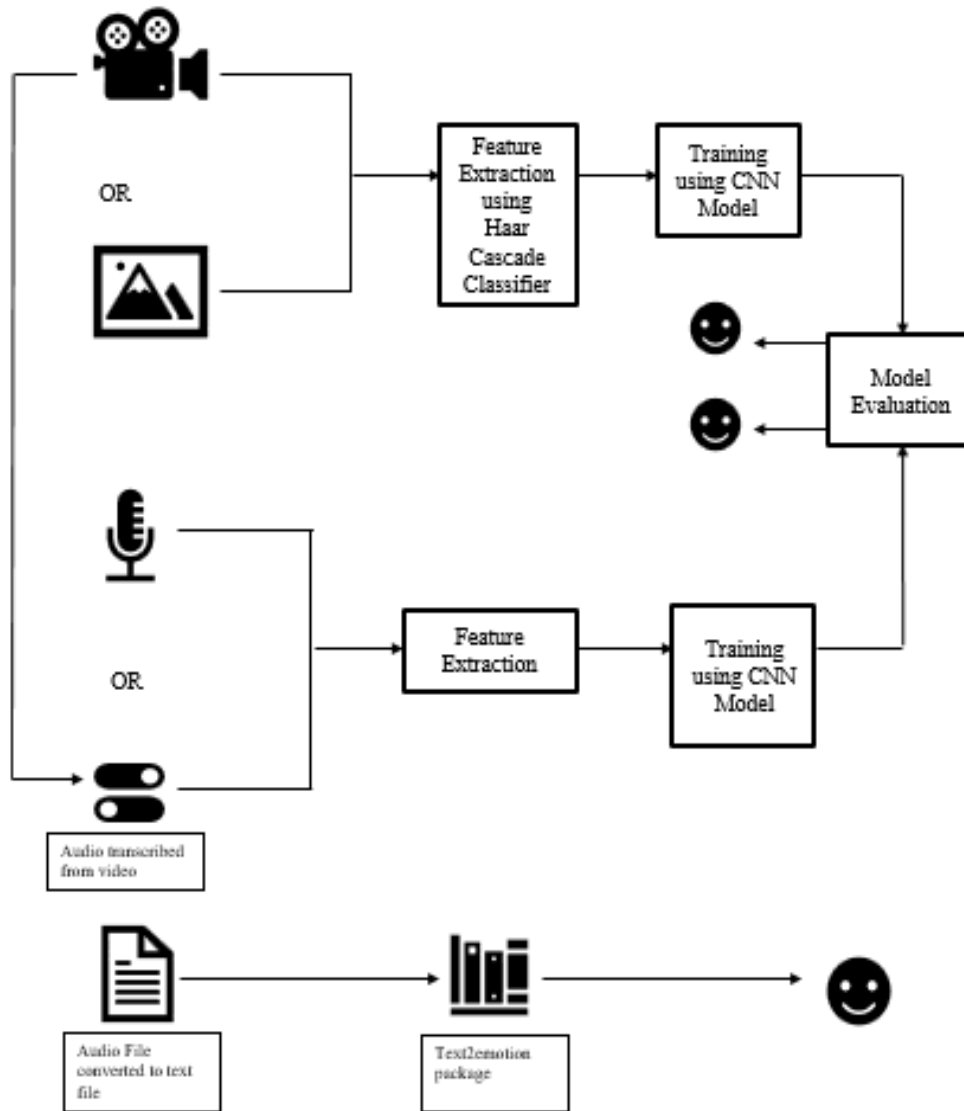


Figure 10 Multi-modal Emotional Recognition Model Architecture

Based on the six basic emotions described by Ekman and Friesen (1975) in their research study “Universal Facial Expressions of Emotions”, six emotion classes were selected to be used in the current research to perform multi-modal emotion recognition consisting of Facial Emotion Recognition and Speech Emotion Recognition. The classes of emotions are happiness, angry, sad, neutral, disgust and surprise.

### 3.1. Facial Emotion Recognition

Facial expression is a form of non-verbal communication which can be recognized by a human brain which is referred to as emotional intelligence. With the advent of machine learning and exponential growth in the real time application in Artificial Intelligence, an attempt to develop emotion recognition application began to arise. This was first conceptualized with the development of object detection framework proposed by Paul Viola and Michael Jones in 2001 which can accurately detect objects particularly faces and is still widely used in CNN based machine learning models. This framework combines the concept of Haar-Like-Features, Integral images, cascade classifier and Adaboost algorithm to develop successfully predicting real time object detection model.

#### 3.1.1 Viola-jones Object Detection Algorithm

The first and foremost step in emotion recognition is face detection for which Viola Jones Object Detection Algorithm is used. Object Detection or Face Detection (in this case) is a binary classification problem, and it is very important to have a classifier constructed which can minimize the misclassification risk. The algorithm must also be able to minimize false negative and false positive rates to achieve a satisfactory performance.

The Viola Jones Algorithm has four main steps which are as follows:

- a. Selecting Haar-like features
- b. Creating an integral image

- c. Running Adaboost Training
- d. Creating classifier cascades

### 3.1.1.a. Haar-like feature

(Wang, 2014) The Viola- Jones Algorithm used Haar-like features which is a scalar product between image and Haar-like templates.

Let I and P denote an image and pattern, both of same size N x N, then the feature associated with Pattern P of Image I is defined by,

$$\sum_{1 \leq i \leq N} \sum_{1 \leq j \leq N} I(i, j) 1_{P(i, j) \text{ is white}} - \sum_{1 \leq i \leq N} \sum_{1 \leq j \leq N} I(i, j) 1_{P(i, j) \text{ is black}}$$

To neutralise the effect of different lighting conditions, all the images must be mean and variance normalized beforehand. Those images with variance lower than one with little or no information are omitted.

In the below picture, the background of a template like (b) is painted grey to highlight the pattern's support. Only those pixels marked in black or white are used when the corresponding feature is calculated.

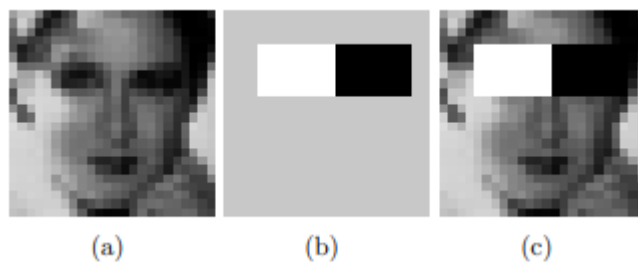


Figure 11 Haar-Like Features example picture (source - (Wang,2014))

The number of features one can draw from an image depend on their relative positions. For instance, a 24 x 24 image has 43200, 27600, 43200, 27600 and 20736



features of category (a), (b), (c), (d) and (e) respectively, hence 162336 features in total.

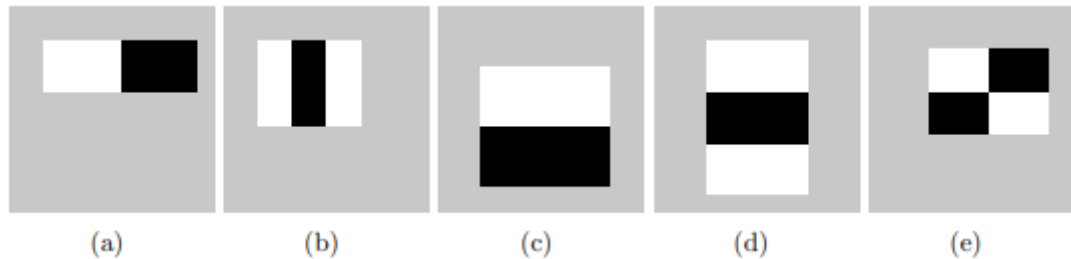


Figure 12 Five Haar-like Patterns (source-(Wang,2014))

In the above picture, only five patterns are considered, and the derived features will have all the details required to characterize a face. Below is the algorithm to compute Haar-like feature vector for a  $24 \times 24$  image. (Wang,2014)

There are 3 types of Haar-like features that Viola and Jones identified in their research, these features help the machine to understand what the image is.

1. Edge features
2. Line-features
3. Four-sided features

Some, when the images are received, each feature has a value of its own.

**(Wang,2014) Algorithm - Computing a  $24 \times 24$  image's Haar-like feature vector**

- 1: Input: a  $24 \times 24$  image with zero mean and unit variance
- 2: Output: a  $d \times 1$  scalar vector with its feature index  $f$  ranging from 1 to  $d$
- 3: Set the feature index  $f \leftarrow 0$
- 4: Compute feature type (a)
- 5: for all  $(i, j)$  such that  $1 \leq i \leq 24$  and  $1 \leq j \leq 24$  do
- 6: for all  $(w, h)$  such that  $i + h - 1 \leq 24$  and  $j + 2w - 1 \leq 24$  do
- 7: compute the sum  $S1$  of the pixels in  $[i, i + h - 1] \times [j, j + w - 1]$
- 8: compute the sum  $S2$  of the pixels in  $[i, i + h - 1] \times [j + w, j + 2w - 1]$
- 9: record this feature parametrized by  $(1, i, j, w, h)$ :  $S1 - S2$
- 10:  $f \leftarrow f + 1$
- 11: end for
- 12: end for
- 13: Compute feature type (b)
- 14: for all  $(i, j)$  such that  $1 \leq i \leq 24$  and  $1 \leq j \leq 24$  do

```

15: for all (w, h) such that  $i + h - 1 \leq 24$  and  $j + 3w - 1 \leq 24$  do
16: compute the sum S1 of the pixels in  $[i, i + h - 1] \times [j, j + w - 1]$ 
17: compute the sum S2 of the pixels in  $[i, i + h - 1] \times [j + w, j + 2w - 1]$ 
18: compute the sum S3 of the pixels in  $[i, i + h - 1] \times [j + 2w, j + 3w - 1]$ 
19: record this feature parametrized by (2, i, j, w, h):  $S1 - S2 + S3$ 
20:  $f \leftarrow f + 1$ 
21: end for
22: end for
23: Compute feature type (c)
24: for all (i, j) such that  $1 \leq i \leq 24$  and  $1 \leq j \leq 24$  do
25: for all (w, h) such that  $i + 2h - 1 \leq 24$  and  $j + w - 1 \leq 24$  do
26: compute the sum S1 of the pixels in  $[i, i + h - 1] \times [j, j + w - 1]$ 
27: compute the sum S2 of the pixels in  $[i + h, i + 2h - 1] \times [j, j + w - 1]$ 
28: record this feature parametrized by (3, i, j, w, h):  $S1 - S2$ 
29:  $f \leftarrow f + 1$ 
30: end for
31: end for
32: Compute feature type (d)
33: for all (i, j) such that  $1 \leq i \leq 24$  and  $1 \leq j \leq 24$  do
34: for all (w, h) such that  $i + 3h - 1 \leq 24$  and  $j + w - 1 \leq 24$  do
35: compute the sum S1 of the pixels in  $[i, i + h - 1] \times [j, j + w - 1]$ 
36: compute the sum S2 of the pixels in  $[i + h, i + 2h - 1] \times [j, j + w - 1]$ 
37: compute the sum S3 of the pixels in  $[i + 2h, i + 3h - 1] \times [j, j + w - 1]$ 
38: record this feature parametrized by (4, i, j, w, h):  $S1 - S2 + S3$ 
39:  $f \leftarrow f + 1$ 
40: end for
41: end for
42: Compute feature type (e)
43: for all (i, j) such that  $1 \leq i \leq 24$  and  $1 \leq j \leq 24$  do
44: for all (w, h) such that  $i + 2h - 1 \leq 24$  and  $j + 2w - 1 \leq 24$  do
45: compute the sum S1 of the pixels in  $[i, i + h - 1] \times [j, j + w - 1]$ 
46: compute the sum S2 of the pixels in  $[i + h, i + 2h - 1] \times [j, j + w - 1]$ 
47: compute the sum S3 of the pixels in  $[i, i + h - 1] \times [j + w, j + 2w - 1]$ 
48: compute the sum S4 of the pixels in  $[i + h, i + 2h - 1] \times [j + w, j + 2w - 1]$ 
49: record this feature parametrized by (5, i, j, w, h):  $S1 - S2 - S3 + S4$ 
50:  $f \leftarrow f + 1$ 
51: end for
52: end for

```

### 3.1.2. Convolutional Neural Network (CNN) Model Architecture

Convolutional Neural Network (CNN) is a network architecture for deep learning which learns directly from data without the need for manual feature extraction. CNNs are useful for object detection, faces by finding patterns in images to perform recognition. CNN model is also useful for classifying non-image data such as audio, time series and signal data. (“What Is a Convolutional Neural Network?”)

Advantages of using CNNs for deep learning are due to the below three factors:

1. Eliminate the need for manual feature extraction
2. Produce highly accurate recognition results
3. CNNs can be retrained and enable to build on pre-existing networks

#### 2.1.2.a. CNN Workflow

A CNN network can have ten or hundreds of layers and each layer learn to detect different features of an image. The output of each convolved image is used as the input to next layer. The filters can start with simple features and may increase in complexity to features that uniquely define the object.

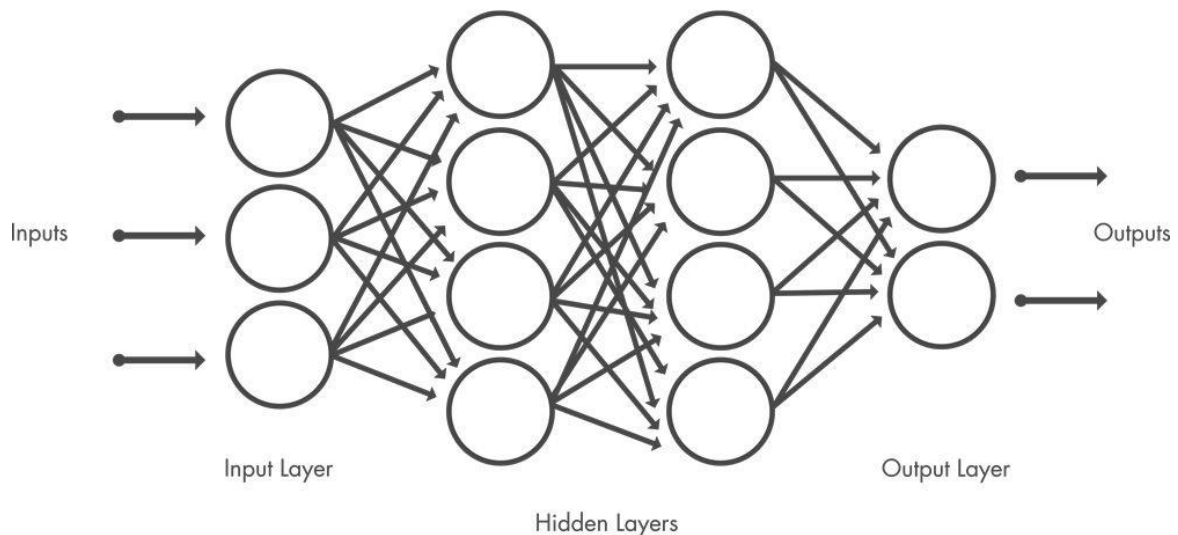


Figure 13 CNN Layers (source - <https://uk.mathworks.com/discovery/convolutional-neural-network-matlab.html#how-they-work>)

Three of the most common layers of CNN architecture are:

1. Convolution layer allows the input images to pass through a set of convolutional filters, each of which activates certain features from the images.
2. Rectified linear unit (ReLU) allows for faster and more effective training by mapping negative values to zero and maintaining positive values. This is sometimes referred to as activation, because only the activated features are carried forward into the next layer.
3. Pooling simplifies the output by performing nonlinear down sampling, reducing the number of parameters that the network needs to learn.

These operations are repeated over tens or hundreds of layers, with each layer learning to identify different features.

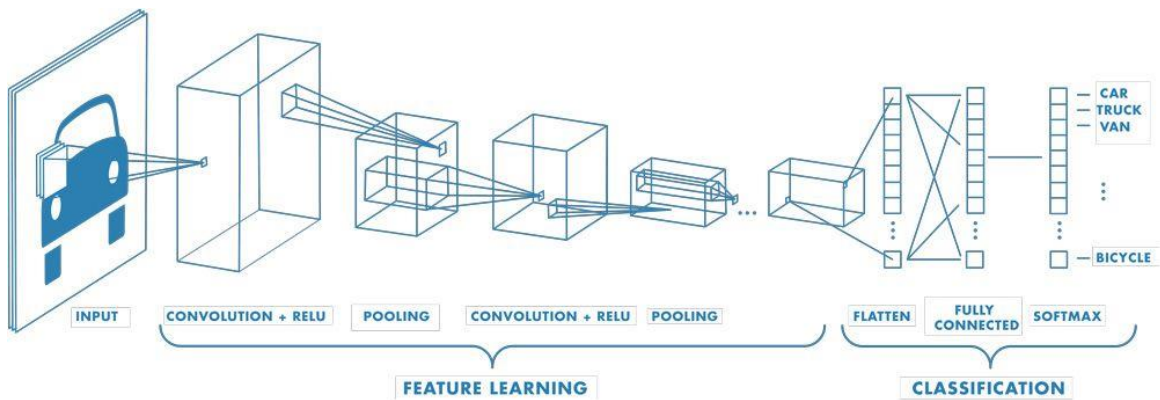


Figure 14 CNN network with many convolutional layers(source - <https://uk.mathworks.com/discovery/convolutional-neural-network-matlab.html#how-they-work>)

## 4. Methodology

This chapter discusses in detail, the steps carried out to develop a multi-modal emotional recognition model by identifying a reliable and suitable dataset required to train and test a facial emotion recognition and speech emotion recognition model. The research approach was primarily to gather secondary dataset which is already been created solely for the purpose of emotion recognition research purposes. Greater attention was given to ensure that sufficient data was available to train the model with the chosen classes of emotions.

### 4.1. Introduction

The current research focuses on an inductive approach. The step-by-step approach used for the development of multi-modal emotion recognition is discussed in detail below.

For the ease of access and security and to resolve low disk space issues, google colab was used in this research to perform the tasks. Google Colab provided collaborative interface and computing resources to perform dataset storage and model training.

This section of current research is divided into four which consist of:

1. Facial Emotion Recognition Model
2. Speech Emotion Recognition Model
3. Text2emotion Package
4. GUI Application

### 4.2. Facial Emotion Recognition Model

The development of facial emotion recognition model includes the below steps.

1. Importing libraries and packages
2. Dataset
3. Data Pre-processing
4. Creating model structure
5. Training the CNN model
6. Saving model structure

7. Saving trained model
8. Model Evaluation

#### 4.2.1. Importing libraries and packages

```
!pip install numpy
!pip install opencv-python
!pip install keras
!pip install pillow
!pip install tensorflow==2.8
!pip3 install --upgrade tensorflow
!apt install --allow-change-held-packages libcudnn8=8.1.0.77-1+cuda11.2

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (1.21.6)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: opencv-python in /usr/local/lib/python3.7/dist-packages (4.6.0.66)
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from opencv-python) (1.21.6)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: keras in /usr/local/lib/python3.7/dist-packages (2.9.0)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-packages (7.1.2)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting tensorflow==2.8
  Downloading https://us-python.pkg.dev/colab-wheels/public/tensorflow/tensorflow-2.8.0%2Bzzzcolab20220506162203-cp37-cp37m-linux_x86_64.whl (668.3 MB)
  668.3 MB 17 kB/s
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.8) (1.1.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.8) (3.3.0)
Requirement already satisfied: tf-estimator-nightly==2.8.0.dev2021122109 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.8) (2.8.0.dev2021122109)
Requirement already satisfied: flatbuffers>=1.12 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.8) (1.12)
Collecting tensorboard<2.9,>=2.8
  Downloading tensorboard-2.8.0-py3-none-any.whl (5.8 MB)
  5.8 MB 5.2 MB/s
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.8) (0.26.0)
```

Figure 15 Import Libraries and Packages

```
# import required packages
import cv2
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dense, Dropout, Flatten
from tensorflow.keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
```

Figure 16 FER - Importing libraries & packages

a. Numpy Package

(Google Colab syntax: !pip install numpy)

“NumPy is the short form for Numerical Python, one of the basic and fundamental packages in Python language. It provides support for large multidimensional arrays matrices along with a collection of high-level mathematical functions.”

(Vidhya,2020)

b. OpenCV-Python

(Google Colab syntax: !pip install opencv-python)

“OpenCV-Python is a library of Python bindings designed to solve computer vision problems. OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.” (Learning,2021)

c. Keras

(Google Colab syntax: `!pip install keras`)

“Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make implementing deep learning models as fast and easy as possible for research and development. It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks.” (Brownlee)

d. Pillow package

(Google Colab syntax: `pip install opencv-python`)

“Python Imaging Library (also known as PIL or Pillow) is a free and open-source library use for image manipulation and processing in Python. Pillow is the newer version, built upon PIL with greater support for Operating Systems and Python3.” (“Python Pillow (PIL) Tutorial - Image Manipulation”)

e. TensorFlow

(Google Colab syntax: `pip install tensorflow==2.8`)

“TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms.” (“Tensorflow”)

#### 4.2.2. Dataset

To mount the contents of Google Drive where the dataset is stored

```
# Run this cell to mount your Google Drive.  
from google.colab import drive  
drive.mount('/content/drive')
```

Mounted at /content/drive

Figure 17 Mounting Google Drive to access dataset

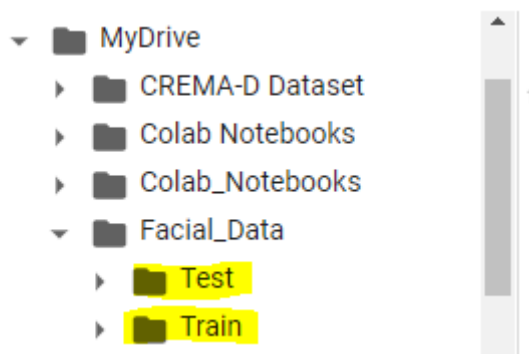


Figure 18 Dataset - Train and Test Folders

“The dataset used to train Facial Emotion Recognition model is FER 2013 Dataset which contains 28,709 facial RGB images of different expressions with size restricted to 48×48, and the main labels of it can be divided into 7 types: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral. The Disgust expression has the minimal number of images which is 600, while other labels have nearly 5,000 samples each.” (“Papers with Code - FER2013 Dataset”)



Figure 19 FER-2013 Dataset (source - Researchgate.net)



### 4.2.3. Data Pre-processing

```
# Initialize image data generator with rescaling
train_data_gen = ImageDataGenerator(rescale=1./255)
validation_data_gen = ImageDataGenerator(rescale=1./255)
```

Figure 20 ImageDataGenerator

‘ImageDataGenerator method is a “standardize” method which performs in-place normalization to the batch of inputs. It is an important data augmentation step. Different normalization techniques including centering the sample, rescaling input or performing zca whitening are all performed by ImageDataGenerator method.’ (Kang & Atul)  
Here, this parameter is used to scale array of original image pixel values to be between [0,1] and specify the parameter rescale=1./255.

```
# Preprocess all test images
train_generator = train_data_gen.flow_from_directory(
    '/content/drive/MyDrive/Facial_Data /Train',
    target_size=(48, 48),
    batch_size=64,
    color_mode="grayscale",
    class_mode='categorical')

# Preprocess all train images
validation_generator = validation_data_gen.flow_from_directory(
    '/content/drive/MyDrive/Facial_Data /Test',
    target_size=(48, 48),
    batch_size=64,
    color_mode="grayscale",
    class_mode='categorical')
```

Figure 21 Data Pre-processing - Train and Test Image Dataset

‘Keras API has ImageDataGenerator class which allows the users to perform image augmentation. The ImageDataGenerator class has three methods **flow** (), **flow\_from\_directory()** and **flow\_from\_dataframe()** to read the images from a big

numpy array and folders containing images.’ (J)

The `flow_from_directory ()` method has the following attributes:

- The directory must be set to the path where your ‘n’ classes of folders are present.
- The `target_size` is the size of your input images; every image will be resized to this 48 x 48 size
- `color_mode`: if the image is either black and white or grayscale set “grayscale” or if the image has three color channels, set “rgb”.
- `batch_size`: No. of images to be yielded from the generator per batch.
- `class_mode`: Set “binary” if only two classes to predict, if not set to “categorical”,
- `shuffle`: Set True if order of the image that is being yielded need to be shuffled, else set False.
- `seed`: Random seed for applying random image augmentation and shuffling the order of the image. (J)

#### 4.2.4. Creating model structure

```
# create model structure
emotion_model = Sequential()

emotion_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48, 48, 1)))
emotion_model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Flatten())
emotion_model.add(Dense(1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(7, activation='softmax'))

cv2.occl.setUseOpenCL(False)

emotion_model.compile(loss='categorical_crossentropy', optimizer=Adam(lr=0.0001, decay=1e-6), metrics=['accuracy'])
```

Figure 22 CNN Model Structure for Facial Emotion Recognition

A Sequential model is created with a plain stack of layers where each layer has exactly one input tensor and one output tensor.

Convolution is a mathematical operation that require two inputs, if two inputs are named x and f, then the convolution operation takes selective part of input x, transforms it into Y using the values of filter f.

i. e.  $Y = xf$

Here, x is a 2-dimensional array of images and f is a 2D matrix and the content of each of these matrices will evaluate the content of the output image Y. For the ease of mathematical computation, the sizes are always odd numbers and symmetrical.

Here, the model used 32 filters stacked one after another and each filter is of size 3x3, and model will learn a total of 288 parameters(32x3x3)

A Pooling layer is added where MaxPooling method is used to reduce the training parameters by half in width and height.

A fully connected layer will add a final layer to the model with dimensions equal to the number of categories of the classification problem.

The compile method requires several parameters. The loss parameter is specified to have type 'categorical\_crossentropy' where Categorical cross entropy is a loss function that is used in multi-class classification tasks. The metrics parameter is set to 'accuracy' and Adam optimizer is used for training the network. Adam Optimizer - Adaptive Moment Estimation is an algorithm for optimization technique for gradient descent.

#### 4.2.5 Training the CNN model

```
# Train the neural network/model
emotion_model_info = emotion_model.fit_generator(
    train_generator,
    steps_per_epoch=28709 // 64,
    epochs=50,
    validation_data=validation_generator,
    validation_steps=7178 // 64)
```

Figure 23 Training the CNN Model

The next step is to train the model by parsing the training data. A subset of the dataset is used for validation. Epochs are the number of iterations in which training takes place. In each iteration, a limited number of images are passed to the model in batches (defined as

batch\_size which is 64).

```
□ Found 28719 images belonging to 7 classes.
Found 7178 images belonging to 7 classes.
/usr/local/lib/python3.7/dist-packages/keras/optimizers/optimizer_v2/adam.py:110: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adam, self).__init__(name, **kwargs)
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:57: UserWarning: `Model.fit_generator` is deprecated and will be removed in Keras 3.0.0. Use `Model.fit` instead.
Epoch 1/50
448/448 [=====] - 5941s 13s/step - loss: 1.7868 - accuracy: 0.2713 - val_loss: 1.6584 - val_accuracy: 0.3643
Epoch 2/50
448/448 [=====] - 51s 114ms/step - loss: 1.6118 - accuracy: 0.3749 - val_loss: 1.5384 - val_accuracy: 0.4053
Epoch 3/50
448/448 [=====] - 51s 114ms/step - loss: 1.5202 - accuracy: 0.4155 - val_loss: 1.4557 - val_accuracy: 0.4474
Epoch 4/50
448/448 [=====] - 51s 113ms/step - loss: 1.4523 - accuracy: 0.4444 - val_loss: 1.4003 - val_accuracy: 0.4661
Epoch 5/50
448/448 [=====] - 53s 119ms/step - loss: 1.3948 - accuracy: 0.4690 - val_loss: 1.3549 - val_accuracy: 0.4824
Epoch 6/50
448/448 [=====] - 50s 112ms/step - loss: 1.3422 - accuracy: 0.4904 - val_loss: 1.3113 - val_accuracy: 0.4999
Epoch 7/50
448/448 [=====] - 50s 111ms/step - loss: 1.2977 - accuracy: 0.5091 - val_loss: 1.2806 - val_accuracy: 0.5144
Epoch 8/50
448/448 [=====] - 49s 109ms/step - loss: 1.2568 - accuracy: 0.5241 - val_loss: 1.2544 - val_accuracy: 0.5237
```

Figure 24 Training epochs

#### 4.2.6. Saving the model structure & saving trained model

```
# save model structure in json file
model_json = emotion_model.to_json()
with open("emotion_model.json", "w") as json_file:
    json_file.write(model_json)

# save trained model weight in .h5 file
emotion_model.save_weights('emotion_model.h5')
```

Figure 252 Save model structure

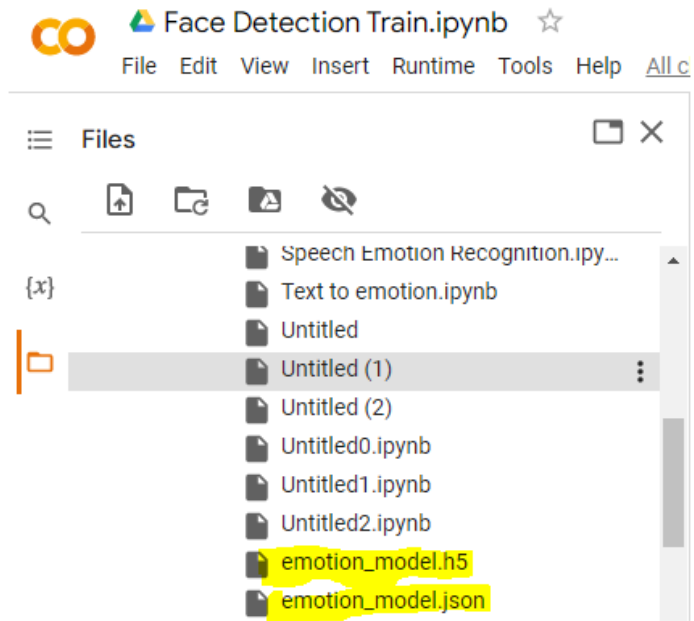


Figure 26 Dataset

## 4.2.8 Model Evaluation

```
import numpy as np
from keras.models import model_from_json
import matplotlib.pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import confusion_matrix, classification_report, ConfusionMatrixDisplay

emotion_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4: "Neutral", 5: "Sad", 6: "Surprised"}

# load json and create model
json_file = open('/content/drive/MyDrive/Colab Notebooks/emotion_modelT1.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
emotion_model = model_from_json(loaded_model_json)

# load weights into new model
emotion_model.load_weights("/content/drive/MyDrive/Colab Notebooks/emotion_modelT1.h5")
print("Loaded model from disk")

# Initialize image data generator with rescaling
test_data_gen = ImageDataGenerator(rescale=1./255)

# Preprocess all test images
test_generator = test_data_gen.flow_from_directory(
    '/content/drive/MyDrive/Facial_Data /Test',
    target_size=(48, 48),
    batch_size=64,
    color_mode="grayscale",
    class_mode='categorical')
```

Figure 3 Model Evaluation

The saved model is loaded to perform evaluation. The test image dataset is normalized and pre-processed before passed through the model

```
# predict on test data
predictions = emotion_model.predict_generator(test_generator)

#display predictions
for result in predictions:
    max_index = int(np.argmax(result))

print(emotion_dict[max_index])
print("-----")
# confusion matrix
c_matrix = confusion_matrix(test_generator.classes, predictions.argmax(axis=1))
print(c_matrix)
cm_display = ConfusionMatrixDisplay(confusion_matrix=c_matrix, display_labels=emotion_dict)
cm_display.plot(cmap=plt.cm.Blues)
plt.show()

# Classification report
print("-----")
print(classification_report(test_generator.classes, predictions.argmax(axis=1)))

Loaded model from disk
Found 7178 images belonging to 7 classes.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:32: UserWarning: `Model.predict_generat
Happy
-----
```

Figure 27 Model Evaluation

Confusion Matrix and Classification report are generated to perform model evaluation.

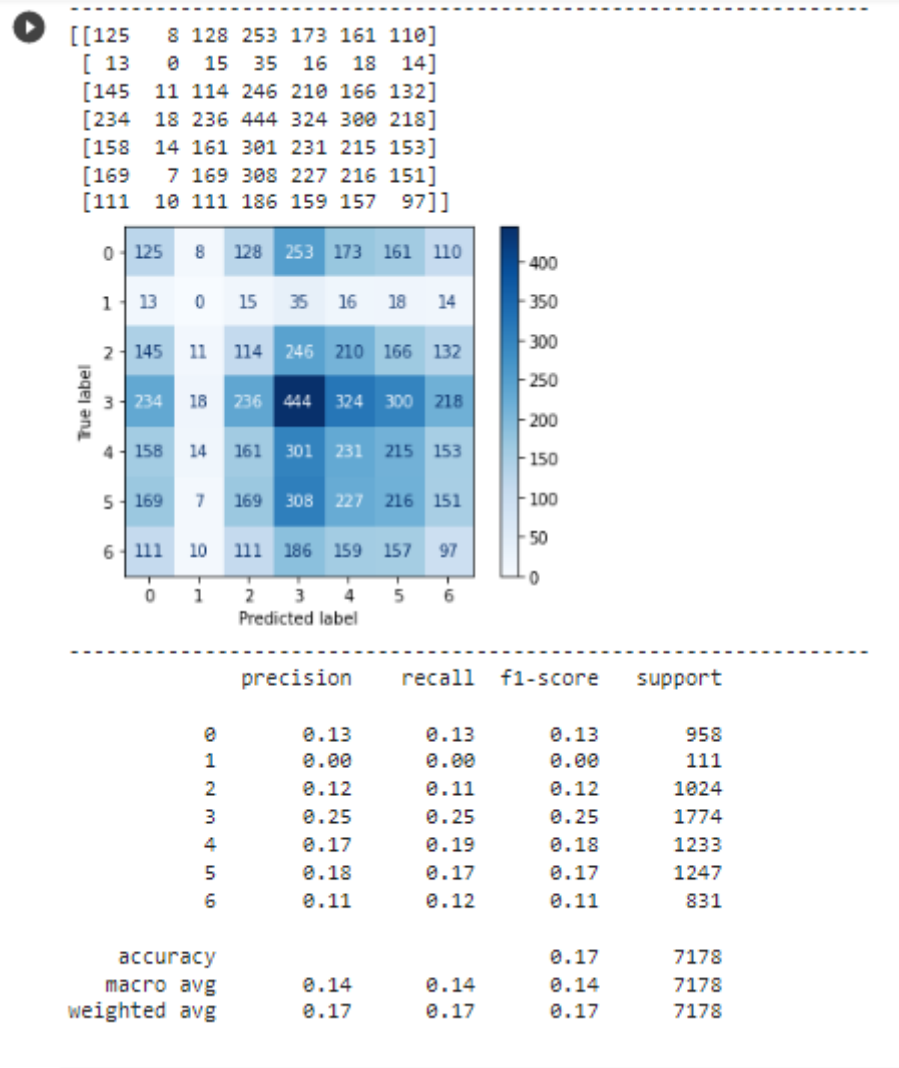


Figure 28 Confusion Matrix and Classification Report

### 4.3. Speech Emotion Recognition Model

The development of Speech Emotion Recognition model includes the below steps.

1. Import libraries and packages
2. Data Preparation
3. Data Augmentation
4. Audio Feature Extraction
5. Model Training
6. Model Evaluation

Four Datasets have been used in this project and are freely available to download from kaggle website. Data is downloaded and is stored in google drive. Below are the detail of dataset.

- a. Ryerson Audio Visual Database of Emotional Speech and Song (Ravdess) dataset description:

Dataset link to download: <https://www.kaggle.com/uwrfkagglers/ravdess-emotional-speech-audio>

Dataset has sub folders and wav file names saved in number format 03-01-01-01-01-01-01.wav.

Actor (01 to 24. Odd numbered actors are male, even numbered actors are female). Numbers can be used as identifiers and can be identified as below:

**Modality** (01 = full-AV, 02 = video-only, 03 = audio-only).

**Vocal channel** (01 = speech, 02 = song).

**Emotion** (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).

**Emotional intensity** (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion.

**Statement Recorded**(01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").

**Repetition** (01 = 1st repetition, 02 = 2nd repetition).



Therefore file 03-01-01-01-01-01-01.wav can be understood as 03=audio-only, 01=speech, 01=neutral, 01=normal, 01=statement kids and 01=1st repetition.

- b. Crowd sourced Emotional Multimodal Actors Dataset (CREMA-D) dataset description:

Dataset link to download: "<https://www.kaggle.com/ejlok1/cremad>"

The format of files is 1001\_DFA\_ANG\_XX.wav, where ANG stands for angry emotion.

Other emotion mappings are as follows:

{'SAD':'sad','ANG':'angry','DIS':'disgust','FEA':'fear','HAP':'happy','NEU':'neutral'}

- c. Toronto emotional speech set (Tess) dataset description:

Dataset link to download: "<https://www.kaggle.com/ejlok1/toronto-emotional-speech-set-tess>"

There are folders in format OAF\_angry, OAF\_neural, OAF\_disgust, YAF\_sad and so on, where name after the underscore of the folder name contains the emotion information, so the name after the underscore of the folder name is taken and files residing inside the folders are labelled accordingly.

- d. Surrey Audio Visual Expressed Emotion (Savee) dataset description:

Dataset link to download: "<https://www.kaggle.com/ejlok1/surrey-audiovisual-expressed-emotion-savee>"

The files are in a format DC\_a01.wav where a single character contains the emotion information, for example character 'a' after underscore in the file name "DC\_a01.wav" means emotion is angry.

Similarly other emotion mappings are as follows:

{'a':'anger','d':'disgust','f':'fear','h':'happiness','n':'neutral','sa':'sadness','su':'surprise'}

### 4.3.1. Import libraries and packages

```
!pip install ffmpeg-python
import pandas as pd
import numpy as np
import tensorflow as tf
import os,time,librosa,warnings,glob
import regex as re
from sklearn.metrics import confusion_matrix,classification_report
import librosa.display
from sklearn.preprocessing import MinMaxScaler,OneHotEncoder
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from keras.layers import Dense,Input,Add,Flatten,Dropout,Activation,AveragePooling1D,Conv1D
from keras.models import Model,Sequential,load_model
from tensorflow.keras.optimizers import Adam
from keras.callbacks import LearningRateScheduler,EarlyStopping,ReduceLROnPlateau,ModelCheckpoint
from google.colab.output import eval_js
from base64 import b64decode
from IPython.display import Audio,HTML
from scipy.io.wavfile import read as wav_read
import io
import ffmpeg
warnings.filterwarnings("ignore")

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: ffmpeg-python in /usr/local/lib/python3.7/dist-packages (0.2.0)
Requirement already satisfied: future in /usr/local/lib/python3.7/dist-packages (from ffmpeg-python) (0.16.0)
```

Figure 29 Libraries and packages

1. FFmpeg is an open-source software project with a suite of libraries and programs for handling video, audio, multimedia files and streams. FFmpeg is a tool itself designed for processing of video and audio files
2. Librosa is powerful Python library built to work with audio and perform analysis on it. It is the starting point towards working with audio data at scale for a wide range of applications such as detecting voice from a person to finding personal characteristics from an audio.
3. Pandas is a library used in Python for data manipulation and analysis
4. Io is a library that deals with input/output; mainly three types – text, binary and raw/I/O
5. Base64 is a library used for encoding binary data and decoding into human readable format
6. Scipy is a library used to wrap highly optimized implementations written in low level languages used for technical computing and scientific computing

7. Regex is a sequence of characters that can be used check if a specific string contains a specified search pattern

```
# Run this cell to mount your Google Drive.
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

Figure 30 Mounting google drive

### 4.3.2. Data Preparation

Functions defined to retrieve 4 dataset and to save them on dataframe

```
#Function to retrieve ravdess dataset from google drive and label them
def ravdess_data():
    #directory of the audio dataset
    ravdess = "/content/drive/MyDrive/Ravdess"
    #label ravdess data
    emotion_ravdess = {'01':'neutral','02':'calm','03':'happy','04':'sad','05':'angry','06':'fearful','07':'disgust','08':'surprised'}
    #list to store ravdess emotion
    ravdess_emotion = []
    #list to store ravdess audio path
    ravdess_path = []
    #get subfolders from the path
    ravdess_folder = os.listdir(ravdess)
    for i in ravdess_folder:
        inner_files = os.listdir(ravdess+i+'/')
        for j in inner_files:
            #get the split part which contains the emotion information then append it into lists
            emotion = j.split('-')[2]
            ravdess_path.append(ravdess+i+'/'+j)
            ravdess_emotion.append(emotion_ravdess[emotion])

    #convert to dataframe
    df_ravdess = pd.DataFrame([ravdess_path,ravdess_emotion]).T
    df_ravdess.columns = ["AudioPath","Label"]
    print("length of ravdess dataset",len(df_ravdess))

    return df_ravdess
```

Figure 31 ravdess

```
#function for getting crema dataset details and labeling
def crema_data():
    #directory of the audio dataset
    crema = "/content/drive/MyDrive/CREMA-D Dataset"
    #label ravdess data
    emotion_crema = {'SAD':'sad','ANG':'angry','DIS':'disgust','FEA':'fear','HAP':'happy','NEU':'neutral'}
    #list to store crema emotion
    crema_emotion = []
    #list to store crema audio path
    crema_path = []
    #get crema files in directory
    crema_files = os.listdir(crema)
    for i in crema_files:
        emotion = i.split('_')[2]
        crema_emotion.append(emotion_crema[emotion])
        crema_path.append(crema+i)

    #convert to dataframe
    df_crema = pd.DataFrame([crema_path,crema_emotion]).T
    df_crema.columns = ["AudioPath","Label"]
    print("length of crema dataset",len(df_crema))

    return df_crema
```

Figure 32 Crema Dataset

```

#function for getting tess dataset and labeling
def tess_data():
    #directory of the audio dataset
    tess = "/content/drive/MyDrive/TESS Toronto emotional speech set data"
    tess_emotion = []
    tess_path = []
    tess_folder = os.listdir(tess)
    for i in tess_folder:
        emotion = i.split('.')[1]
        inner_files = os.listdir(tess+i+'/')
        for j in inner_files:
            tess_path.append(tess+i+'/'+j)
            tess_emotion.append(emotion)

    #convert to dataframe
    df_tess = pd.DataFrame([tess_path,tess_emotion]).T
    df_tess.columns = ["AudioPath","Label"]
    print("length of tess dataset",len(df_tess))

    return df_tess

```

```

#function to get savee dataset and labeling
def savee_data():
    #directory of the audio dataset
    savee = "/content/drive/MyDrive"
    emotion_savee = {'a':'anger','d':'disgust','f':'fear','h':'happiness','n':'neutral','sa':'sadness','su':'surprise'}
    savee_emotion = []
    savee_path = []
    savee_files = os.listdir(savee)
    for i in savee_files:
        emotion = i.split('.')[1]
        emotion = re.match(r"([a-z]+)([0-9]+)",emotion)[1]
        savee_emotion.append(emotion_savee[emotion])
        savee_path.append(savee+i)

    #convert to dataframe
    df_savee = pd.DataFrame([savee_path,savee_emotion]).T
    df_savee.columns = ["AudioPath","Label"]
    print("length of savee dataset",len(df_savee))

    return df_savee

```

Figure 4 Function to retrieve 4 datasets and save as a dataframe

A function `fetch_data()` is defined to fetch all 4 data frames.

```

def fetch_data():
    #get ravdess data
    df_ravdess = ravdess_data()
    #get crema data
    df_crema = crema_data()
    #get tess data
    df_tess = tess_data()
    #get savee data
    df_savee = savee_data()
    #combine all four dataset into one single dataset and create a dataframe
    frames = [df_ravdess,df_crema,df_tess,df_savee]
    final_combined = pd.concat(frames)
    final_combined.reset_index(drop=True,inplace=True)
    #save the information of datasets with their path and labels into a csv file
    final_combined.to_csv("/content/drive/MyDrive/preprocesseddata.csv",index=False,header=True)
    print("Total length of the dataset is {}".format(len(final_combined)))
    return final_combined

```

Figure 5 Functin to fetch dataframes with data info

### 4.3.3. Data Augmentation

```
#below are four data agumentation functions for noise, stretch, shift, pitch
#function to add noise to audio
def noise(data):
    noise_amp = 0.035*np.random.uniform()*np.amax(data)
    data = data + noise_amp*np.random.normal(size=data.shape[0])
    return data

#fuction to strech audio
def stretch(data, rate=0.8):
    return librosa.effects.time_stretch(data, rate)

#fuction to shift audio range
def shift(data):
    shift_range = int(np.random.uniform(low=-5, high = 5)*1000)
    return np.roll(data, shift_range)

#function to change pitch
def pitch(data, sampling_rate, pitch_factor=0.7):
    return librosa.effects.pitch_shift(data, sampling_rate, pitch_factor)
```

Figure 6 Data Augmentation

Data Augmentation is performed to increase the samples of training dataset.

Four Data Augmentation Techniques are used.

1. Add Noise – Adding noise to the dataset will increase the size of the dataset. Random noise is added to the dataset to make each sample different thus reduce overfitting.
2. Stretch Audio , shift audio range and change pitch - Stretching audio by noise injection, changing speed and pitch of the audio and shifting time are a few stretching activities performed on audio dataset which will make audio samples different to each other.

#### 4.3.4. Feature Extraction

```
def extract_features(data,sample_rate):

    #zero crossing rate
    result = np.array([])
    zcr = np.mean(librosa.feature.zero_crossing_rate(y=data).T, axis=0)
    result = np.hstack((result, zcr))
    #print('zcr',result.shape)

    #chroma shift
    stft = np.abs(librosa.stft(data))
    chroma_stft = np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T, axis=0)
    result = np.hstack((result, chroma_stft))
    #print('chroma',result.shape)

    #mfcc
    mfcc = np.mean(librosa.feature.mfcc(y=data, sr=sample_rate).T, axis=0)
    result = np.hstack((result, mfcc))
    #print('mfcc',result.shape)

    #rmse
    rms = np.mean(librosa.feature.rms(y=data).T, axis=0)
    result = np.hstack((result, rms))
    #print('rmse',result.shape)

    #melspectrogram
    mel = np.mean(librosa.feature.melspectrogram(y=data, sr=sample_rate).T, axis=0)
    result = np.hstack((result, mel))
    #print('mel',result.shape)

    #rollof
    rollof = np.mean(librosa.feature.spectral_rolloff(y=data, sr=sample_rate).T, axis=0)
    result = np.hstack((result, rollof))
    #print('rollof',result.shape)
```

Figure 7 Audio Feature Extraction

The features from the audio file are extracted and saved in a csv file

```

] #This function will extract features from each audio file
#extracted audio features with label are stored in a csv file

def Audio_features_extract():
    #this function is used to fetch the data from all the four datasets
    df = fetch_data()
    #count is used to keep a check of number of files processed
    count = 0
    #list to store audio features and their label information
    X_data, Y_label = [], []
    #zip audio path and label information and then iterate over them
    for path, emotion in zip(df["AudioPath"], df["Label"]):
        print("Number of files processed ",count)
        #get the features
        #for one audio file it get three sets of features
        #original features, features with noise(augmentation) and feature with change in stretch and pitch
        #so one audio file generates three output and the label is same for all the outputs
        feature = get_features(path)
        for ele in feature:
            X_data.append(ele)
            Y_label.append(emotion)
        count+=1
    #create a dataframe of audio features
    Features = pd.DataFrame(X_data)
    #add label information
    Features['Label'] = Y_label
    #store the extracted features in a csv file
    Features.to_csv('/content/drive/MyDrive/Audio_features_All_pr.csv',index=False,header=True )

#once the features are extracted then these features are used for making model

```

Figure 8 Extracted features saved into csv

The output of the above function is as below.

```

Number of files processed 12002
zcr (1,)
chroma (13,)
mfcc (33,)
rmse (34,)
mel (162,)
rollof (163,)
centroids (164,)
contrast (171,)
bandwidth (172,)
tonnetz (178,)
zcr (1,)
chroma (13,)
mfcc (33,)
rmse (34,)
mel (162,)
rollof (163,)

```

Figure 9 Output - Audio Feature Extraction

```

#function to plot loss and accuracy curves on training set
def plotgraph(history):
    plt.figure(figsize=[8,6])
    plt.plot(history.history['loss'],'firebrick',linewidth=3.0)
    plt.plot(history.history['accuracy'],'turquoise',linewidth=3.0)
    plt.legend(['Training loss','Training Accuracy'],fontsize=18)
    plt.xlabel('Epochs ',fontsize=16)
    plt.ylabel('Loss and Accuracy',fontsize=16)
    plt.title('Loss Curves and Accuracy Curves',fontsize=16)

#This function performs additional preprocessing and EDA
#The selected emotions are labelled and saved and Emotions are renamed to group into specific classes
def additional_preprocess(filepath):
    #read the csv file of extracted features
    df = pd.read_csv(filepath)
    print("\nEmotions present in dataset\n",df["Label"].unique())
    #replace label names with name common for each emotion
    df["Label"] = df["Label"].str.replace("sadness", "sad", case = True)
    df["Label"] = df["Label"].str.replace("happiness", "happy", case = True)
    df["Label"] = df["Label"].str.replace("Fear", "fear", case = True)
    df["Label"] = df["Label"].str.replace("Sad", "sad", case = True)
    df["Label"] = df["Label"].str.replace("Pleasant_surprise", "surprise", case = True)
    df["Label"] = df["Label"].str.replace("pleasant_surprised", "surprise", case = True)
    df["Label"] = df["Label"].str.replace("surprised", "surprise", case = True)
    df["Label"] = df["Label"].str.replace("fearful", "fear", case = True)
    df["Label"] = df["Label"].str.replace("anger", "angry", case = True)
    #drop labels surprised and calm
    #these label dosent contain sufficient amount of data and may lead to missclassification
    print("\nUnique count of labels or emotions\n",df["Label"].value_counts())
    #drop labels or emotions which can lead to misclassification
    df.drop((np.where(df['Label'].isin(["surprise","calm"])))[0]), inplace = True)
    print("\nUnique count of labels or emotions after dropping selected labels\n",df["Label"].value_counts())
    print("\nlength of the total data is {}".format(len(df)))
    return df

```

Figure 10 Loss and accuracy plot, additional data pre-processing



```

C>
Emotions present in dataset
['calm' 'neutral' 'sad' 'happy' 'angry' 'disgust' 'fearful' 'surprised'
'fear' 'pleasant_surprised' 'Sad' 'Fear' 'Pleasant_surprise' 'anger'
'happiness' 'surprise' 'sadness']

Unique count of labels or emotions
sad      5769
happy    5769
angry    5769
disgust  5769
fear     5769
neutral  5109
surprise 1956
calm     576
Name: Label, dtype: int64

Unique count of labels or emotions after dropping selected labels
sad      5769
happy    5769
angry    5769
disgust  5769
fear     5769
neutral  5109
Name: Label, dtype: int64

length of the total data is 33954

length of train data is 27163, test data is 3395 and validation set is 3396

shape of train features and label is (27163, 178)

shape of test features and label is (3395, 178)

shape of validation features and label is (3396, 178)
Model: "model_1"

```

The above code snippet plots the loss and accuracy curve. Also perform additional pre-processing of data. Also, the classes surprised, and calm are dropped as they does not contain sufficient data samples.

```

#this function is used to get audio features perform one hot encoding and split datasets into train, test and validation
def audio_features_final():
    df = additional_preprocess("/content/drive/MyDrive/Audio_features_All_pr.csv")
    #get all the audio features as numpy array from the dataframe
    #last column is label so last column is not fetched only 0 to:-1
    data=df[df.columns[0:-1]].values
    #perform one hot encoding on labels
    encoder = OneHotEncoder()
    #fetch the last column of labels and perform one hot encoding on them
    label=df["Label"].values
    label = encoder.fit_transform(np.array(label).reshape(-1,1)).toarray()
    #min max scaler is used to normalize the data
    scaler = MinMaxScaler()
    data=scaler.fit_transform(data)
    #split the dataframe into train and test 80% train, 10% validation and 10% test datasets
    x_train, x_test, y_train, y_test = train_test_split(data, label, test_size=0.20, random_state=42,shuffle=True)
    x_test, x_val, y_test, y_val = train_test_split(x_test, y_test, test_size=0.50, random_state=42, shuffle=True)
    print("\nlength of train data is {}, test data is {} and validation set is {}".format(len(x_train),len(x_test),len(x_val)))
    print("\n shape of train features and label is {}".format(x_train.shape, y_train.shape))
    print("\n shape of test features and label is {}".format(x_test.shape, y_test.shape))
    print("\n shape of validation features and label is {}".format(x_val.shape,y_val.shape))
    return x_train, x_test, y_train, y_test, x_val, y_val, encoder

```

Figure 11 One Hot Encoding and MinMaxScaler

In the above code snippet, One Hot Encoding is performed to label the fetched data

Data frame is split into train and test dataset in 80:10:10 ratio

```
#reduce the learning rate if plateau is encountered
reduce_lr = ReduceLRonPlateau(monitor='loss', factor=0.2, patience=5, min_lr=0.001)
#early stopping method is used to monitor the loss if there are no significant reductions in loss then halt the training
es = EarlyStopping(monitor='loss', patience=20)
#checkpoint to save the best model with highest validation accuracy
filepath = "/content/drive/MyDrive/Speech-emotion-recognition.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
#create a combined list of reduce learning rate, early stopping and checkpoint
callbacks_list = [reduce_lr, es, checkpoint]
def residual_block(x, filters, conv_num=3, activation="relu"):
    #function to create residual blocks and add residual blocks
    s = Conv1D(filters, 1, padding="same")(x)
    for i in range(conv_num - 1):
        x = Conv1D(filters, 3, padding="same")(x)
        x = Activation(activation)(x)
    x = Conv1D(filters, 3, padding="same")(x)
    x = Add()([x, s])
    x = Activation(activation)(x)
    return x
#function to build the model
def build_model():
    inputs = Input(shape=(x_train.shape[1],1))
    x = Dense(256, activation="relu")(inputs)
    x = residual_block(x, 16, 2)
    x = residual_block(x, 32, 2)
    x = residual_block(x, 32, 2)
    x = residual_block(x, 64, 3)
    x = residual_block(x, 64, 3)
    x = residual_block(x, 128, 3)
    x = residual_block(x, 128, 3)
    #perform the average pooling after last residual block
    x = AveragePooling1D(pool_size=3, strides=3)(x)
    x = Flatten()(x)
    x = Dense(256, activation="relu")(x)
    x = Dense(128, activation="relu")(x)
    outputs = Dense(6, activation="softmax", name="output")(x)
    return Model(inputs=inputs, outputs=outputs)

res_model = build_model()
#display the summary of the model
res_model.summary()
#compile the model
res_model.compile(optimizer=Adam(lr=1e-4, decay=1e-4 / 50), loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True), metrics=['accuracy'])
res_model.compile(loss='categorical_crossentropy', optimizer = Adam(lr=1e-4, decay=1e-4 / 50) , metrics=['accuracy'])
history = res_model.fit(np.expand_dims(x_train,-1), y_train,
```

Figure 12 Model Training

### 4.3.5. Model Training

There are a few differences between Facial Emotion Recognition and Speech Emotion Recognition models although both are Convolutional Neural Networks.

In the Speech Emotion Recognition model, a few parameters are added which are mentioned below.

1. ReduceLRonPlateau class from Keras is used to reduce learning rate when a metric has stopped improving.
2. ModelCheckpoint is a Keras call-back to save model weights or entire model at a specific frequency or whenever a quantity (for example, training loss) is optimum when compared to last epoch/batch. ModelCheckpoint captures the weights of the model or entire model during training.

### 4.3.6. Model Evaluation

```

Epoch 49: val_accuracy did not improve from 0.73675
849/849 [=====] - 333s 393ms/step - loss: 0.0704 - accuracy: 0.9777 - val_loss: 1.9251 - val_accuracy: 0.7362 - lr: 1.0000e-04
Epoch 50/50
849/849 [=====] - ETA: 0s - loss: 0.0673 - accuracy: 0.9800
Epoch 50: val_accuracy did not improve from 0.73675
849/849 [=====] - 332s 391ms/step - loss: 0.0673 - accuracy: 0.9800 - val_loss: 1.9089 - val_accuracy: 0.7329 - lr: 1.0000e-04
27163/27163 [=====] - 253s 9ms/step - loss: 0.0513 - accuracy: 0.9840
3395/3395 [=====] - 32s 9ms/step - loss: 1.7653 - accuracy: 0.7426
3396/3396 [=====] - 39s 12ms/step - loss: 1.7134 - accuracy: 0.7367

*****

Training accuracy of the model is 98.4

Testing accuracy of the model is 74.26

Validation accuracy of the model is 73.67
*****

Classification report for Emotion Recognition
      precision    recall  f1-score   support

     0       0.81     0.88     0.84     577
     1       0.71     0.71     0.71     583
     2       0.80     0.64     0.71     573
     3       0.70     0.73     0.72     583
     4       0.73     0.75     0.74     519
     5       0.71     0.75     0.73     560

 accuracy         0.74
 macro avg        0.74
 weighted avg     0.74

Confusion matrix for Emotion Recognition
[[507  16  10  36  6  2]
 [ 31 413  19  39  41 40]
 [ 25  42 366  46  22 72]
 [ 59  36  27 427  21 13]
 [  4  41  9  32 390 43]
 [  2  36  25  26  53 418]]

```

Figure 13SER Model Evaluation

## 4.4. Text to Emotion

Text2Emotion is a python package that will assist you to pull out emotions from the content. Processes any textual data, recognizes the emotion embedded in it, and provides the output in the form of a dictionary. Well suited with 5 basic emotion categories such as Happy, Angry, Sad, Surprise, and Fear.

Natural Language Toolkit is also downloaded along with text2emotion package which would provide natural language processing capabilities by providing easy to use interfaces to over 50 lexical resources. In addition, it also provides a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing and semantic reasoning.

```
!pip install text2emotion

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: text2emotion in /usr/local/lib/python3.7/dist-packages (0.0.5)
Requirement already satisfied: emoji>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from text2emotion) (1.7.0)
Requirement already satisfied: nltk in /usr/local/lib/python3.7/dist-packages (from text2emotion) (3.7)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages (from nltk->text2emotion) (7.1.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from nltk->text2emotion) (4.64.0)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from nltk->text2emotion) (1.1.0)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.7/dist-packages (from nltk->text2emotion) (2022.6.2)

[ ] # Run this cell to mount your Google Drive.
from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[ ] import text2emotion as te
import nltk
nltk.download('omw-1.4')

[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data] Package omw-1.4 is already up-to-date!
True

[ ] text = "I am good"
te.get_emotion(text)

{'Happy': 0, 'Angry': 0, 'Surprise': 0, 'Sad': 0, 'Fear': 0}

import text2emotion as te

text = []
with open("/content/drive/MyDrive/testtext.txt", 'r') as r:
    text = r.read()
    print(text)

te.get_emotion(text)

I'm not bad but I'm upset
```

Figure 14 Text2emotion package

## 4.5. Audio-Video/Audio Input and File Conversion

### 4.5.1 Audio Input from Microphone

Since Google colab is unable to access microphone or webcam to get input from the user, JavaScript code is used to enable google colab access audio video sources from external hardware.

```
#this javascript is used to tell colab cell to open microphone and record audio
AUDIO_HTML = ""
<script>
var my_div = document.createElement("DIV");
var my_p = document.createElement("P");
var my_btn = document.createElement("BUTTON");
var t = document.createTextNode("Press to start recording");

my_btn.appendChild(t);
//my_p.appendChild(my_btn);
my_div.appendChild(my_btn);
document.body.appendChild(my_div);

var base64data = 0;
var reader;
var recorder, gumStream;
var recordButton = my_btn;

var handleSuccess = function(stream) {
  gumStream = stream;
  var options = {
    //bitsPerSecond: 8000, //chrome seems to ignore, always 48k
    mimeType : 'audio/webm;codecs=opus'
    //mimeType : 'audio/webm;codecs=pcm'
  };
  //recorder = new MediaRecorder(stream, options);
  recorder = new MediaRecorder(stream);
  recorder.ondataavailable = function(e) {
    var url = URL.createObjectURL(e.data);
    var preview = document.createElement('audio');
    preview.controls = true;
    preview.src = url;
    document.body.appendChild(preview);

    reader = new FileReader();
    reader.readAsDataURL(e.data);
    reader.onloadend = function() {
      base64data = reader.result;
      //console.log("Inside FileReader:" + base64data);
    }
  };
  recorder.start();
};
```

Figure 15 JavaScript code to access Microphone

```

#Javascript code to access webcam
def record_video(filename):
    js=Javascript("""
    async function recordVideo() {
        const options = { mimeType: "video/webm; codecs=vp9" };
        const div = document.createElement('div');
        const capture = document.createElement('button');
        const stopCapture = document.createElement("button");

        capture.textContent = "Start Recording";
        capture.style.background = "orange";
        capture.style.color = "white";

        stopCapture.textContent = "Stop Recording";
        stopCapture.style.background = "red";
        stopCapture.style.color = "white";
        div.appendChild(capture);

        const video = document.createElement('video');
        const recordingVid = document.createElement("video");
        video.style.display = 'block';

        const stream = await navigator.mediaDevices.getUserMedia({audio:true, video: true});

        let recorder = new MediaRecorder(stream, options);
        document.body.appendChild(div);
        div.appendChild(video);

        video.srcObject = stream;
        video.muted = true;

        await video.play();

        google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

        await new Promise((resolve) => {
            capture.onclick = resolve;
        });
        recorder.start();
        capture.replaceWith(stopCapture);

        await new Promise((resolve) => stopCapture.onclick = resolve);
        recorder.stop();
        let recData = await new Promise((resolve) => recorder.ondataavailable = resolve);
    }
    """);

```

Figure 16 JavaScript code to access webcam

```

from IPython.display import HTML
from base64 import b64encode

def show_video(video_path, video_width = 600):

    video_file = open(video_path, "r+b").read()

    video_url = f"data:video/mp4;base64,{b64encode(video_file).decode()}"
    return HTML(f"<video width={video_width} controls><source src='{video_url}'></video>")

[ ] video_path = "test.mp4"
record_video(video_path)

Finished recording video at:test.mp4

[ ] show_video(video_path)

```

Figure 17 Video File creation to the designated folder

```

#Extracting audio from video
import os, sys, re

video_file_path = "/content/drive/MyDrive/test.mp4" #@param (type:"string")
output_file_extension = "wav" #@param ["m4a", "mp3", "opus", "flac", "wav"]

delsplit = re.search("\\(?:\\.|)+$", video_file_path)
output_file_path = re.search("(\\.|.+)", video_file_path)
filename = re.sub("\\[\\]", "", delsplit.group(0))
filename_raw = re.sub("\\{\\}$", "", filename)

os.environ["inputFile"] = video_file_path
os.environ["outputPath"] = output_file_path.group(0)
os.environ["fileName"] = filename_raw
os.environ["fileType"] = output_file_extension

!ffmpeg -hide_banner -i "$inputFile" -q:a 0 -map a "$outputPath"/"$fileName"-audio."$fileType"

Input #0, matroska,webm, from '/content/drive/MyDrive/test.mp4':
Metadata:
  encoder      : Chrome
Duration: N/A, start: 0.000000, bitrate: N/A
Stream #0:0(eng): Audio: opus, 48000 Hz, mono, fltp (default)
Stream #0:1(eng): Video: vp9 (Profile 0), yuv420p(tv), 640x480, SAR 1:1 DAR 4:3, 1k tbr, 1k tbn, 1k tbc (default)
Metadata:
  alpha_mode   : 1
File '/content/drive/MyDrive/test-audio.wav' already exists. Overwrite ? [y/N] N
Not overwriting - exiting

```

Figure 18 Function to transcribe audio from video file with interface

```

[ ] import shutil
shutil.copy("/content/test-audio.wav", "/content/drive/MyDrive")

'/content/drive/MyDrive/test-audio.wav'

```

Figure 19 Function to mount transcribed audio file to google drive

```
!pip3 install SpeechRecognition
!apt install libasound2-dev portaudio19-dev libportaudio2 libportaudiocpp0 ffmpeg
!pip install PyAudio
import speech_recognition as sr
r = sr.Recognizer()
with sr.AudioFile('/content/drive/MyDrive/test-audio.wav') as source:
    audio = r.listen(source)
    try:
        text = r.recognize_google(audio)
        print('Work in progress...')
        print(text)
    except:
        print("Try Again, please..")
with open('/content/drive/MyDrive/testtext.txt', 'w') as f:
    f.write(text)
```

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>  
Requirement already satisfied: SpeechRecognition in /usr/local/lib/python3.7/dist-packages (3.8.1)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
libportaudio2 is already the newest version (19.6.0-1).  
libportaudiocpp0 is already the newest version (19.6.0-1).  
portaudio19-dev is already the newest version (19.6.0-1).  
libasound2-dev is already the newest version (1.1.3-5ubuntu0.6).  
ffmpeg is already the newest version (7:3.4.11-0ubuntu0.1).  
The following package was automatically installed and is no longer required:  
  libnvidia-common-460  
Use 'apt autoremove' to remove it.  
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.  
Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>  
Requirement already satisfied: PyAudio in /usr/local/lib/python3.7/dist-packages (0.2.12)  
Work in progress...  
I'm not bad but I'm upset

Figure 20 SpeechRecognition module to convert audio file to text file



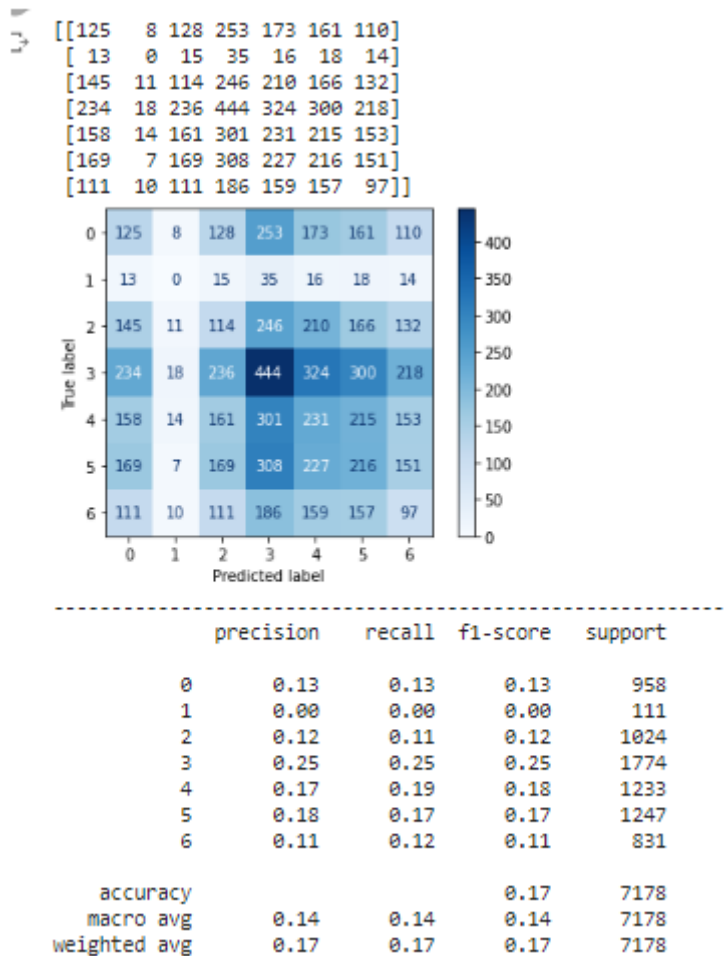
## 5. Results

	Emotion	Facial Emotion Recognition CNN Model	Speech Audio Recognition CNN Model	
	0 Angry	0.13	0.84	F1 Score
	1 Disgust	0.00	0.71	F1 Score
	2 Fear	0.12	0.71	F1 Score
	3 Happy	0.25	0.72	F1 Score
	4 Neutral	0.18	0.74	F1 Score
	5 Sad	0.17	0.73	F1 Score
	6 Surprised	0.11	Dropped due to low training data set	F1 Score

## 6. Discussion

### 6.1 Facial Emotion Recognition Model

As per the confusion matrix, the correlation between True Label and Predicted Label for the emotion Disgust is 0. This could be because the training dataset was too less for this emotion.



### 5.2. Speech Emotion Recognition Model

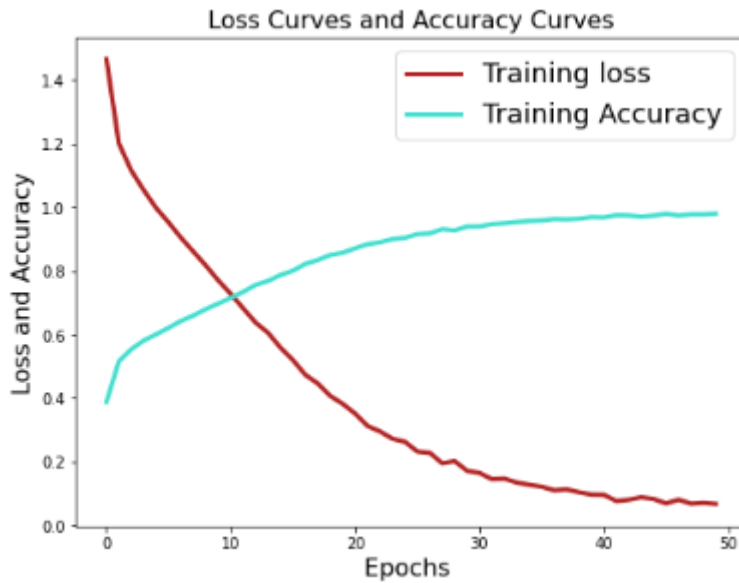
As per the classification report, the F1 score for all emotion classes are above 0.5 and this is a good sign as it shows the model has predicted almost accurately. This could be because the dataset was large.

Training accuracy of the model is 98.4  
 Testing accuracy of the model is 74.26  
 Validation accuracy of the model is 73.67  
 \*\*\*\*\*

Classification report for Emotion Recognition

	precision	recall	f1-score	support
0	0.81	0.88	0.84	577
1	0.71	0.71	0.71	583
2	0.80	0.64	0.71	573
3	0.70	0.73	0.72	583
4	0.73	0.75	0.74	519
5	0.71	0.75	0.73	560
accuracy			0.74	3395
macro avg	0.74	0.74	0.74	3395
weighted avg	0.74	0.74	0.74	3395

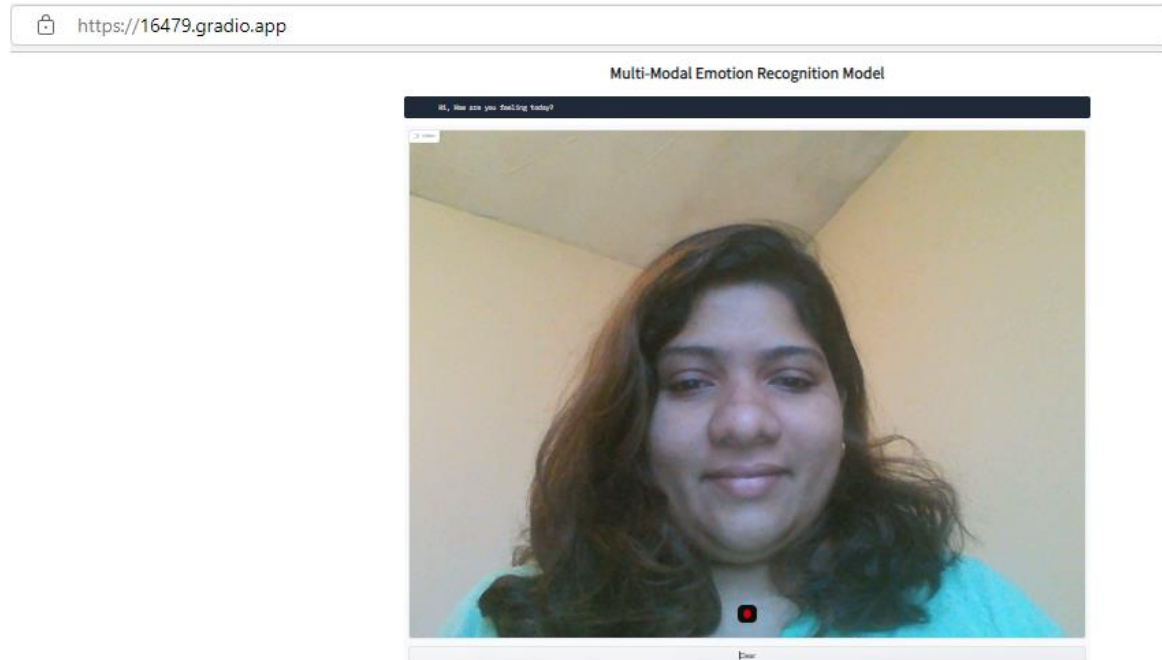
Confusion matrix for Emotion Recognition  
 [[507 16 10 36 6 2]  
 [ 31 413 19 39 41 40]  
 [ 25 42 366 46 22 72]  
 [ 59 36 27 427 21 13]  
 [ 4 41 9 32 390 43]  
 [ 2 36 25 26 53 418]]  
 \*\*\*\*\*



From the above loss and accuracy curve, it is evident that after 30 epochs, there is no variation in accuracy and loss.

## User Testing and Results.

A simple gradio based GUI was developed and is able to take input from webcam.

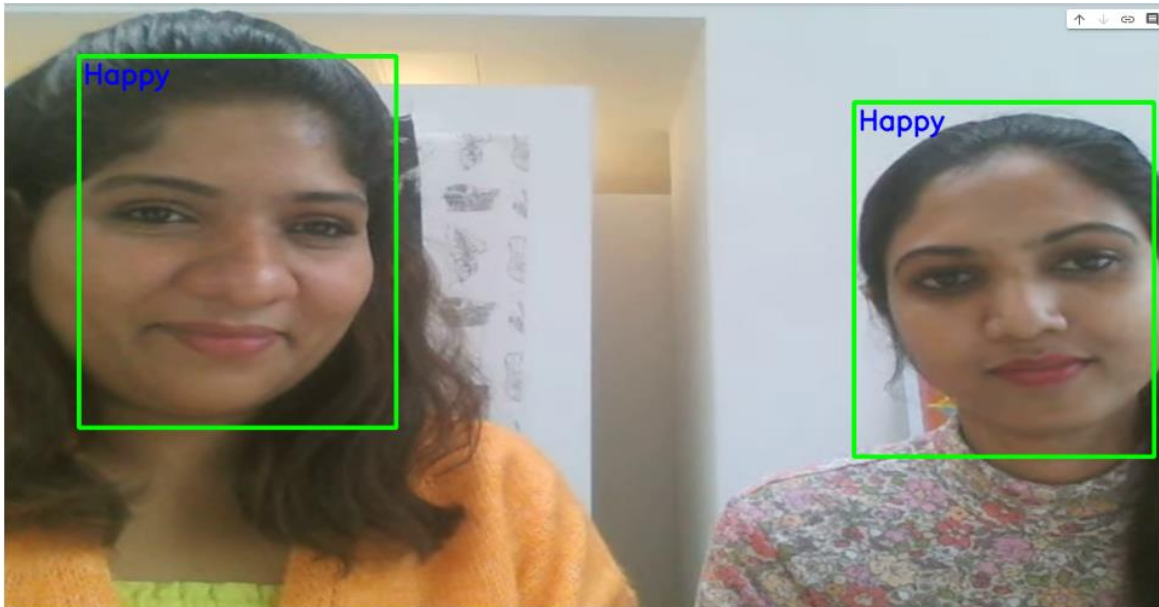
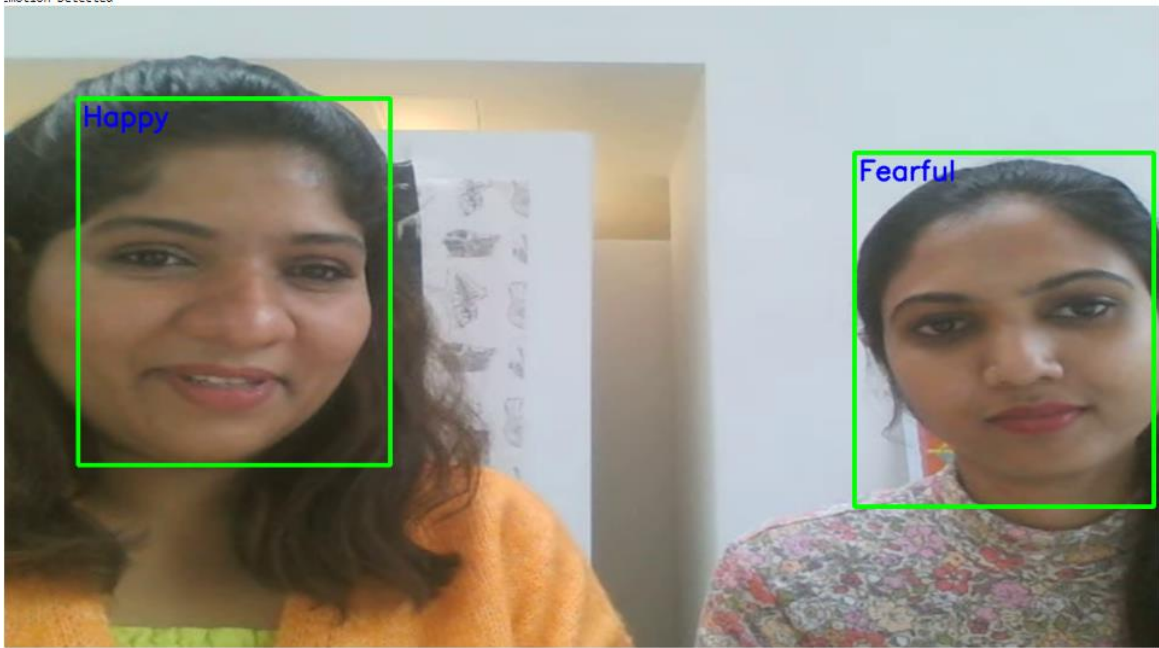


*Figure 21 Gradio GUI APP*

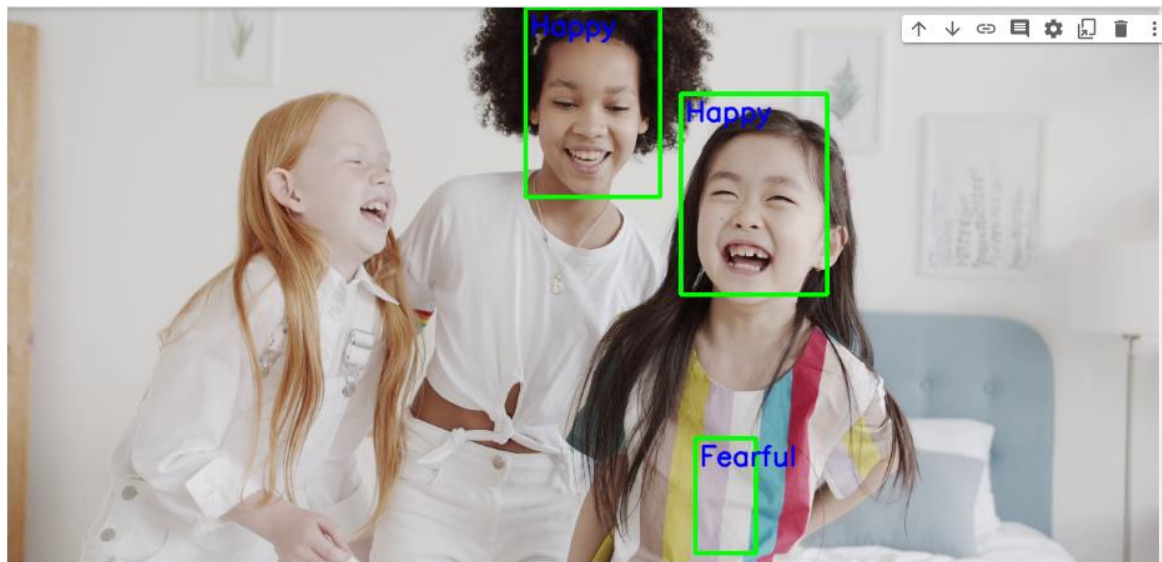
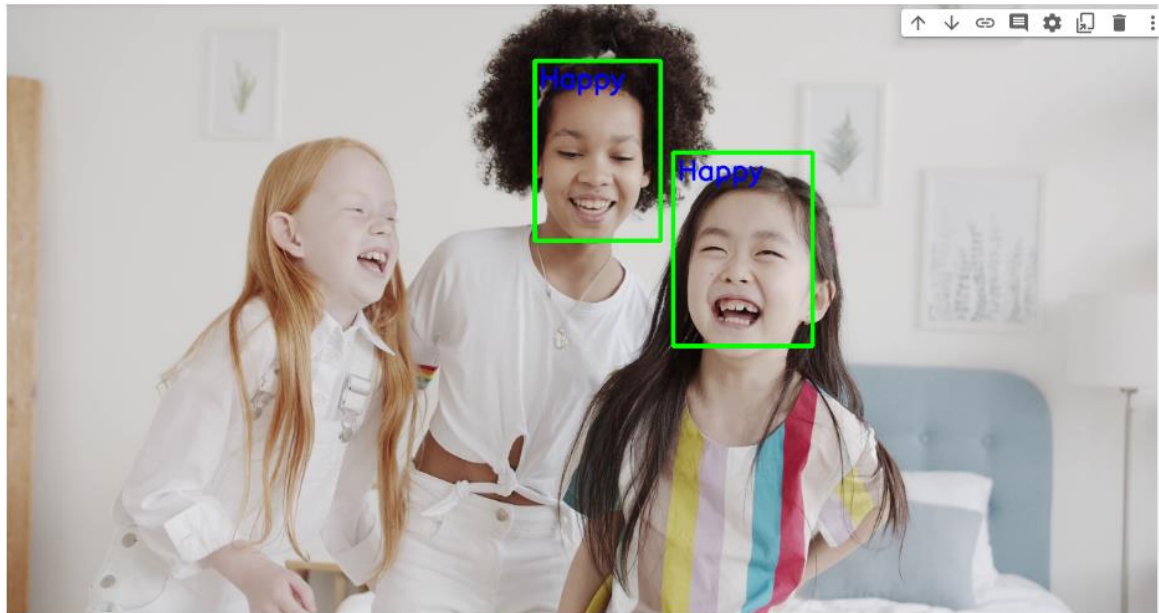
Facial Emotion Recognition system performed well below when

1. Live video was uploaded from webcam
2. Multiple faces are recognized by the model

Emotion Detected



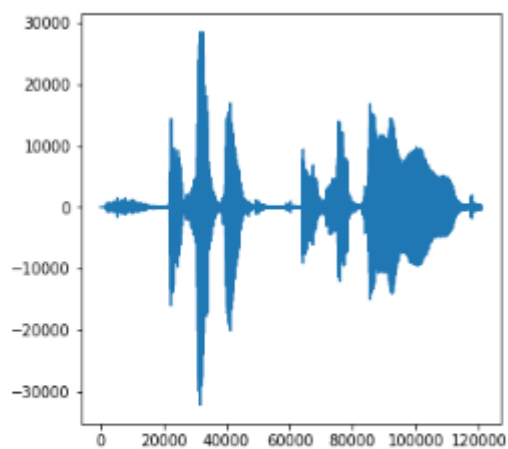
3. Multiple faces were recognized from video however, there was a mistake in identifying face.



```
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 37ms/step  
3uffered data was truncated after reaching the output size limit.
```

## Speech Emotion Recognition system accurately identify live audio recorder

```
[ ] test_realtime(encoder)
```



WARNING:tensorflow:5 out of the last 5 calls to <function Model.make\_pre

The Emotion Predicted For Recorded Audio Using Microphone is ['happy']

<Figure size 432x288 with 0 Axes>

Recording... press to stop

## 7. Conclusion

This paper describes in detail the use of Artificial Intelligence algorithm to perform emotion recognition from facial expressions using deep learning convolutional neural network architectures to detect, identify, extract, and evaluate facial features from images and live web camera. Computer vision is a field of Artificial Intelligence which allows computers to obtain significant knowledge from visual inputs. Deep learning algorithm helps to learn by itself by looking at the labelled images or video. Same concept is applied to Speech signals which performs audio processing and feature extraction.

The convolutional neural network has multiple layers each performing various transforming function to accurately predict the underlying emotion from the facial expression and audio signal. The 7 fundamental human facial expressions are used in this classification problem. A live webcam input converted to audio and text at the same time is passed through Facial and Speech Emotion Recognition system at the same time. However, text to emotion is not achieved through machine learning algorithm. Nevertheless, the training and validation accuracy is 98.4% and 73.67% for Speech Emotion Recognition system and 71.78% for Facial Emotion Recognition System.



## 8. Limitations & Future Works

### I. Limitations:

1. Accuracy of Facial Emotion Recognition system is very low compared to the Speech Emotion Recognition and this could be due to the lack of training dataset for emotion classes.
2. The desired multi-modal emotion recognition system was to develop a combination of RNN and CNN model. However, this could not be achieved due to lack of time.
3. The current study did not include comparative study with other conventional and pre-trained models to accurately measure performance.

### II. Future Works:

1. Future additions to the system can be to include interfaces with other third party applications like meditation app, social media, mental health support network groups, Drinking water reminder application, activity monitoring application to make sure that the individual is active throughout the day, charity/donation applications for the individual to be connected with people who are in need of financial or other support to empower the stressed individuals and to motivate them to be part of charitable work. These are all the pointers mentioned on the NHS website to manage stress effectively
2. Develop a multi-modal emotion recognition model with exceptional performance achieved by combining capabilities of RNN and CNN model. Also to create own dataset for the purpose of training.

## 9. Reference list / Bibliography

- [1] Kosti, Ronak, Jose M. Alvarez, Adria Recasens, and Agata Lapedriza. "Context based emotion recognition using emotic dataset." *IEEE transactions on pattern analysis and machine intelligence* 42, no. 11 (2019): 2755-2766
- [2] Kosti, Ronak, Jose M. Alvarez, Adria Recasens, and Agata Lapedriza. "Emotion recognition in context." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1667-1675. 2017.[3] Kosti, Ronak, Jose M. Alvarez, Adria Recasens, and Agata Lapedriza. "EMOTIC: Emotions in Context dataset." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 61-69. 2017.
- Bilakhiya, Karan. "Text2emotion: Uncovering Human Emotions from Textual Data in Python." *Analytics Vidhya*, 7 Sept. 2020, [medium.com/analytics-vidhya/text2emotion-uncovering-human-emotions-from-textual-data-in-python-cde808995707](https://medium.com/analytics-vidhya/text2emotion-uncovering-human-emotions-from-textual-data-in-python-cde808995707). Accessed 23 Aug. 2022.
- Bouhlal, M., et al. "Emotions Recognition as Innovative Tool for Improving Students' Performance and Learning Approaches." *Procedia Computer Science*, vol. 175, no. ISSN 1877-0509, 2020, pp. 597–602, [10.1016/j.procs.2020.07.086](https://doi.org/10.1016/j.procs.2020.07.086). Accessed 5 Dec. 2020.
- Brownlee, Jason. "Introduction to Python Deep Learning with Keras." *Machine Learning Mastery*, 9 May 2016, [machinelearningmastery.com/introduction-python-deep-learning-library-keras/](https://machinelearningmastery.com/introduction-python-deep-learning-library-keras/).
- Busso, Carlos, et al. *Analysis of Emotion Recognition Using Facial Expressions, Speech and Multimodal Information*. Proceedings of the 6th International Conference on

- Multimodal Interfaces, ICMI 2004, State College, PA, USA, October 13-15, 2004, Jan. 2004.
- Cao, H., et al. “CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset.” *IEEE Transactions on Affective Computing*, vol. 5, no. 4, 1 Oct. 2014, pp. 377–390, [ieeexplore.ieee.org/document/6849440](http://ieeexplore.ieee.org/document/6849440), [10.1109/TAFFC.2014.2336244](https://doi.org/10.1109/TAFFC.2014.2336244). Accessed 2 Apr. 2021.
- Chang, Xin, and Władysław Skarbek. “Multi-Modal Residual Perceptron Network for Audio–Video Emotion Recognition.” *Sensors*, vol. 21, no. 16, 12 Aug. 2021, p. 5452, [10.3390/s21165452](https://doi.org/10.3390/s21165452). Accessed 7 Oct. 2021.
- Cherry, Kendra. “How Many Human Emotions Are There?” *Verywell Mind*, Verywellmind, 26 Feb. 2015, [www.verywellmind.com/how-many-emotions-are-there-2795179](http://www.verywellmind.com/how-many-emotions-are-there-2795179).
- Deng, Jun, et al. “Semisupervised Autoencoders for Speech Emotion Recognition.” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, Jan. 2018, pp. 31–43, [10.1109/taslp.2017.2759338](https://doi.org/10.1109/taslp.2017.2759338).
- Dobrišek, Simon, et al. “Towards Efficient Multi-Modal Emotion Recognition.” *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, Jan. 2013, p. 53, [10.5772/54002](https://doi.org/10.5772/54002). Accessed 21 July 2019.
- Ebrahimi Kahou, Samira, et al. “Recurrent Neural Networks for Emotion Recognition in Video.” *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction - ICMI '15*, 2015, [10.1145/2818346.2830596](https://doi.org/10.1145/2818346.2830596).
- Editor, Michael Kwan. “Effective Problem Statement Examples.” *Examples.yourdictionary.com*, [examples.yourdictionary.com/problem-statement-examples.html](http://examples.yourdictionary.com/problem-statement-examples.html).

- Ekman, Paul, and Wallace V Friesen. “Measuring Facial Movement.” *Measuring Facial Movement*, 1976.
- Ekman, Paul. “Basic Emotions.” *Handbook of Cognition and Emotion*, no. 10.1002/0470013494.ch3, 28 Jan. 2005, pp. 45–60, [onlinelibrary.wiley.com/doi/10.1002/0470013494.ch3?eventSubType=NONE&eventType=FULLTEXT\\_PDF\\_REG](https://onlinelibrary.wiley.com/doi/10.1002/0470013494.ch3?eventSubType=NONE&eventType=FULLTEXT_PDF_REG), [10.1002/0470013494.ch3](https://doi.org/10.1002/0470013494.ch3).
- . “Universal Facial Expression of Emotions.” *Universal Facial Expressions of Emotion*, vol. 8, no. 4, 1970. *California Mental Health Digest*, Autumn 1970.
- Gupta, Aarohi. “Emotion Detection: A Machine Learning Project.” *Medium*, 29 Dec. 2019, [towardsdatascience.com/emotion-detection-a-machine-learning-project-f7431f652b1f](https://towardsdatascience.com/emotion-detection-a-machine-learning-project-f7431f652b1f).
- Gupta, Rohan. “The Intuition behind Facial Detection: The Viola-Jones Algorithm.” *Medium*, 12 Feb. 2020, [towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algorithm-29d9106b6999](https://towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algorithm-29d9106b6999).
- Guy, The AI. “Colab-Webcam.” *GitHub*, 19 July 2022, [github.com/theAIGuysCode/colab-webcam/blob/main/colab\\_webcam.ipynb](https://github.com/theAIGuysCode/colab-webcam/blob/main/colab_webcam.ipynb).
- Hassan, Mohammad Mehedi, et al. “Human Emotion Recognition Using Deep Belief Network Architecture.” *Information Fusion*, vol. 51, Nov. 2019, pp. 10–18, [10.1016/j.inffus.2018.10.009](https://doi.org/10.1016/j.inffus.2018.10.009). Accessed 16 Aug. 2020.
- Hess, Ursula, and Pascal Thibault. “Darwin and Emotion Expression.” *American Psychologist*, vol. 64, no. 2, 2009, pp. 120–128, [10.1037/a0013386](https://doi.org/10.1037/a0013386). Accessed 9 Feb. 2020.
- Hossain, M. Shamim, and Ghulam Muhammad. “Emotion Recognition Using Deep Learning Approach from Audio–Visual Emotional Big Data.” *Information Fusion*,

- vol. 49, Sept. 2019, pp. 69–78, [10.1016/j.inffus.2018.09.008](https://doi.org/10.1016/j.inffus.2018.09.008). Accessed 29 July 2020.
- <https://www.facebook.com/verywell>. “5 Tips to Better Understand Facial Expressions.” *Verywell Mind*, 2019, [www.verywellmind.com/understanding-emotions-through-facial-expressions-3024851](https://www.verywellmind.com/understanding-emotions-through-facial-expressions-3024851).
- IBM. “Computer Vision.” *Ibm.com*, 2019, [www.ibm.com/topics/computer-vision](https://www.ibm.com/topics/computer-vision). *Deep Learning for Emotion Recognition on Small Datasets Using Transfer Learning*. Nov. 2015.
- Jackson, Philip. “Multimodal Emotion Recognition.” *Principles, Algorithms and Systems*, 1 Jan. 2011, [www.academia.edu/11213368/Multimodal\\_Emotion\\_Recognition](https://www.academia.edu/11213368/Multimodal_Emotion_Recognition). Accessed 7 Sept. 2022.
- J, Vijayabhaskar. “Tutorial on Using Keras Flow\_from\_directory and Generators.” *Medium*, 2 Dec. 2019, [vijayabhaskar96.medium.com/tutorial-image-classification-with-keras-flow-from-directory-and-generators-95f75ebe5720](https://vijayabhaskar96.medium.com/tutorial-image-classification-with-keras-flow-from-directory-and-generators-95f75ebe5720).
- Kang & Atul. “ImageDataGenerator – Standardize Method.” *TheAILearner*, 6 July 2019, [theailearner.com/2019/07/06/imagadatagenerator-standardize-method/](https://theailearner.com/2019/07/06/imagadatagenerator-standardize-method/). Accessed 9 Sept. 2022.
- Kazawa, Sunny. “Emotion Detection Using CNN | Emotion Detection Deep Learning Project | Machine Learning | Data Magic.” *Www.youtube.com*, 17 June 2021, [www.youtube.com/watch?v=UHdRxHPRBng](https://www.youtube.com/watch?v=UHdRxHPRBng). Accessed 28 July 2022.
- Khan, Tanwir. “Computer Vision — Detecting Objects Using Haar Cascade Classifier.” *Medium*, 19 Dec. 2019, [towardsdatascience.com/computer-vision-detecting-objects-using-haar-cascade-classifier-4585472829a9](https://towardsdatascience.com/computer-vision-detecting-objects-using-haar-cascade-classifier-4585472829a9).
- Kosaka, Muriel. “Speech Emotion Recognition Using RAVDESS Audio Dataset.”

- Medium*, 3 Nov. 2020, [towardsdatascience.com/speech-emotion-recognition-using-ravdess-audio-dataset-ce19d162690](https://towardsdatascience.com/speech-emotion-recognition-using-ravdess-audio-dataset-ce19d162690). Accessed 21 Aug. 2022.
- . "Speech Emotion Recognition Using RAVDESS Audio Dataset." *Medium*, 3 Nov. 2020, [towardsdatascience.com/speech-emotion-recognition-using-ravdess-audio-dataset-ce19d162690](https://towardsdatascience.com/speech-emotion-recognition-using-ravdess-audio-dataset-ce19d162690).
- K S, Ajil, et al. "Recording Sound with Python." *Www.youtube.com*, 10 June 2021, [www.youtube.com/watch?v=e9CRZEi\\_feA](https://www.youtube.com/watch?v=e9CRZEi_feA). Accessed 24 Aug. 2022.
- K Scott, Sophie, et al. "Perceptual Cues in Nonverbal Vocal Expressions of Emotion." *Perceptual Cues in Nonverbal Vocal Expressions of Emotion*, Nov. 2010, [10.1080/17470211003721642](https://doi.org/10.1080/17470211003721642).
- Kuhn, Lisa. *EMOTION RECOGNITION in the HUMAN FACE and VOICE*. 2014.
- Learning, Machine. "Introduction to OpenCV Python Tutorials." *Machine Learning Tutorials, Courses and Certifications*, 4 Apr. 2021, [machinelearning.org.in/introduction-to-opencv-python-tutorials/#:~:text=Getting%20Started%20with%20OpenCV-Python%20OpenCV%20is%20a%20huge](https://machinelearning.org.in/introduction-to-opencv-python-tutorials/#:~:text=Getting%20Started%20with%20OpenCV-Python%20OpenCV%20is%20a%20huge). Accessed 9 Sept. 2022.
- Li, Chao, et al. "Exploring Temporal Representations by Leveraging Attention-Based Bidirectional LSTM-RNNs for Multi-Modal Emotion Recognition." *Information Processing & Management*, vol. 57, no. 3, May 2020, p. 102185, [10.1016/j.ipm.2019.102185](https://doi.org/10.1016/j.ipm.2019.102185). Accessed 15 Apr. 2021.
- Liu, Yishu, and Guifang Fu. "Emotion Recognition by Deeply Learned Multi-Channel Textual and EEG Features." *Future Generation Computer Systems*, Jan. 2021, [10.1016/j.future.2021.01.010](https://doi.org/10.1016/j.future.2021.01.010). Accessed 25 Jan. 2021.
- Livingstone, Steven R., and Frank A. Russo. "The Ryerson Audio-Visual Database of

- Emotional Speech and Song (RAVDESS).” *Zenodo*, Zenodo, 5 Apr. 2018, [zenodo.org/record/1188976](https://zenodo.org/record/1188976).
- . “The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A Dynamic, Multimodal Set of Facial and Vocal Expressions in North American English.” *PLOS ONE*, vol. 13, no. 5, 16 May 2018, p. e0196391, [journals.plos.org/plosone/article?id=10.1371/journal.pone.0196391](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0196391), [10.1371/journal.pone.0196391](https://doi.org/10.1371/journal.pone.0196391).
- Mehta, Dhvani, et al. “Recognition of Emotion Intensities Using Machine Learning Algorithms: A Comparative Study.” *Sensors*, vol. 19, no. 8, 21 Apr. 2019, p. 1897, [10.3390/s19081897](https://doi.org/10.3390/s19081897). Accessed 13 Oct. 2020.
- Mishra, Mayank. “Convolutional Neural Networks, Explained.” *Medium*, 2 Sept. 2020, [towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939#:~:text=%20Convolutional%20Neural%20Networks%2C%20Explained%20%201%20Convolutional](https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939#:~:text=%20Convolutional%20Neural%20Networks%2C%20Explained%20%201%20Convolutional).
- Mody, Devansh. “Emotion Recognition from Speech.” *Www.youtube.com*, 13 Aug. 2021, [youtu.be/CWoMpUu0rHg](https://youtu.be/CWoMpUu0rHg). Accessed 28 Aug. 2022.
- . “Emotion Recognition from Speech Installation and Usage.” *Www.youtube.com*, 13 Aug. 2021, [youtu.be/kjttI89pIrI](https://youtu.be/kjttI89pIrI). Accessed 28 Aug. 2022.
- Moore, Susan. “13 Surprising Uses for Emotion Ai Technology.” *Gartner*, 11 Sept. 2018, [www.gartner.com/smarterwithgartner/13-surprising-uses-for-emotion-ai-technology#:~:text=These%20uses%20include%3A%20Video%20gaming.%20Using%20computer%20vision%2C](https://www.gartner.com/smarterwithgartner/13-surprising-uses-for-emotion-ai-technology#:~:text=These%20uses%20include%3A%20Video%20gaming.%20Using%20computer%20vision%2C). Accessed 8 July 2022.
- NHS. “10 Stress Busters.” *Nhs.uk*, 1 Feb. 2021, [www.nhs.uk/mental-health/self-help/guides-tools-and-activities/tips-to-reduce-stress/](https://www.nhs.uk/mental-health/self-help/guides-tools-and-activities/tips-to-reduce-stress/).

- Nik. “Python: Get Dictionary Key with the Max Value (4 Ways) • Datagy.” *Datagy*, 26 Sept. 2021, [datagy.io/python-get-dictionary-key-with-max-value/#:~:text=max\\_value%20%3D%20max%20%28ages.values%20%28%29%29%20print%20%28max\\_value%29%20%23](https://datagy.io/python-get-dictionary-key-with-max-value/#:~:text=max_value%20%3D%20max%20%28ages.values%20%28%29%29%20print%20%28max_value%29%20%23). Accessed 25 Aug. 2022.
- Noema, Yaniv. “6 Amazing Algorithms for Detecting Faces in Images with Python Code References.” *Imagescv*, 16 Jan. 2022, [medium.com/imagescv/6-amazing-algorithms-for-detecting-faces-in-images-23d4d2106e13](https://medium.com/imagescv/6-amazing-algorithms-for-detecting-faces-in-images-23d4d2106e13). Accessed 7 July 2022.
- . “Python Computer Vision Libraries Every Developer Should Know.” *Imagescv*, 27 Feb. 2022, [medium.com/imagescv/python-computer-vision-libraries-every-developer-should-know-b7ab71734dc6#:~:text=%20Python%20Computer%20Vision%20Libraries%20Every%20Developer%20Should](https://medium.com/imagescv/python-computer-vision-libraries-every-developer-should-know-b7ab71734dc6#:~:text=%20Python%20Computer%20Vision%20Libraries%20Every%20Developer%20Should). Accessed 7 July 2022.
- Ooi, Chien Shing, et al. “A New Approach of Audio Emotion Recognition.” *Expert Systems with Applications*, vol. 41, no. 13, Oct. 2014, pp. 5858–5869, [10.1016/j.eswa.2014.03.026](https://doi.org/10.1016/j.eswa.2014.03.026). Accessed 25 Jan. 2021.
- OpenCV. “OpenCV: Cascade Classifier.” *Docs.opencv.org*, 4 Aug. 2022, [docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html).
- Pai Thon, Aditya. “Cascade Classifier Training — OpenCV 2.4.13.7 Documentation.” *Docs.opencv.org*, 21 Nov. 2018, [docs.opencv.org/2.4/doc/user\\_guide/ug\\_traincascade.html](https://docs.opencv.org/2.4/doc/user_guide/ug_traincascade.html).
- Pao, James. *Emotion Detection through Facial Feature Recognition*.
- Pragati. “How to Capture and Play Video in Google Colab?” *Knowledge Transfer*, 19 Dec. 2020, [androidkt.com/how-to-capture-and-play-video-in-google-colab/](https://androidkt.com/how-to-capture-and-play-video-in-google-colab/). Accessed 25 Aug. 2022.



Rajeev Thaware. "Real-Time Face Detection and Recognition with SVM and HOG Features - EEWeb." *EEWeb*, 28 May 2018, [www.eeweb.com/real-time-face-detection-and-recognition-with-svm-and-hog-features/](http://www.eeweb.com/real-time-face-detection-and-recognition-with-svm-and-hog-features/).

"RAVDESS Facial Landmark Tracking" by Swanson, Livingstone, & Russo is licensed under CC BY-NA-SC 4.0

ReportLinker. "The Global Emotion Detection and Recognition Market Size Is Projected to Grow from USD 23.6 Billion in 2022 to USD 43.3 Billion by 2027, at a Compound Annual Growth Rate (CAGR) of 12.9%." *GlobeNewswire News Room*, 31 Mar. 2022, [www.globenewswire.com/news-release/2022/03/31/2413491/0/en/The-global-emotion-detection-and-recognition-market-size-is-projected-to-grow-from-USD-23-6-billion-in-2022-to-USD-43-3-billion-by-2027-at-a-Compound-Annual-Growth-Rate-CAGR-of-12-.html](http://www.globenewswire.com/news-release/2022/03/31/2413491/0/en/The-global-emotion-detection-and-recognition-market-size-is-projected-to-grow-from-USD-23-6-billion-in-2022-to-USD-43-3-billion-by-2027-at-a-Compound-Annual-Growth-Rate-CAGR-of-12-.html). Accessed 6 Sept. 2022.

RSIS. "A Survey on Various Techniques of Human Emotion Recognition and Its Applications." *Research and Scientific Innovation Society (RSIS International)*, 24 Feb. 2019, [www.rsisinternational.org/virtual-library/papers/a-survey-on-various-techniques-of-human-emotion-recognition-and-its-applications/](http://www.rsisinternational.org/virtual-library/papers/a-survey-on-various-techniques-of-human-emotion-recognition-and-its-applications/). Accessed 9 Sept. 2022.

Sahu, Shipra. "The Future of Emotion Recognition in Machine Learning and AI." *Visionify*, 10 Feb. 2022, [visionify.ai/the-future-of-emotion-recognition-in-machine-learning-and-ai/](http://visionify.ai/the-future-of-emotion-recognition-in-machine-learning-and-ai/).

Seo, Naotoshi. "Tutorial: OpenCV Haartraining (Rapid Object Detection with a Cascade of Boosted Classifiers Based on Haar-like Features) - Naotoshi Seo." *Note.sonots.com*, 3 July 2007, [note.sonots.com/SciSoftware/haartraining.html](http://note.sonots.com/SciSoftware/haartraining.html).

- Sightcorp. "How Facial Recognition Is Used in Healthcare | Sightcorp." *Sightcorp.com*, 23 Mar. 2019, [sightcorp.com/blog/how-facial-recognition-is-used-in-healthcare/#:~:text=Real-time%20emotion%20detection%20is%20yet%20another%20valuable%20application](https://sightcorp.com/blog/how-facial-recognition-is-used-in-healthcare/#:~:text=Real-time%20emotion%20detection%20is%20yet%20another%20valuable%20application). Accessed 7 Sept. 2022.
- Singh, Aditya. "OPENCV in COMPUTER VISION." *Medium*, 8 June 2022, [medium.com/@adityasingh8717/opencv-in-computer-vision-204f682b688d](https://medium.com/@adityasingh8717/opencv-in-computer-vision-204f682b688d). Accessed 7 July 2022.
- Singh, Mandeep, and Yuan Fang. "Emotion Recognition in Audio and Video Using Deep Neural Networks." *ArXiv:2006.08129 [Cs, Eess]*, 15 June 2020, [arxiv.org/abs/2006.08129](https://arxiv.org/abs/2006.08129). Accessed 7 Sept. 2022.
- Stewart, Conor. "Topic: Stress in the UK." *Statista*, 30 May 2022, [www.statista.com/topics/6735/stress-in-the-uk/#topicHeader\\_\\_wrapper](https://www.statista.com/topics/6735/stress-in-the-uk/#topicHeader__wrapper).
- Sydorenko, Iryna. "Can AI Detect Emotion?" *Labelyourdata.com*, 25 Aug. 2021, [labelyourdata.com/articles/ai-emotion-recognition#:~:text=So%20basically%20what%20emotion%20recognition%20algorithms%20do%20is](https://labelyourdata.com/articles/ai-emotion-recognition#:~:text=So%20basically%20what%20emotion%20recognition%20algorithms%20do%20is). Accessed 7 July 2022.
- . "Can AI Detect Emotion?" *Labelyourdata.com*, 25 Aug. 2021, [labelyourdata.com/articles/ai-emotion-recognition](https://labelyourdata.com/articles/ai-emotion-recognition).
- Tan, Edwin. "How to Train a Deep Learning Sentiment Analysis Model." *Medium*, 28 Jan. 2022, [towardsdatascience.com/how-to-train-a-deep-learning-sentiment-analysis-model-4716c946c2ea](https://towardsdatascience.com/how-to-train-a-deep-learning-sentiment-analysis-model-4716c946c2ea). Accessed 21 Aug. 2022.
- . "Sentiment Analysis in Python with 3 Lines of Code." *Medium*, 9 Aug. 2021, [python.plainenglish.io/sentiment-analysis-in-python-with-3-lines-of-code](https://python.plainenglish.io/sentiment-analysis-in-python-with-3-lines-of-code).

[9382a649c23d](#). Accessed 21 Aug. 2022.

Team, Gradio. “Gradio Docs.” *Gradio.app*, [gradio.app/docs/#video](https://gradio.app/docs/#video). Accessed 5 Sept. 2022.

Team, Great Learning. “Face Detection Using Viola Jones Algorithm.” *GreatLearning Blog: Free Resources What Matters to Shape Your Career!*, 2 Sept. 2020, [www.mygreatlearning.com/blog/viola-jones-algorithm/#:~:text=%20The%20Viola%20Jones%20algorithm%20has%20four%20main](https://www.mygreatlearning.com/blog/viola-jones-algorithm/#:~:text=%20The%20Viola%20Jones%20algorithm%20has%20four%20main). Accessed 8 Sept. 2022.

Team. “Text2emotion: Detecting Emotions behind the Text, Text2emotion Package Will Help You to Understand the Emotions in Textual Meassages.” *PyPI*, 2 Sept. 2020, [pypi.org/project/text2emotion/](https://pypi.org/project/text2emotion/).

Tivatansakul, Somchanok, and Michiko Ohkura. “The Design, Implementation and Evaluation of a Relaxation Service with Facial Emotion Detection.” *IEEE Xplore*, 1 Dec. 2014, [ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7007832](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7007832). Accessed 7 July 2022.

“An Introduction to LibROSA for Working with Audio.” *OpenGenus IQ: Computing Expertise & Legacy*, 13 Sept. 2019, [iq.opengenus.org/introduction-to-librosa/](https://iq.opengenus.org/introduction-to-librosa/). Accessed 9 Sept. 2022.

“Compare Your Deep Learning Models on Web App| Python|.” *Www.youtube.com*, [www.youtube.com/watch?v=pg6qHLEsJEU&list=PLqYFiz7NM\\_SO0VpdeaaLTkcYq4DMoi72S&index=4](https://www.youtube.com/watch?v=pg6qHLEsJEU&list=PLqYFiz7NM_SO0VpdeaaLTkcYq4DMoi72S&index=4). Accessed 5 Sept. 2022.

“Crema\_d | TensorFlow Datasets.” *TensorFlow*, [www.tensorflow.org/datasets/catalog/crema\\_d](https://www.tensorflow.org/datasets/catalog/crema_d). Accessed 28 Aug. 2022.

“Emotion Detection and Recognition Market by Technology & Software Tool - 2024|

MarketsandMarkets.” *Www.marketsandmarkets.com*, Mar. 2022,  
[www.marketsandmarkets.com/Market-Reports/emotion-detection-recognition-market-23376176.html](http://www.marketsandmarkets.com/Market-Reports/emotion-detection-recognition-market-23376176.html).

“Emotion Detection Using Python | Convolutional Neural Network | Python Applications | Great Learning.” *Www.youtube.com*,  
[www.youtube.com/watch?v=m0fWjP3yIEo](http://www.youtube.com/watch?v=m0fWjP3yIEo). Accessed 7 May 2021.

“Facial Emotion Recognition Project Using CNN with Source Code.” *ProjectPro*,  
[www.projectpro.io/article/facial-emotion-recognition-project-using-cnn-with-source-code/570](http://www.projectpro.io/article/facial-emotion-recognition-project-using-cnn-with-source-code/570). Accessed 9 Sept. 2022.

“Google Colab - Executing External Python Files.” *Www.tutorialspoint.com*,  
[www.tutorialspoint.com/google\\_colab/google\\_colab\\_executing\\_external\\_python\\_files.htm#:~:text=Type%20a%20few%20letters%20like%20%E2%80%9Cm%E2%80%9D%20in%20the](http://www.tutorialspoint.com/google_colab/google_colab_executing_external_python_files.htm#:~:text=Type%20a%20few%20letters%20like%20%E2%80%9Cm%E2%80%9D%20in%20the). Accessed 11 Aug. 2022.

“Google Colaboratory - How Can Restart Googlecolab Runtime with Everything Clear?” *Stack Overflow*, [stackoverflow.com/questions/50361934/how-can-restart-googlecolab-runtime-with-everything-clear#:~:text=In%20latest%20Google%20Colab%20version%2C%20we%20should%20select](https://stackoverflow.com/questions/50361934/how-can-restart-googlecolab-runtime-with-everything-clear#:~:text=In%20latest%20Google%20Colab%20version%2C%20we%20should%20select). Accessed 19 Aug. 2022.

“Google Colaboratory.” *Colab.research.google.com*,  
[colab.research.google.com/drive/1b8pVMMoR37a3b9ICo8TMqMLVD-WvbzTk#scrollTo=pD3e08HvWsF3](https://colab.research.google.com/drive/1b8pVMMoR37a3b9ICo8TMqMLVD-WvbzTk#scrollTo=pD3e08HvWsF3). Accessed 11 Aug. 2022.

“Google Colaboratory.” *Colab.research.google.com*,  
[colab.research.google.com/github/yunooooo/FFmpeg-for-Google-Drive/blob/master/FFmpeg.ipynb#scrollTo=nSeO98YQoTJe](https://colab.research.google.com/github/yunooooo/FFmpeg-for-Google-Drive/blob/master/FFmpeg.ipynb#scrollTo=nSeO98YQoTJe). Accessed 30 Aug.

2022.

“Gradio Course - Create User Interfaces for Machine Learning Models.”

*Www.youtube.com*, [www.youtube.com/watch?v=RiCQzBluTxU](http://www.youtube.com/watch?v=RiCQzBluTxU). Accessed 5 Sept.

2022.

“How to Use Webcam in Google Colab for Images and Video (FACE DETECTION).”

*Www.youtube.com*, 15 Dec. 2020,

[www.youtube.com/watch?v=YjWh7QvVH60&t=134s](http://www.youtube.com/watch?v=YjWh7QvVH60&t=134s). Accessed 21 Aug. 2022.

“Intuition of Adam Optimizer.” *GeeksforGeeks*, 22 Oct. 2020,

[www.geeksforgeeks.org/intuition-of-adam-optimizer/](http://www.geeksforgeeks.org/intuition-of-adam-optimizer/).

“Pandas.DataFrame.T() Function in Python.” *GeeksforGeeks*, 1 Oct. 2020,

[www.geeksforgeeks.org/pandas-dataframe-t-function-in-python/](http://www.geeksforgeeks.org/pandas-dataframe-t-function-in-python/). Accessed 3

Sept. 2022.

“Papers with Code - FER2013 Dataset.” *Paperswithcode.com*,

[paperswithcode.com/dataset/fer2013](http://paperswithcode.com/dataset/fer2013). Accessed 9 Sept. 2022.

“Papers with Code - RAVDESS Dataset.” *Paperswithcode.com*,

[paperswithcode.com/dataset/ravdess](http://paperswithcode.com/dataset/ravdess). Accessed 21 Aug. 2022.

“Papers with Code - Speech Emotion Recognition.” *Paperswithcode.com*,

[paperswithcode.com/task/speech-emotion-recognition](http://paperswithcode.com/task/speech-emotion-recognition). Accessed 17 Aug. 2022.

Contact us on: [hello@paperswithcode.com](mailto:hello@paperswithcode.com). Papers With Code is a free resource with all data licensed under CC-BY-SA. Terms Data policy Cookies policy from .

“Parsing - How Can I Extract Audio from Video with Ffmpeg?” *Stack Overflow*,

[stackoverflow.com/questions/9913032/how-can-i-extract-audio-from-video-with-ffmpeg](http://stackoverflow.com/questions/9913032/how-can-i-extract-audio-from-video-with-ffmpeg). Accessed 30 Aug. 2022.

“Python 3.x - Upload File from Colab to Google Drive Folder.” *Stack Overflow*,

- [stackoverflow.com/questions/50647677/upload-file-from-colab-to-google-drive-folder](https://stackoverflow.com/questions/50647677/upload-file-from-colab-to-google-drive-folder). Accessed 31 Aug. 2022.
- “Python Mini Project - Speech Emotion Recognition with Librosa.” *DataFlair*, 17 Sept. 2019, [data-flair.training/blogs/python-mini-project-speech-emotion-recognition/](https://data-flair.training/blogs/python-mini-project-speech-emotion-recognition/).
- “Python Pillow (PIL) Tutorial - Image Manipulation.” *CodersLegacy*, [coderslegacy.com/python/pillow-pil-tutorial/#:~:text=You%20can%20also%20use%20the%20Pillow%20Library%20together](https://coderslegacy.com/python/pillow-pil-tutorial/#:~:text=You%20can%20also%20use%20the%20Pillow%20Library%20together). Accessed 9 Sept. 2022.
- “Sample Masters Big Data Full Dissertation.” *Research Prospect*, [www.researchprospect.com/masters-big-data-full-dissertation-sample/](http://www.researchprospect.com/masters-big-data-full-dissertation-sample/). Accessed 7 Sept. 2022.
- “Tensorflow.” *PyPI*, 26 Feb. 2019, [pypi.org/project/tensorflow/](https://pypi.org/project/tensorflow/). Accessed 27 Mar. 2019.
- “Understanding ROC Curves with Python.” *Stack Abuse*, 25 Feb. 2019, [stackabuse.com/understanding-roc-curves-with-python/](https://stackabuse.com/understanding-roc-curves-with-python/).
- “What Is a Convolutional Neural Network?” *Uk.mathworks.com*, [uk.mathworks.com/discovery/convolutional-neural-network-matlab.html#how-they-work](https://uk.mathworks.com/discovery/convolutional-neural-network-matlab.html#how-they-work). Accessed 8 Sept. 2022.
- Vidhya, Analytics. “NumPy Basics: Machine Learning in Python.” *Analytics Vidhya*, 30 Dec. 2020, [medium.com/analytics-vidhya/numpy-basics-machine-learning-in-python-795c39d85bb4](https://medium.com/analytics-vidhya/numpy-basics-machine-learning-in-python-795c39d85bb4).
- VINEETA. “Real Time Emotion Analysis (Sound and Face) Using Python, Deep Neural Networks.” *YouTube*, 22 Dec. 2020, [www.youtube.com/watch?v=bdENUrwdx5s](https://www.youtube.com/watch?v=bdENUrwdx5s). Accessed 24 May 2022.
- Viola, Paul, and Michael Jones. “Rapid Object Detection Using a Boosted Cascade of

Simple Features.” *ACCEPTED CONFERENCE on COMPUTER VISION and PATTERN RECOGNITION*, 2001,

[www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf](http://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf).

Wang, Yi-Qing. “An Analysis of the Viola-Jones Face Detection Algorithm.”

*Researchgate.net*, June 2014,

[www.researchgate.net/publication/272643562\\_An\\_Analysis\\_of\\_the\\_Viola-Jones\\_Face\\_Detection\\_Algorithm](http://www.researchgate.net/publication/272643562_An_Analysis_of_the_Viola-Jones_Face_Detection_Algorithm). 10.5201/ipol.2014.104.

Xu, Wei, et al. “The Relationship between Stress and Negative Emotion: The Mediating

Role of Rumination.” *Www.oatext.com*, 31 Jan. 2018, [www.oatext.com/the-relationship-between-stress-and-negative-emotion-the-mediating-role-of-rumination.php#jumpmenu3](http://www.oatext.com/the-relationship-between-stress-and-negative-emotion-the-mediating-role-of-rumination.php#jumpmenu3). Accessed 6 Sept. 2022.


Zinjad, Saurabh. “Speech Emotion Recognition | Deep Learning | NLP.”

*Www.youtube.com*, June 12AD, [www.youtube.com/watch?v=yvxpxcncSGs](http://www.youtube.com/watch?v=yvxpxcncSGs). Accessed 28 Aug. 2022.

## 10. Appendices

### 10.1 Appendix A: Ethics Application

[Home](#) | [About Us](#) | [Contact Us](#) | [Research Methodology](#) | [Research Guidelines](#) | [Privacy](#)



---

## Ethical clearance for research and innovation projects

**Project status**

**Status**

Approved

**Actions**

Date	Who	Action	Comments
17:39:00 08 July 2022	Femi Isiaq	Supervisor approved	Ensure the images you have downloaded and will use for the research work only are authorized with no issues surrounding copyrights.
20:25:00 07 July 2022	Sony Abraham	Principal investigator submitted	
20:23:00 07 July 2022	Sony Abraham	Principal investigator saved	
20:20:00 07 July 2022	Sony Abraham	Principal investigator saved	

[Get Help](#)

### Ethics release checklist (ERC)

**Project details**

Project name:

Principal investigator:

Faculty:

Level:

Course:

Unit code:

Figure 22 Ethics Application (1)



Supervisor name:

Supervisor search:

Other investigators:

**Checklist**

Question	Yes	No
Q1. Will the project involve human participants other than the investigator(s)?	<input checked="" type="radio"/>	<input checked="" type="radio"/>
Q1a. Will the project involve vulnerable participants such as children, young people, disabled people, the elderly, people with declared mental health issues, prisoners, people in health or social care settings, addicts, or those with learning difficulties or cognitive impairment either contacted directly or via a gatekeeper (for example a professional who runs an organisation through which participants are accessed), a service provider, a care-giver, a relative or a guardian?	<input type="radio"/>	<input checked="" type="radio"/>
Q1b. Will the project involve the use of control groups or the use of deception?	<input type="radio"/>	<input checked="" type="radio"/>
Q1c. Will the project involve any risk to the participants' health (e.g. intensive intervention such as the administration of drugs or other substances, or vigorous physical exercise), or involve psychological stress, anxiety, humiliation, physical pain or discomfort to the investigator(s) and/or the participants?	<input type="radio"/>	<input checked="" type="radio"/>
Q1d. Will the project involve financial inducement offered to participants other than reasonable expenses and compensation for time?	<input type="radio"/>	<input checked="" type="radio"/>
Q1e. Will the project be carried out by individuals unconnected with the University but who wish to use staff and/or students of the University as participants?	<input type="radio"/>	<input checked="" type="radio"/>
Q2. Will the project involve sensitive materials or topics that might be considered offensive, distressing, politically or socially sensitive, deeply personal or in breach of the law (for example criminal activities, sexual behaviour, ethnic status, personal appearance, experience of violence, addiction, religion, or financial circumstances)?	<input checked="" type="radio"/>	<input checked="" type="radio"/>
Q3. Will the project have detrimental impact on the environment, habitat or species?	<input checked="" type="radio"/>	<input checked="" type="radio"/>
Q4. Will the project involve living animal subjects?	<input checked="" type="radio"/>	<input checked="" type="radio"/>
Q5. Will the project involve the development for export of 'controlled' goods regulated by the Export Control Organisation (ECO)? (This specifically means military goods, so-called dual-use goods (which are civilian goods but with a potential military use or application), products used for torture and repression, radioactive sources.) <a href="https://www.gov.uk/government/organisations/export-control-organisation">Further information from the Export Control Organisation (https://www.gov.uk/government/organisations/export-control-organisation)</a>	<input type="radio"/>	<input checked="" type="radio"/>
Q6. Does your research involve the storage of records on a computer, electronic transmissions, or visits to websites, which are associated with terrorist or extreme groups or other security sensitive material? <a href="https://ico.org.uk/for-organisations/uk-to-data-protection/">Further information from the Information Commissioner's Office (https://ico.org.uk/for-organisations/uk-to-data-protection/)</a>	<input checked="" type="radio"/>	<input checked="" type="radio"/>

Get Help

**Declarations**

I/we, the investigator(s), confirm that:

Figure 23 Ethics Application (2)

The information contained in this checklist is correct.

I/we have assessed the ethical considerations in relation to the project in line with the University Ethics Policy.

I/we understand that the ethical considerations of the project will need to be re-assessed if there are any changes to it.

I/we will endeavour to preserve the reputation of the University and protect the health and safety of all those involved when conducting this research/enterprise project.

If personal data is to be collected as part of my project, I confirm that my project and I, as Principal Investigator, will adhere to the General Data Protection Regulation (GDPR) and the Data Protection Act 2018. I also confirm that I will seek advice on the DPA, as necessary, by referring to the [Information Commissioner's Office further guidance on DPA](#) (<https://ico.org.uk/the-commissioner/why-ask-ico-data-protection-dpa/>) and/or by contacting [information.rights@ucl.ac.uk](mailto:information.rights@ucl.ac.uk) [ ]. By Personal data, I understand any data that I will collect as part of my project that can identify an individual, whether in personal or family life, business or profession.

I/we have read the [research grants \(https://www.ucl.ac.uk/research/grants\) research duty guidance \(https://www.ucl.ac.uk/research/grants\) for further information on ethics and safety](#).

[Privacy policy](#) | [Cookies](#) | [Disclaimer](#) | [Accessibility statement](#)  
 © UCL 2024

Figure 24 Ethics Application (3)