Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Software Department

Classifying Emotions Based on Visual and Audio Modalities Using Deep Learning Networks

A Dissertation

Submitted to the Council of the College of Information Technology at University of Babylon in Partial Fulfillment of the Requirements for the Degree of Doctorate of Philosophy in Information Technology / Software

By

Ahmed Samit Hatem Abdul-Kareem

Supervised by

Prof. Dr. Abbas Mohsen Al-Bakry

2023 A.C. **1445 A.H.**

بِسْـمِ اللَّهِ الرحمن الرَّحِيـمِ

﴿ قَالَ رَبِّ أَوْزِعْنِي أَنْ أَشْكُرَ نِعْمَتَكَ الَّتِي أَنْعَمْتَ
عَلَيَّ وَعَلَىٰ وَالِدَيَّ وَأَنْ أَعْمَلَ صَالِحًا تَرْضَاهُ
وَأَدْخِلْنِي بِرَحْمَتِكَ فِي عِبَادِكَ الصَّالِحِينَ﴾

صدق الله العلي العظيم

سورة النمل / آية 19

Supervisor Certification

I certify that this dissertation was prepared under my supervision at the Department of Software / Collage of Information Technology / Babylon University, by **Ahmed Samit Hatem** as a partial fulfillment of the requirements for the degree of **Ph.D. in Information Technology**.

Signature:
Name: **Dr Abbas Mohsen Al-Bakry**
Title: **Professor**
Date: / / 2023

The Head of the Department Certification

In view of the available recommendation, we forward this dissertation for debate by the examining committee.

Signature:
Name: **Dr. Sura Zaki Alrashid**
Title**: Assistant Professor**
Date: / / 2023

Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled (**Classifying Emotions Based on Visual and Audio Modalities Using Deep Learning Networks**) presented by the student (**Ahmed Samit Hatem**) and examined him in its content and what is related to it, and that, in our opinion, it is adequate with (**Very Good**) standing as a thesis for the degree of Doctor of Philosophy in Information Technology-Software.

Signature:
Name: **Dr. Eman Salih Al-Shamery**
Title: **Professor**
Date: / / 202
(**Chairman**)

Signature:
Name: **Dr.Wafaa Mohammed Saeed**
Title: **Professor**
Date: / / 202
(**Member**)

Signature:
Name: **Dr.Karim Qasim Hussein**
Title: **Professor**
Date: / / 202
(**Member**)

Signature:
Name: **Dr. May A. Salih**
Title: **Assistant Professor**
Date: / / 202
(**Member**)

Signature:
Name: **Ahmed Habeeb Al-Azawei**
Title: **Assistant Professor**
Date: / / 202
(**Member**)

Signature:
Name: **Abbas Mohsin Al-Bakry**
Title: **Professor**
Date: / / 202
(**Member and Supervisor**)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name: **Dr.Wesam S. Bhaya**
Title: **Professor**
Date: / / 202
(**Dean of College of Information Technology**)

Declaration

I hereby declare that this dissertation, "**Classifying Emotions Based on Visual and Audio Modalities Using Deep Learning Networks**" submitted to University of Babylon in partial fulfillment of requirements for the degree of Doctorate of Philosophy in Information Technology-Software has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for reports and summaries whose sources are appropriately cited in the references.

Signature:

Name: **Ahmed Samit Hatem**

Title:

Date: / / 202

Acknowledgements

First of all, I thank ***Allah***, my Creator, for His help and acceptance of my prays that makes the accomplishment of this work more than a dream after my hard conditions.

I would like to express my deep thanks and gratitude to my supervisor; **Prof. Dr. Abbas Mohsen Al-Bakry** for his valuable guidance and encouragement during the development of my work.

Special thanks go to the staff of the College of information Technology for their faithful efforts to give us the utmost scientific topics and endless support in all directions, especially **Dr. Sura Zaki Alrashid, Dr. Ahmed Saleem Abbas** and **Dr. Mazin Kazim** for the great assistance they provided me.

I must express my very profound gratitude to my mother and my wife for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation.

Last but not least, I would like to thank all the kind, helpful and lovely people who helped me directly or indirectly to complete this work and apologize to them for not being able to mention them by name here, though they are in my heart.

Ahmed Samit

Dedication

To

My father, although he was my inspiration to

pursue my doctoral degree, he was unable to see

my graduation. This is for his soul.

To

The soul of My martyred brothers, Muneer and Naseem.

To

My Family,

My lovely Mother,

My dear Wife,

My darling sons,

My lovely daughter.

Ahmed Samit

Abstract

Emotions play a main role in many aspects of our lives, and they may influence or even determine our thinking and decision-making. Emotion recognition using multimodal data, such as video, audio, text, etc., is a challenging topic but an important area of research that has captured much attention from academics. A multimodal emotion recognition system based on visual and audio modalities is presented in this dissertation.

For the visual path, number of key-frames are selected from each video in sequence. Three pretrained convolution neural network (CNN) models have been trained by changing their classifiers with a proposed new classifier and selecting the best one. This model extracts the appropriate features from visual data and classifies them according to seven emotion classes with two cases of training by using two databases namely, AffectNet and The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) databases. In the audio path, different features from the audio parts are selected to classify them by the proposed multi-layer perceptron (MLP) model into seven emotion classes. The results of the proposed system depend on the decision-level-fusion method to obtain the output result of both visual and audio modalities. The ensample decision method is used in the fusion stage to obtain the final prediction from the output values of different predictors.

The experiments results suggest that in the visual path, the accuracy was 98%, for the best model of the RAVDESS database, while the AffectNet database is used to investigate the proposed model generalizability. In the audio path, the best model obtains an accuracy of 69% for the RAVDESS database. The final accuracy of the proposed system is 99% based on the ensemble decision fusion method.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
AER	Automatic Emotion Recognition
ANN	Artificial Neural Network
CNN	Convolution Neural Network
DBN	Deep Belief Network
DL	Deep Learning
EMODB	Berlin Database of Emotional Speech
ER	Emotion Recognition
FER	Facial Emotion Recognizer
HCI	Human Computer Interaction
HDF5	Hierarchical Data Format version 5
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LPCC	Linear Prediction Cepstral Coefficients
LSTM	Long Short-Term Memory
MFCCs	Mel Frequency Cepstral Coefficients
ML	Machine Learning
MLP	Multilayer Perceptron
PANNs	Pretrained Audio Neural Networks
RAVDESS	Ryerson Audio-Visual Database of Emotional Speech and Song
ReLU	Rectified Linear Unit
RMS	Root Mean Square
RNN	Recurrent Neural Network
SAVEE	Surrey Audio-Visual Expressed Emotion Database
SER	Speech Emotion Recognition
SGD	Stochastic Gradient Descent
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
TL	Transfer learning
WPT	Wavelet Packet Transform
ZCR	Zero Crossing Rate

Chapter One

General Introduction

1.1 Introduction

Emotions are fundamental features of humans that play important roles in social communication. Emotion recognition (ER) has received a lot of attention in the areas of human-machine interaction and affective computing [1]. Human-computer interactions (HCI) are the study of how humans communicate with computers. Computers are designed to execute certain activities in a manner that is pleasant or efficient. The first move towards a smart HCI is to have the ability to react and feel properly based on users' input [2]. The affective computing (also known as Artificial Emotional Intelligence) is an HCI sub-field inspired by the area of psychology that the "effect" is basically an analogous term for the concept of "emotion" [3]. When it comes to affective computing, the computer should be able to identify people's emotions and react appropriately to them. It is a multi-disciplinary area that includes computer science, psychology, sociology and cognitive science [4].

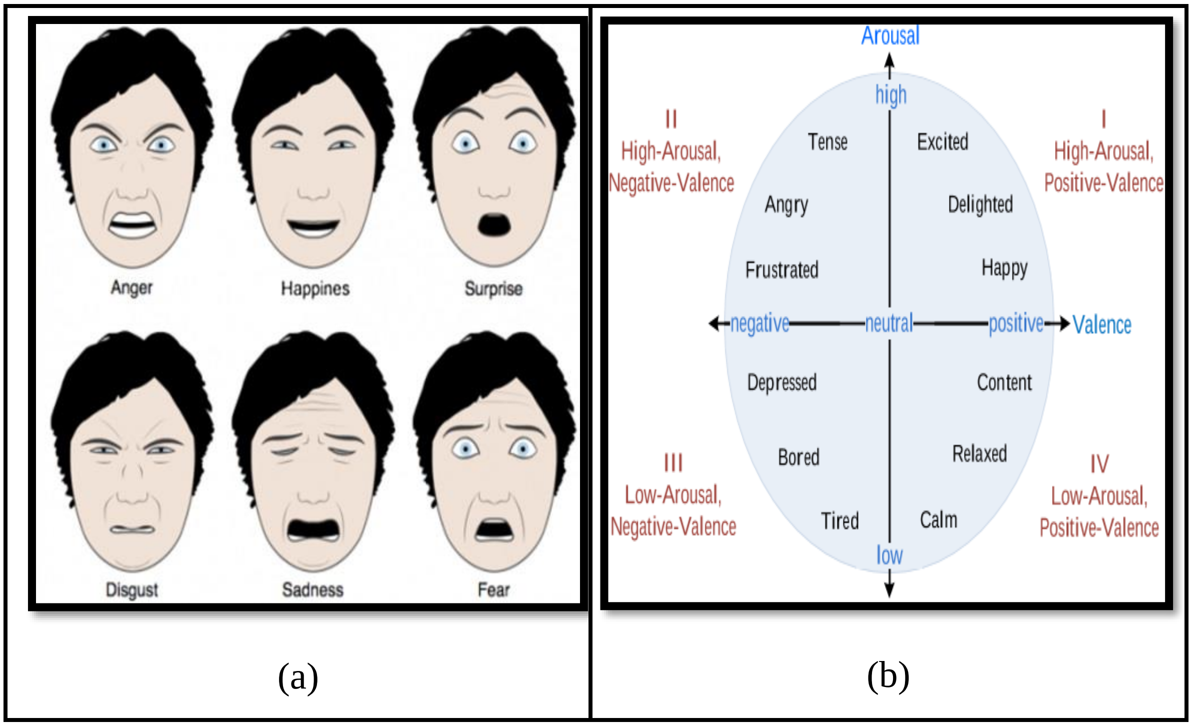
Emotion recognition is the method of recognizing emotions based on different signals such as facial and verbal expressions. Humans sense emotion through combining and processing multimodal, temporal, and asynchronous communication signals. Emotion recognition is ideally suited to multimodal scenarios, such as those encountered by humans; otherwise, it is impossible to infer someone's emotions. Emotion recognition works well in multimodal scenarios, as humans do. It is not always possible to deduce someone's emotion from staring at a single modality at a time [5].

One important question that may be asked when building an emotion recognition system is: which information resource is to be used for deducing emotions? People use different modalities to express their emotions, such as facial expressions, body language, and human speech. According to Mehrabian [6], three basic elements can be used during any face-to-face communication: (1) face, (2) speech acoustics, (3) spoken words. Facial expression and speech acoustics are the most beneficial informatics channels with percent 55% and 38%, respectively, while the spoken words channel is the least contributing one with 7% of the overall impression. However, the combination of these channels can increase the level of information that exists to deduce the emotional state [7], [8].

Efforts to develop emotionally intelligent entities have focused on empowering them to sense emotions. Intelligent agent systems are another research area related to the implementation of emotional intelligence in real-world domains. The key goal is to use rational approaches to provide a robust specification of how an artificial entity can execute emotions. Recent research in this field has focused on allowing software agents to sense and communicate emotions through expression and gestures using verbal, nonverbal, and textual cues [9].

1.2 Emotion Theories

Emotion is a complicated human function that can be studied from cognitive, physiological, and motivational perspectives. There are several scientific theories on emotion categorization based on psychological research. Categorical and dimensional emotion theories are two well-known emotion theories [10]. The categorical emotion theory is based on a discrete group of fundamental emotions. Ekman established the most well-known and widely agreed approach to fundamental emotions. According to Ekman, six basic emotions can be defined: happiness, anger, fear, disgust, surprise and sadness as described in Figure 1.1a [11]. Unlike categorical emotion theory, dimensional emotion theory asserts that emotions are associated in a systematic manner and can be expressed on a multidimensional space, as shown in Figure 1.1b [12].



(a) (b)

Figure 1.1: Emotion Theories, (a) categorical theory, (b) dimensional theory [11], [12]

The raters describe various verbal stimuli on bipolar scales comprised of two opposite adjective pairs, such as hot-cold, white-black, fast-slow, and so on. The two main dimensions are valence which means how positive or negative an emotion is) and arousal which refers to how intense an emotion is [13].

1.3 Information Resources of Emotion Recognition

Nonverbal and verbal interactions both contribute to the overall communication. Words and phrases are examples of oral information, while visuals and movement are examples of non-verbal information. The majority of our emotional communication takes place through non-verbal actions. This is supported by the research of Mehrabian [6] which discovered that non-verbal correspondence is the most common type, with the manner of speaking and body language accounting for 38 % and 55 % of emotional information, respectively, while the remaining 7% is made up of spoken words (the "38% - 55% -7% rule"). In this way, non-verbal information plays a critical role in the comprehension and synthesis of emotions [14].

Recently, researchers have focused on the use of non-verbal communication, such as eye movements, facial expressions, and physiological reactions, which use a singular channel of information, known as "mode" for example, speech, facial expression, and body language [15]. Approaches that only use speech or visual data separately fail in actual situations. Shadow, Occlusions, illumination conditions, and a variety of other factors, for example, are common conditions in which the precision of systems for visual frameworks decreases. Similarly, in speech frameworks, environmental noise or maybe a person moving while speaking is regarded as error sound sources. To solve problems of one modality, the emotions can also be conveyed through more than one mode at the same time. In multimodal emotion-based communication, the person can communicate his/her emotional state using a variety of information resources [16].

When communicating with others, facial expressions and words are highly effective ways of conveying feelings. As a result, this dissertation relies heavily on visual and auditory modalities.

1.4 Challenges of the Emotion Recognition System

It is challenging to recognize human emotions since extracting the best audio and visual characteristics that define such emotions needs high effort. Thus, there is a complex set of issues to consider while developing an emotion recognition system[17]–[21].

1. **The intensity of the expressions.**

   Since humans can express their emotions in a wide variety of contexts, the intensity of those feelings varies.

2. **Emotions in a talking face**

   During speaking, it is difficult to explain facial emotions since certain muscles of the face are shifted as the person speaks and this can affect the appearance of the face and therefore make it more difficult to determine expressed facial emotions.

3. **Time of Response**

   When interacting with a human, understanding his/her reactions and time of response is important for optimizing interaction performance. The response time and real-time reaction measurement would help adequately respond to humans.

1.5 Related Works

The development of audio-visual emotion recognizers has recently emerged as a fruitful study area in pattern recognition and AI. The following are examples of works on audio facial expression recognition:

Song et al. [22] proposed a fusion approach for diverse modalities to overcome the low recognition ability of facial and speech emotion recognizers. They used a facial expression recognizer that is developed based on CNN. A facial expression recognizer and a speech analytics engine developed for speech emotion recognition. The outputs of the two recognizers were done using a neural network. For simulations, eNTERFACE’05 and Surrey Audio-Visual Expressed Emotion (SAVEE) were utilized as test databases and the accuracy was 43.4% for fusion.

Zhang et al. [23] proposed a method for audio-visual emotion recognition by mixing CNN, 3D-CNN, and Deep Belief Network (DBN). The Mel-spectrogram illustration of the speech is inputted to CNN. 3D-CNNs are used to extract emotional features from video segments. The outputs of audio and visual networks are combined with a deep DBN model to fuse audio and visual cues. Audio path results were 66.17% for the RML database and 78.08% for eNTERFACE’05. Visual path results were 68.9% for RML and 54.35% for the eNTERFACE’05 database. For the fusion, the results were 80.36% and 85.97% for the two databases, respectively.

Jaratrotkamjorn and Choksuriwong [24] suggested a bimodal emotion identification system to identify human emotions. In this technique, 68 facial features were extracted using a combination of facial landmarks and a Gabor filter bank. In addition, 34 speech features were gleaned with the aid of the pyAudioAnalysis module. To classify these eight primary emotions, data from the face and the voice are combined at the feature level before being sent to the Deep belief network. Using the RAVDESS database, the obtained accuracy results were 97.92%.

Issa et al. [25] presented a new method, which extracts mel-frequency cepstral coefficients, chromagram, mel-scale spectrogram, Tonnetz representation, and spectral contrast features from sound files and uses them as inputs for the proposed one-dimensional Convolutional Neural Network for the recognition of emotions using samples from the RAVDESS, EMO-DB, and IEMOCAP datasets. The obtained accuracy was 71.61% for RAVDESS with 8 classes, 86.1% for EMO-DB, and 64.3% for IEMOCAP.

Siddiqui and Javaid [26] established a multimodal automatic emotion recognition (AER) framework that involves implementing an ensemble-based approach for the AER By fusing both visible and infrared pictures with speech. The framework is built using two distinct levels. Two convolutional neural networks (CNNs) were trained independently on visible and infrared images, and then transfer learning was used to extract features from both. At this point, a fusion was performed at the feature level, and the result was passed into a support vector machine for classification. The RAVDESS database's audio spectrogram characteristics were utilized to train a third convolutional neural network to understand the emotions from the speech. In the second layer, a decision-level fusion was fed the output of the SVM and the third CNN (for address). Overall accuracy was 86.36% after decision templates (a technique for fusing decisions at the decision level) were applied to the data from both the images and the speech.

Luna-Jiménez et al. [27] proposed an automatic emotion recognizer system that consisted of a speech emotion recognizer (SER) and a facial emotion recognizer (FER). For the SER, they evaluated a pre-trained xlsr-Wav2Vec2.0 transformer using two transfer-learning techniques: embedding extraction and fine-tuning. For the facial emotion recognizer, they extracted the Action Units of the videos.

Finally, combining these two modalities with a late fusion strategy, they achieved 86.70% accuracy on the RAVDESS dataset on a subject-wise 5-Cross Validation evaluation, classifying eight emotions.

Luna-Jiménez et al. [28] proposed a multimodal emotion recognition system that relies on speech and facial information. For the speech-based modality, They tested various transfer learning strategies, including embedding extraction and Fine-Tuning the CNN-14 within the Large-Scale Pretrained Audio Neural Networks (PANNs) architecture, to determine the most effective for the speech-based modality. They offer a framework for face emotion recognizers, including a Spatial Transformer Network pre-trained on saliency maps and facial images, followed by an LSTM network with an attention mechanism. Using a late fusion method to combine the results from both modalities, they improved accuracy to 80.08% on the RAVDESS dataset.

Puri et al. [29] developed a CNN-based network with eight consecutive layers of the convolution neural approach to classify the feelings conveyed in audio recordings. The researchers listened to recordings from the RAVDESS database of expressive speech and music. Mel-Frequency Cepstral Coefficients (MFCCs) and a Log Mel Spectrogram were utilized to display the unprocessed audio. Nearly 98% accuracy was achieved using the RAVDESS database.

Middya et al. [30] suggested independent convolutional neural networks (CNNs) to extract features from audio and video data and then fuse those features at the model level to produce an effective multimodal emotion identification model. Two-dimensional convolutional neural network (2-D CNN) with convolution, pooling, and flattening layers were utilized to extract video features. The input shape is of size (6, 64, 64, 3) where six frames are taken from each video; each frame is 64 by 64 pixels, and there are three channels in the input structure. A one-dimensional convolutional neural network (CNN) with convolution, pooling, dropout, and flattening layers is used as the audio model's feature extractor. The audio feature extractor has an input shape of (181 × 1), where 181 is the total size of the feature set (Mel spectrogram, spectral contrast, and tonnes).This 1-D CNN is comprised of three convolution layers and two pooling layers. The optimal multimodal emotion recognition model is created by fusing audio and video features at the model level. Two multimodal datasets, the RAVDESS and SAVEE , evaluate the proposed models' performances using various performance criteria. The accuracy of the SAVEE and RAVDESS datasets was 99% and 86%, respectively.

Bhangale and Kothandaraman [31] used a proposed one-dimensional deep convolutional neural network (1-D DCNN) to minimize the computational complexity and classify the acoustic feature set based on Mel frequency cepstral coefficients (MFCC), linear prediction cepstral coefficients (LPCC), wavelet packet transform (WPT), zero crossing rate (ZCR), spectrum centroid, spectral roll-off, spectral kurtosis, root mean square (RMS), pitch, jitter, and shimmer. The overall proposed systems’ performance is evaluated on the Berlin Database of Emotional Speech (EMODB) and the RAVDESS datasets. The obtained accuracy was 93.31% and 94.18% for the EMODB and RAVDESS datasets, respectively.

Although several approaches were proposed to predict emotions, such techniques often rely solely on either visual or audio features, leading to performance degradation under certain conditions. For instance, visual features can be negatively impacted by factors such as shadows, illumination, and occlusions, while audio features can be susceptible to environmental noise. This dissertation, therefore, proposes deep learning models that are based on audio and visual modalities to classify emotions accurately and improve the efficiency and performance of the detection system. Table 1.1 shows an overview of the relevant works listed above, together with the methodology, databases, and obtained results.

Table 1.1: An overview of the related works, with their methodology, used databases and obtained results

Authors	Methodology	Database	Accuracy
Song et al. [22].	CNN for the facial expression recognizer, speech analytics engine, ANN for classification.	eNTERFACE'05 and SAVEE	43.40%
Zhang et al. [23]	The Mel-spectrogram representation of the speech is inputted to CNN. 3D-CNNs to extract emotional features from video segments. DBN model to fuse audio and visual cues	RML and eNTERFACE'05	80.36% for RML and 85.97% for eNTERFAC E'05
Jaratrotkamjorn and Choksuriwong [24]	Gabor filter bank to extract 68 facial features, they also used the pyAudioAnalysis library to extract 34 speech features, feature-level fusion to audio& facial features, and Deep belief network for the classification.	RAVDESS	97.92%.
Issa et al. [25]	Extracts mel-frequency cepstral coefficients, chromagram, mel-scale spectrogram, Tonnetz, and spectral contrast features from the speech are inputted to 1D-CNN.	RAVDESS EMO-DB, and IEMOCAP datasets	71.61 % for RAVDESS, 86.1% for EMO-DB, and 64.3% for IEMOCAP .
Siddiqui and Javaid [26].	Two CNNs were trained using visible and infrared images, Transfer Learning to extract features from the image, and feature-level fusion to fuse the features for classification. Third CNN extracts audio spectrogram features, SVM and third CNN fed to decision-level fusion.	RAVDESS dataset	86.36%
Luna-Jiménez et al.[27].	For SER evaluated a pre-trained xlsr-Wav2Vec2.0 transformer using two transfer-learning techniques. For the facial emotion recognizer, they extracted the Action Units of the videos. Finally, combining these two modalities with a late fusion strategy.	RAVDESS dataset	86.70%
Luna-Jiménez et al.[28].	CNN-14 of the Large-Scale Pretrained Audio Neural Networks (PANNs) framework for speech path. Pre-trained Spatial Transformer Network on saliency	RAVDESS dataset	80.08%

	maps and facial images followed by an LSTM with an attention mechanism for facial emotion. The late fusion strategy to fuse the two stages.		
Puri et al. [29].	CNN contain 8 layers to classify Log Mel Spectrogram and Mel-Frequency Cepstral Coefficients (MFCCs) audio features	RAVDESS database	near to 98%.
Middya et al. [30].	CNN as video features extractor,1-D CNN for audio mfccs, melspectrogram, spectral contrast and tonnetz features extractor, fusing audio and video features at the model level.	RAVDESS and SAVEE database	99% for SAVEE and 86% for RAVDESS
Bhangale and Kothandaraman [31].	I-D CNN to classify feature sets (MFCC, LPCC , ZCR, spectrum centroid, spectral roll-off, spectral kurtosis, (RMS), pitch, jitter, shimmer and WPT)	EMODB and RAVDESS	93.31% for EMODB and 94.18% for RAVDESS
Proposed system	CNN to extract the features with a new classifier for the visual path, MLP to classify 5 types of features for the audio path, and the decision-level fusion strategy to fuse the two paths.	RAVDESS database	-

1.6 Problem Statement

The primary issue for constructing an emotional recognition system is the extraction of an appropriate range of features, regardless of the problems and challenges that still reduce the recognition accuracy rate, such as noise, illumination, talking faces, and head poses that affect the features extraction as in [25],[26],[28], and [30]. In other forms, the emotional gaps are simply the variations between the emotions and features that have been extracted.

1.7 Aims and Objectives

The primary aim is to design a multimodal emotion recognition system that can generate accurate emotions and bridge the “emotional gap” by extracting the optimal sequence of features based on deep learning and transfer learning models and decreasing the time of response to provide an understanding of the state of emotions. The objectives of this dissertation are:

- Extracting visual and audio features separately by using developed deep learning models and training them on a dataset close to the one in which the training is required to improve the detection of the multimodal emotion recognition system.
- Developing deep learning models for visual and audio modalities to classify emotions accurately.
- Evaluating the performance of the proposed system by using two databases.

1.8 Contributions

1. Developing robust deep learning models for visual path based on Transfer Learning as feature extractors with new classifiers and training them on two cases by freezing and unfreezing the feature extraction part to obtain the best model that provides more efficient categorization and more generalization.
2. Proposing a deep learning model for audio path which consists of seven layers.
3. Reducing the training time by initializing and feeding images in the visual path to the main training process.
4. Integrating the decisions of visual and audio paths to improve the system accuracy.
5. The proposed system also suggests a real-time recognition which is more reliable.

1.9 Dissertation Organization

In addition to this introductory chapter, this dissertation contains the following chapters:

Chapter 2: "Theoretical Background".

This chapter provides an overview of the techniques employed in the field of recognizing human emotions, deep learning, convolutional neural network models, and other significant subjects.

Chapter 3: "The Proposed System "

This chapter discusses the proposed multimodal emotion recognition system and covers the methods used by the audio and visual phases, as well as the fusion stage, which combines the results of both of them to produce the final emotion prediction.

Chapter 4: " Results and Discussion"

This chapter presents the results of experimental training and testing for all models in each path, as well as the related performance for each one, databases, and comparisons with other models.

Chapter 5: "Conclusions and Future Works".

This chapter includes a summary of conclusions reached after evaluating the results of carrying out the proposed system. This chapter also includes a list of possible recommendations for future works.

Chapter Two

Theoretical Background

2.1 Overview

This chapter reviews the conventional framework for a multimodal emotion recognition system and discusses some of the ideas employed at each stage. It demonstrates the system in two independent routes, speech and visual paths , each of which processes a distinct kind of input . Each route has its section detailing the fundamental ideas employed in the data pre-processing and feature extraction processes. Standard fusion methods, databases and performance evaluation metrics are also discussed.

2.2 Data Preprocessing

The term preprocessing points to all operations that must be carried out on the facial image and speech signal before extracting attributes. Preprocessing is considered as the first stage of the other stages in any recognition system by manipulating or adjusting the signal to make it more suitable for the feature extraction stage [32].

2.2.1 Speech Normalization

Feature values in a dataset can be transformed to a consistent scale using the statistical technique which is called normalization. The purpose of the normalization process is to allow the comparison of corresponding normalized values for different databases in a way that eliminates the side effects of certain outliers [33]. The following justify the necessity of speech feature normalization:

A. Different persons may speak with different levels of sound loudness.

B. The extracted features have different scales that combine these features to create a feature vector which is not appropriate and will affect the learning process of the classifier.

So, the normalization step is required to compensate for differences among different speech cases. This can be done by subtracting the mean from each feature and then dividing by the standard deviation[33].

2.2.2 Speech-Based Feature Extraction

An important issue in emotion recognition through speech is the extraction of speech features that effectively describe the emotional content of the speech and at the same time do not rely on the speaker's identity or the semantic content of the words within the speech [34]. Although many speech features were investigated in the recognition of speech emotions, till now the researchers have not identified the best speech features for this task. This is due to the similarity in the extracted features for different emotions [35]. Among these features that have been used are:

A. Zero Crossing Rate (ZCR)

Zero-crossing rate is a measure of the number of times that the amplitude of the speech signal crosses through a value of zero within a given interval of time. In the case of unvoiced speech, the signal crosses zero more times than in voiced speech. For silent regions, ZCR is near zero. ZCR can be defined as in Equation 2.1 [36]:

$$Z(n) = \frac{1}{n} \sum\nolimits_{m=-\infty}^{+\infty} |sgn[x(m)] - sgn[x(m-1)]| \; \omega(m-n) \qquad (2.1)$$

where

$$sgn(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

Where n is the number of the frame under processing, the samples of a wave form of the input speech signal are defined as x(m), m is the sample index and $\omega(n)$ usually uses the rectangular window function.

B. Root Mean Square (RMS)

Energy refers to the intensity of the speech signal and where the accent of the voice signal is [37]. Some basic energy attributes can be extracted from the time domain of the N-voiced speech samples (x1..xN). The RMS represents the energy of the speech signal. The energy of a speech reflects the loudness of the speech and it increases with high-arousal emotions while decreasing with low-arousal emotions [38]. It can be defined as in Equation 2.2 [39]:

$$\boldsymbol{RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} \qquad (2.2)$$

C. Mel Frequency Cepstral Coefficients (MFCCs)

The spectral-based Speech Features identified that the emotional state of an utterance affects the distribution of spectral energy throughout the frequency range of the speech signal. For instance, it is found that speech signal with happiness emotion own high energy at the high-frequency range while the signal with sadness emotion owns small energy at the same range [40]. However; the derived spectrum is often passed through a bank of band-pass filters to better leverage the spectral distribution over the audible frequency range. Spectral features are then computed from the outputs of these filters. An example of spectral-based speech features is the MFCCs [38]. MFCCs are compact representations of the spectrum and are typically used to automatically identify speech and it is also used as a primary feature in many research areas that include audio signals. The MFCCs are computed in the following steps [41].

1- Divide the signal into frames.

2. For each frame, obtain the amplitude spectrum.

3. Take the logarithm.

4. Convert to Mel (a perceptually-based) spectrum.

5. Take the discrete cosine transform (DCT).

D. Mel Spectrogram

Mel scale represents a nonlinear modification of the frequency scale, and when combined with a spectrogram, the result is a Mel spectrogram. Here, the audio signal is segmented into frames and processed individually using a Hamming window. To make the transition from the time domain to the frequency domain, the Short-Time Fourier Transform (STFT) is used. The final step in creating the spectrum utilized to create the Mel spectrogram involves the usage of a logarithm. The Mel Spectrogram is computed in the following steps:

1. The audio signal is first broken down into smaller frames and a hamming window is applied on each frame.
2. The DFT is applied to switch from the time domain to the frequency domain.
3. Mel filterbank, and triangular filters are applied to extract the frequency bands.

4. Log() is used at the final stage to generate the spectrum which consists of the spectral envelope and spectral details.

E. Chroma STFT

Twelve different pitch categories are utilized to analyze music, and the Chroma value of an audio file represents their relative intensity. They can be used to tell apart similar-sounding audio streams based on their pitch classes. Chroma features were calculated using STFT, or short-term Fourier transformation. Regarding pitch and signal structure, STFT is a representation of the data. The chroma STFT is computed in the following steps [41].

1. Short-Time Fourier Transform (STFT) of the audio signal. It is calculated by applying the Fourier Transform to short overlapping windows of the signal (see Equation 2.3).

$$X(m,n) = \sum_{k=0}^{N-1} x(n+k).\,\omega(k).\,e^{-j\frac{2\pi km}{N}} \qquad (2.3)$$

where:

- X(m,n)is the STFT at time n and frequency m,
- X(n) is the audio signal.
- W(k) is a window function (Hamming window),
- N is a window size,
- $e^{-j\frac{2\pi km}{N}}$ is the complex exponential.

Calculate the magnitude spectrum from the complex STFT values. This is done by taking the absolute value of the complex numbers obtained from the STFT as shown in Equation 2.4:

$$|X(m,n)| = \sqrt{Re(X(m,n))^2 + Im(X(m,n))^2} \qquad (2.4)$$

where:

- Re(X(m,n)) is the real part of the complex STFT,
- Im(X(m,n)) is the imaginary part of the complex STFT.

2. Apply a filter bank to the magnitude spectrum to obtain the chroma representation. The filter bank is designed to emphasize different pitch classes (see Equation 2.5). Typically, there are 12 filters, each corresponding to one of the 12 pitch classes (C, C#, D, ..., B).

   $$Ci(n)=\sum_{m\in bin(i)|} |X(m,n| \qquad (2.5)$$

   where:

   - Ci(n) is the Chroma value for pitch class i at time n,
   - bin(i) is the set of frequency bins associated with pitch class i.
3. Normalize the Chroma values to make them invariant to overall loudness variations. This is shown in Equation 2.6:

   $$Chroma(i,n) =\frac{Ci(n)}{\sum_{j=1}^{12} Cj(n)} \qquad (2.6)$$

   where:

   Chroma (i,n) is the normalized Chroma value for pitch class i at time n.

2.2.3 Key-Frame selection

The main goal of key-frame selection is to identify a series of representative frames from an image sequence. In general, there are many approaches to selecting keyframes, such as:

A. Motion analysis-based strategy

This technique measures the optical flow with each frame to specify whether or not the expression of the face alters. The key disadvantage is that while it can detect small changes it may miss essential parts, whereas long segments of identical components can appear many times [42].

B. Shot-boundary based strategy

The first, middle, and last frames of the video are used as keyframes in this technique. This strategy is simple and fast, the chosen frames are usually unstable and do not capture the video's key visual content [43].

C. Visual-content based strategy

This technique employs a range of parameters (for example, criteria depend on the shot, color and movement). The first frame is used as the main frame at first, and the new frame is then compared to the other frame depending on the similarities identified by a color histogram. If the current frame's content has changed significantly, it will be used as the main frame [42].

D. Clustering-based strategy.

It aims to build groups of frames with identical attitudes. Every frame is assigned to a particular cluster, and the frames closest to the center of each cluster are selected as key-frames [43].

2.3 Deep Learning

Deep learning (DL) has recently risen to prominence in the computing world as a subfield of ML and AI with impressive data-driven learning capabilities. DL technology, which has its roots in the ANN, is now being used in many fields, including healthcare, visual recognition, text analytics, cybersecurity, and many more, and many more [44].

Deep learning algorithms can use fast computing power and large data sets to discover previously unknown relationships within the data and provide predictions. Rather than a single method, it is a collection of algorithms and structures that may be used for various issues. The popularity of this method, which employs deep neural networks, has increased alongside the availability of more robust computational resources. Its ability to analyze many features gives it an edge when working with unstructured data [45].Deep learning has made its way out of academia and into many high-tech sectors as the leading machine learning technique [46].

Deep learning algorithms pass the data through several layers. Each layer is capable of extracting features progressively and passing it to the next layer. Initial layers extract low-level features, and succeeding layers combine features to form a complete representation [45] .

2.3.1 Deep Learning Architectures

The advent of Deep Learning has had far-reaching effects on computer vision in recent years, allowing for unprecedented efficiency in areas such as image classification, object localization, tracking, detection, posture estimation, captioning, and segmentation [47].

Although deep learning architectures require more training time, they ultimately outperform ANNs. However, training time can be cut in half with techniques like transfer learning and Graphical Processing Units (GPU) calculation. One of the factors that defines the success of neural networks is the precision with which their network architecture is designed. Pre-trained Unsupervised Networks (PUNs), Recurrent/Recursive Neural Networks (RNNs), and Convolutional Neural Networks (CNNs) are the three main types of deep learning approaches [48]. There are vast and varied architectures and methods that have been used in deep learning. RNNs and CNNs are two of the earliest methods that are still frequently used today across various fields and industries.

I. Convolutional Neural Networks (CNNs)

CNNs are a type of deep learning network used for processing in a grid pattern, with the most common application being the evaluation of visual data [49]. It has been built to mechanically discover spatial hierarchies of features, from low-level to high-level patterns. Using convolutional layers instead of fully connected ones may lower the number of parameters and speed up the computation time for two main reasons: (1) sharing of parameters and (2) sparsity of connections. This reduces the number of parameters and speeds up the computation. In comparison to standard image classification techniques, which need hand-engineered preprocessing filters, CNN requires very little data preprocessing [50]. CNN has three main features namely, local receptive field, weight sharing and pooling.

1. Local Receptive Field

   Each hidden-layer neuron in a convolutional neural network (CNN) communicates with a tiny area of the input layer, and each connection has a parameter weight and offset that can be learned. This area is called the local receptive field of the hidden layer neuron. Each neuron corresponds to a local receptive field.

2. Weight Sharing

   For a local receptive field with 25 pixels, there is a weight of 5 × 5 for each neuron in the hidden layer. Weight sharing is when the weights corresponding to these neurons in the hidden layer are the same. Due to the existence of weight sharing, the amount of network parameters and training time are greatly reduced.

3. Pooling

The pooling layer is generally behind the convolution layer, the purpose is to compress the image after the convolution to reduce the number of parameters [51].

A typical convolutional neural network (CNN) has three distinct layers: convolution, pooling, and fully connected. The first two, convolution and pooling layers, perform feature extraction. Features are extracted by the first two layers (convolution and pooling) and then mapped to the output by the third layer (a fully connected one). A CNN's filters slide across an image using convolution to discover interesting patterns, and usually, its activation function is ReLU, which provides a feature map that adds to the next layer [52]. After this, further layers such as pooling, normalization, and fully connected may be added. CNNs use a series of mathematical operations, including convolution which is a special kind of linear operation. A convolution layer plays a key role in CNN, which is composed of a stack of mathematical operations, such as convolution, a specialized type of linear operation. Pooling is a non-linear down-sampling technique where the pooling layer merges non-overlapping areas from one layer to generate a single value that is transferred to the next layer. Feature maps can evolve hierarchically and progressively and be more complicated as the output from one layer feeds to the next. Training is the process of adjusting parameters like kernels (filters) to reduce the distinction between ground truth labels and outputs using optimization methods such as gradient descent, and backpropagation. Forward propagation can be defined as a method that transforms input data into output data across layers. Figure 2.1 depicts the typical architecture of a conventional CNN and training [53].

In general, the initial value of the convolution kernel is randomly generated. In the process of network training, new weights are constantly learned and updated in real-time until a reasonable weight is finally learned. Downsampling, also called pooling, is a special convolution process. As the image data goes through the CNN layers, it begins to detect bigger components or forms of the item, eventually identifying the desired object [53].

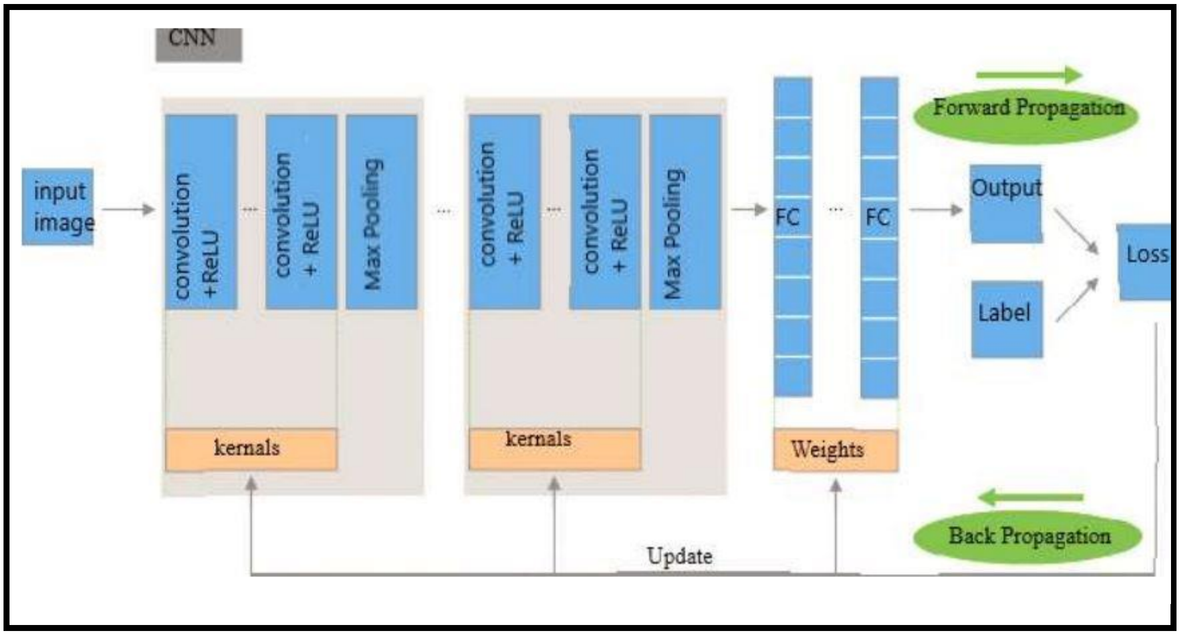


Figure 2.1: The architecture of CNN and the training process [53]

The component of a classic CNN :

A. Convolutional Layer

Convolutional layers are the most hidden layers in CNN. At present, the structure of CNN has gradually tended to stack several convolutional layers in succession, followed by a pooling layer. The input images and the filters are convolved using convolution. The original image's information is improved, and noise is reduced. Features of the local receptive field, as well as weight sharing, are reflected in the convolution process. In convolution, the network will automatically learn the features without manual selection of features, which avoids time and effort. Figure 2.2 shows how the convolution process works [51].

The image is reflected in the feature detector, a two-dimensional (2-D) weighted array. The dimensions of the receptive field are defined by the filter size, which is typically a 3x3 matrix but can vary in size. The filter is then applied to a subset of images, and a dot product of the input pixels and the filtered images is calculated. After that, the filter iteratively sends out the dot product, one step at a time. This process is repeated until the kernels have covered all of the images. Dot products between the input and the filter yield an activation map, feature map, or convolved feature [53].

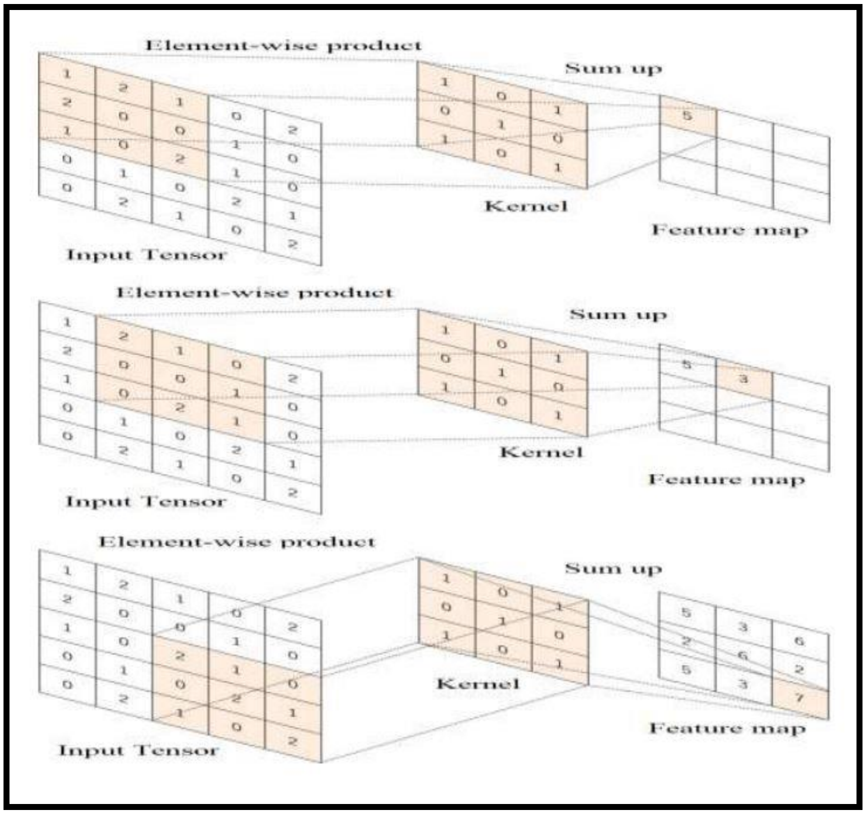


Figure 2.2: The convolution process with filter size 3*3 [51]

B. Pooling Layer

Pooling images may substantially eliminate a lot of calculations while maintaining the image's key features since the image's local features are connected. A pooling layer performs a standard down-sampling operation and reduces the dimension of the feature maps to provide translation invariance to minor distortions and shifts. It also reduced the learnable parameters.

When the image's size is 4 * 4, and the convolution kernel with a size of 2*2 is utilized, the convolution kernel's sliding stride is set to two. Figure 2.3 depicts the most popular pooling methods: average pooling, random pooling, and maximum pooling [52], [53].

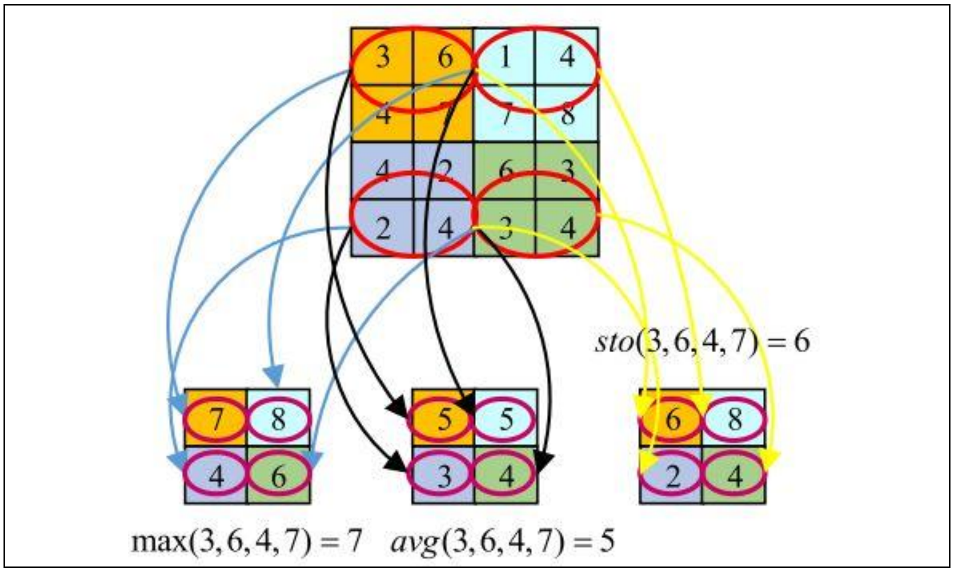


Figure 2.3: The process of the pooling operation [53]

C. Active Layer

The result of a linear operation, such as convolution, is then subjected to a nonlinear activation function. For a neural network to be able to tackle more complex tasks, it needs access to nonlinear variables, which are provided by the activation function. Since the hyperbolic tangent tanh and the sigmoid function are mathematical representations of actual neuron activity, they were formerly used. However, the rectified linear unit ReLU is now the most widely used nonlinear function. Equation 2.7 provides a mathematical description of the ReLU function. The appropriate graphs for these functions are shown in Figure 2.4 [54].

$$F(x) = \max(\text{zero}, x) \tag{2.7}$$

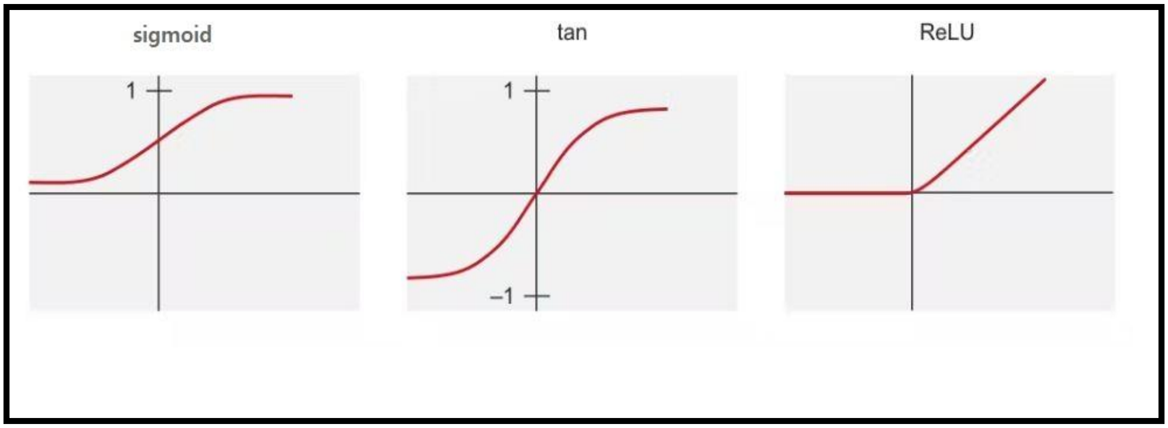


Figure 2.4: Activation Functions [53]

D.Fully Connected Layer

The fully connected layer is the last few layers in the CNN and acts as a classifier in the entire network. The output feature maps of the final convolution or pooling layer are typically flattened, i.e., transformed into a one-dimensional (1D) array of numbers (or vector), and connected to one or more fully connected layers, also known as dense layers, in which every input is connected to every output by a learnable weight. input is connected to every output by a learnable weight. Once the features are extracted by the convolution layers and downsampled by the pooling layers, they are mapped by a subset of fully connected layers to the final outputs of the network, such as the probabilities for each class in classification tasks. The final fully connected layer typically has the same number of output nodes as the number of classes [53].

A unique activation function is typically used for the final fully connected layer. The Softmax function is used as an activation function for the multiclass classification task, and it converts the actual numbers at the end of the fully connected layer into probabilities for the target classes (where the possibilities range from 0 to 1) [55].

II. Three types of the CNN architectures

Because of the advent of new databases, like CIFAR-10 and MNIST, and competitions, including ImageNet which since 2010 has been used in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a benchmark in image classification and object detection, has a variety of alternative CNN architectures have been developed. The different types of CNN architectures include AlexNet, VGGNet, ResNet, EfficientNet, Mobilenet and so on [53].

A.Residual CNNs (ResNet)

The 152-layer ResNet architecture won the ILSVRC2015 competition. Its vital contribution was the use of batch normalization and unique skip connections for training deeper architectures. It is possible to prepare a ResNet with 1000 layers using those methods. Although ResNet is designed to run serially over the whole network length, empirical evidence suggests that it more commonly acts on blocks of relatively low depth (20-30 layers), which work in parallel. Figure 2.5 depicts the concept of skip connection used to avoid passing input to subsequent levels, as in ResNet [56].

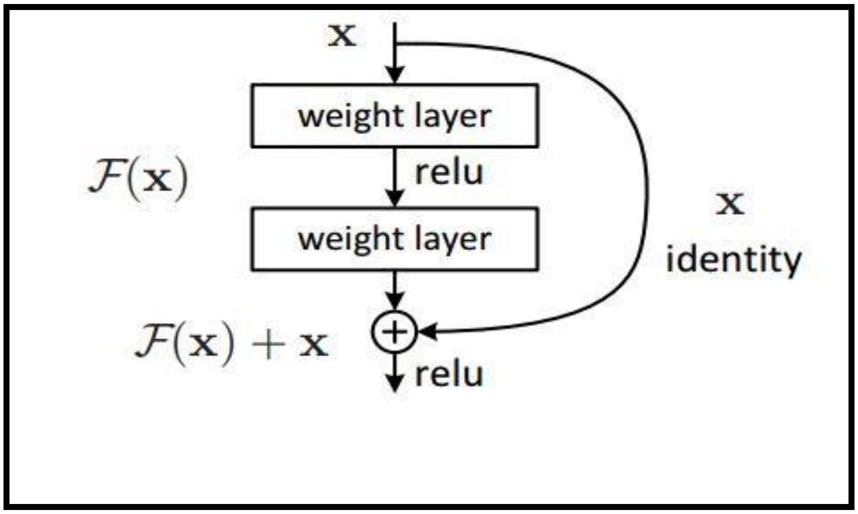

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.5: Structure of residual block [56]

The VGG-19 model, plain network with 34-parameter layers and residual network with 34-parameter layers are examples of network architectures for ImageNet plotted as shown in Figure 2.6 [56].

By replacing each 2-layer block in the 34-layer net with this 3-layer bottleneckblock, resulting in a 50-layer ResNet or ResNet50 as in Figure 2.7 [57].

B.MobileNets

To effectively maximize accuracy while being mindful of the restricted resources for an on-device or embedded application, MobileNet (Efficient Convolutional Neural Networks for Mobile Vision Applications) is an architecture that focuses on making deep learning networks minimal and having low latency. They can be used as building blocks for more complex tasks like categorization, detection, and embeddings, much like prominent large-scale models like Inception [58].

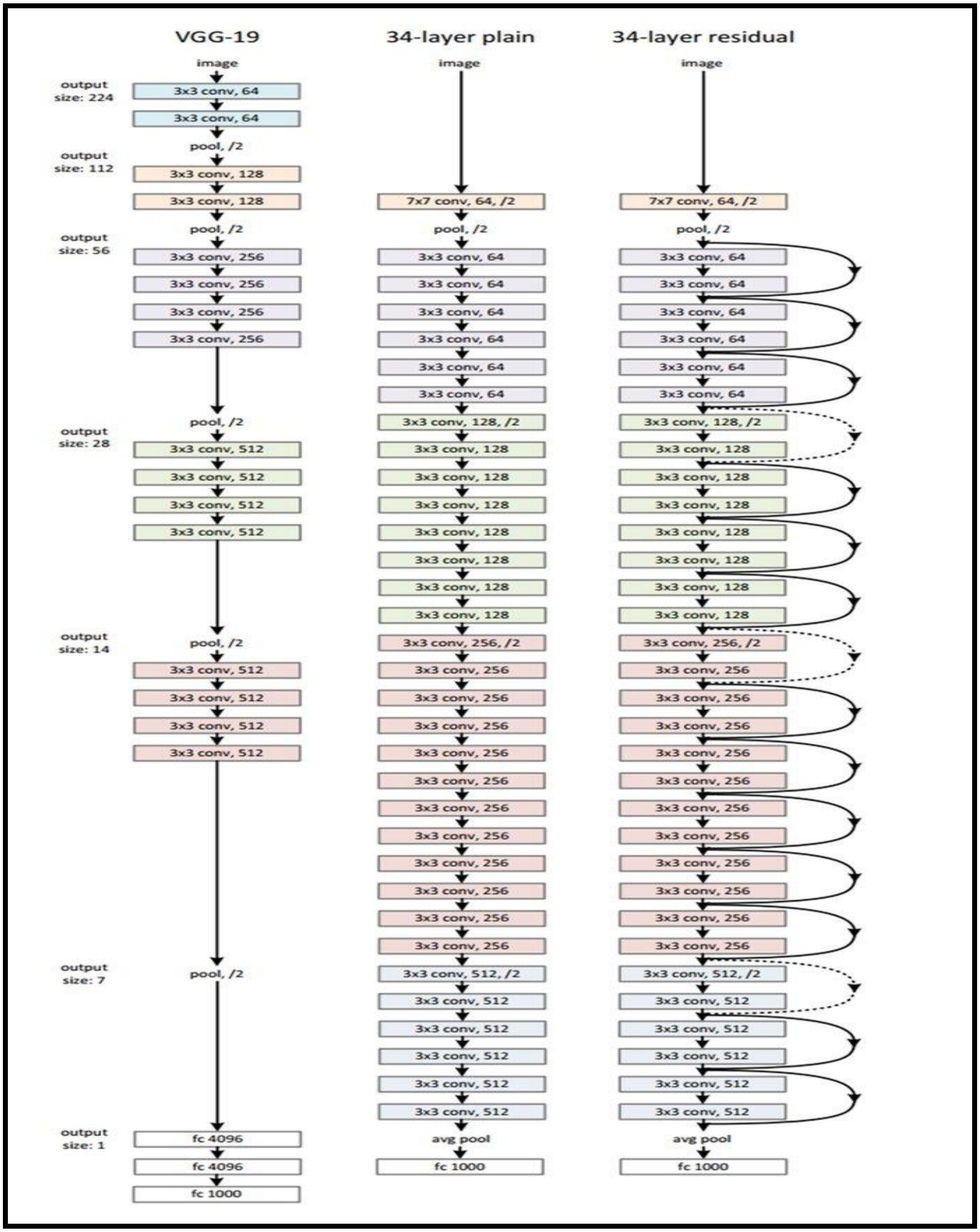


Figure 2.6: Example network architectures for ImageNet. Left: the VGG-19 model. Middle: a plain network with 34-parameter layers Right: a residual network with 34-parameter layers [56]

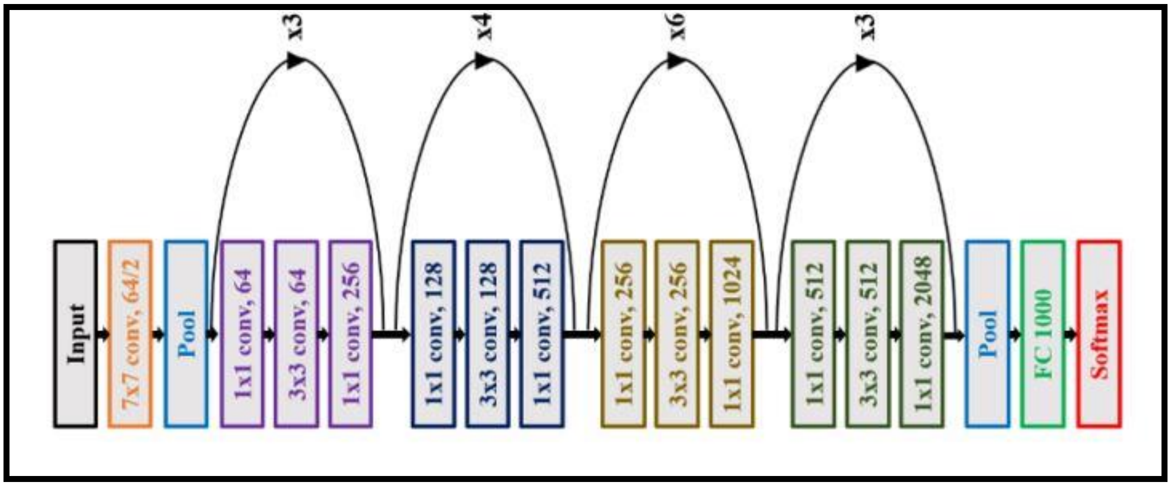


Figure 2.7: The ResNet50 architecture [57]

The MobileNet models can be easily deployed on mobile and embedded edge devices. MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build lightweight deep neural networks. The MobileNet uses a Depthwise separable convolution instead of the usual convolution to reduce the computation time and parameters. It works by applying convolution to each channel of the image instead of as a block n time and then using 1x1 convolution to get n filters. So, it is based on depthwise separable convolutions which is a form of factorized convolutions that factorize a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. Mobilenet is lightweight in its architecture as it uses depthwise separable convolutions which basically means it performs a single convolution on each color channel rather than combining all three and flattening it. This has the effect of filtering the input channels. As explained clearly in [58], the depthwise convolution used by MobileNets only uses a single filter for all input channels. The outputs of the depthwise convolution are then combined using a 1*1 convolution in the pointwise convolution. In a straightforward operation, the inputs of a typical convolution are filtered and combined to produce a new set of outputs. This is separated into filtering and blending layers by the depthwise separable convolution. The size of both the calculation and the model can be considerably decreased because of this factorization.

a) Simple Convolutions

This can occur when the input data is with size PxPxM, where PxP is the image width and image height and M is the number of channels. Moreover, it is supposed that there are N kernels of size KxKxM. After the convolution operation is done, the output size will be RxRxN where R is the output image width and height of the image [59]. Figure 2.8 shows the simple convolutions.

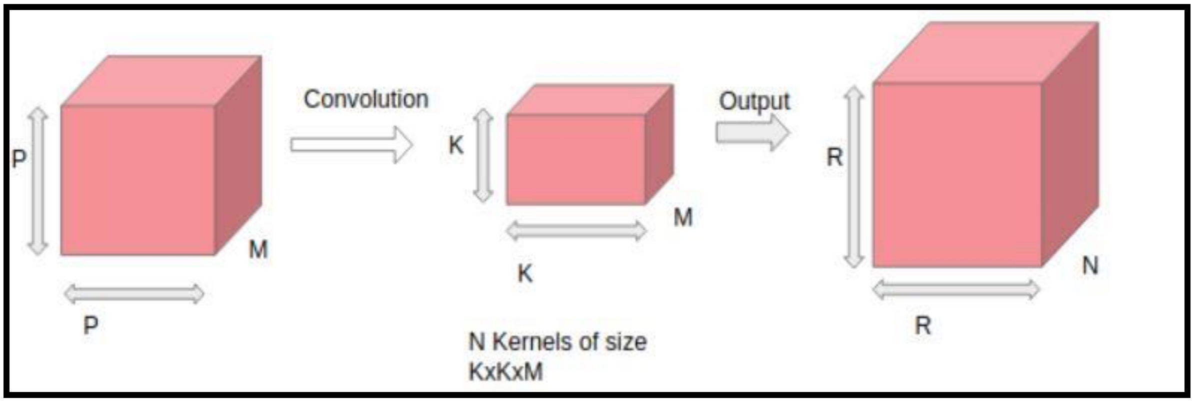


Figure 2.8: Simple Convolutions [59]

The number of the performed operations per convolution will be KxKxM, which is the size of the filter itself. Also, the total number of operations is (Number of operations per convolution) * (RxRxN) = (KxKxM)*(RxRxN).

b) Depth Wise Separable Convolutions

It has two major components. The first one is Depth-wise convolution and the second one is Point-wise convolution.

- Depth-wise convolution

  In this type, convolution is applied to a single channel at a time not like the simple convolutions in which it is done for all the channels together. So here for each convolution, the kernel used is of size K x K x 1. If the number of channels in the input image is M, then M such kernels will be used. The output size will be of size R x R x M. A single convolution requires KxK operations. Figure 2.9 indicates the depth-wise convolution.

- Point-wise convolution

  In point-wise operation, a 1 × 1 convolution operation is applied on the Mchannels. So the filter size for this operation will be 1 x 1 x M. Say we use N such filters, the output size becomes R x R x N. Figure 2.10 explains the point-wise convolution [59].

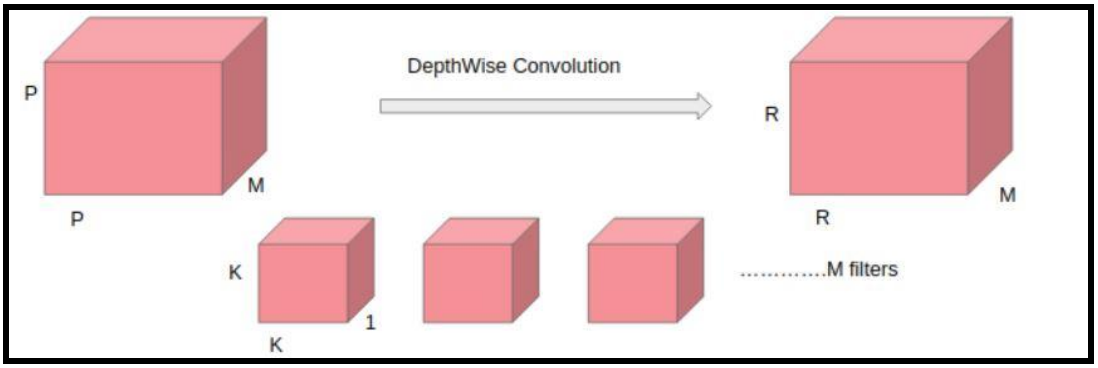


Figure 2.9: Depth-wise convolution [59]

A single point-wise convolution requires 1xM operations since the filter is sliding RxR times, the total number of multiplications required is (1*M) *(RxR) = MxRxRxN.

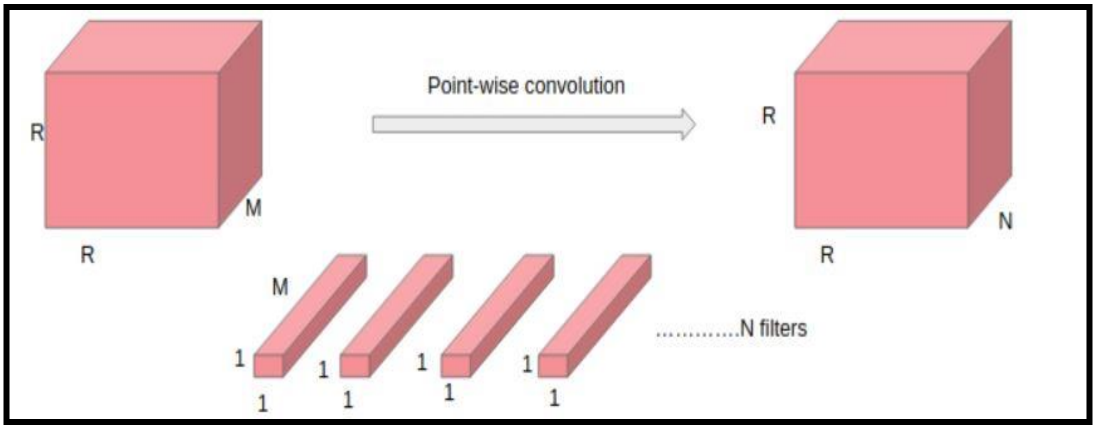
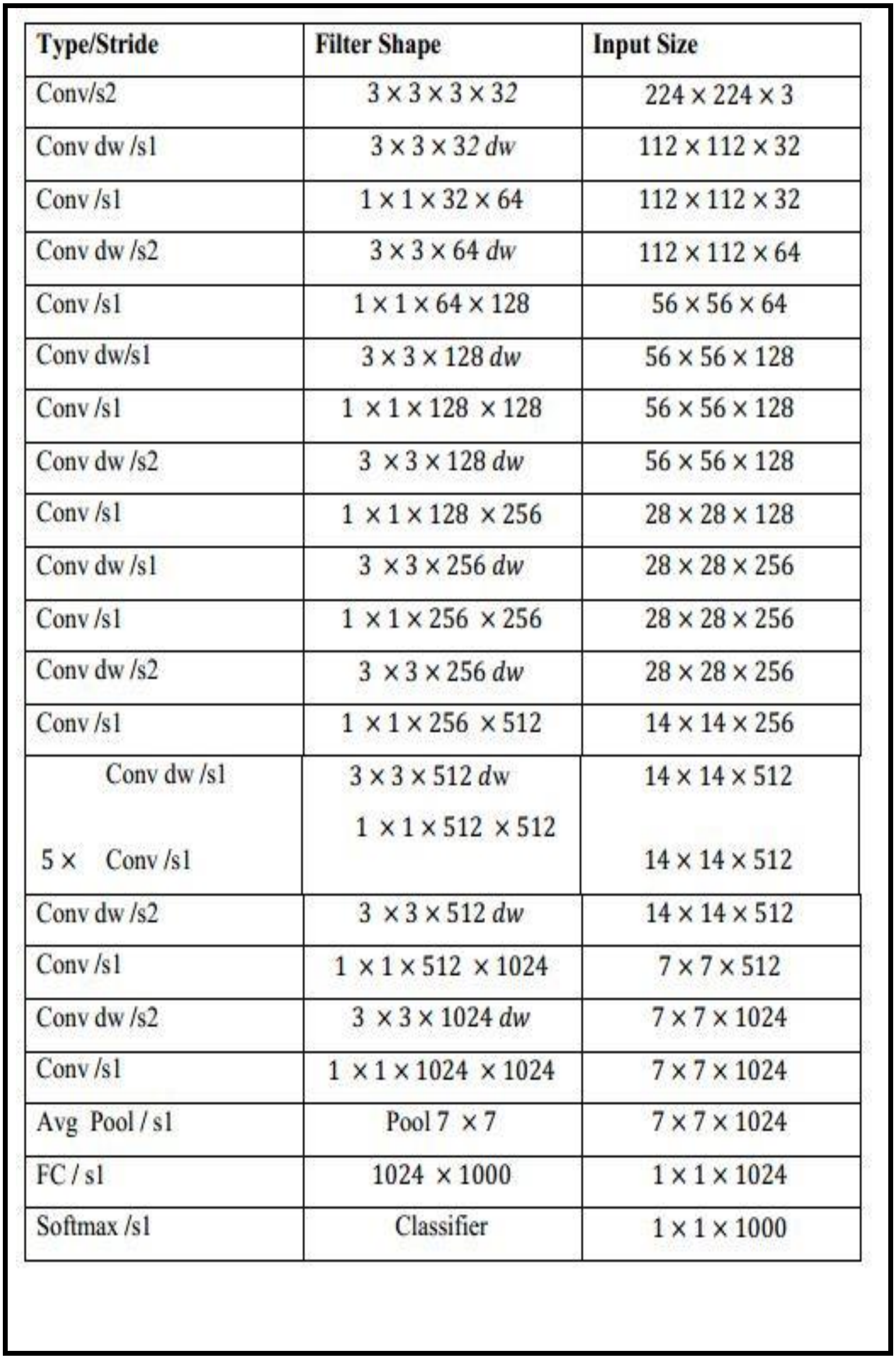


Figure 2.10: Point-wise convolution [59]

The MobileNet architecture is illustrated in Figure 2.11. All layers are followed by a batch normalization [60] and ReLU nonlinearity except the final fully connected layer which has no nonlinearity and feeds into a softmax layerfor classification. MobileNet starts with a basic 2D convolution layer.Then there is a series of convolution layers called Depthwise Separable attached one after another, having different strides and filter counts [59].

By defining the convolutional block, each convolutional block after the input has the following sequence: BatchNormalization, followed by ReLU activation and then passed to the next block.The first convolution block has 32 filters of kernel size (3x3) and a stride of 2. It is followed by a Batch Normalization layer and a ReLU activation. Then comes the main ingredient block of the MobileNet architecture - the Depthwise Separable convolution layer. This process is done in 2 steps: Depthwise convolution and then Pointwise convolution [58].Figure 2.12 shows the depthwise and pointwise Convolution visualization.



Type/Stride	Filter Shape	Input Size
Conv/s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw /s1	3 × 3 × 32 *dw*	112 × 112 × 32
Conv /s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw /s2	3 × 3 × 64 *dw*	112 × 112 × 64
Conv /s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw/s1	3 × 3 × 128 *dw*	56 × 56 × 128
Conv /s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw /s2	3 × 3 × 128 *dw*	56 × 56 × 128
Conv /s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw /s1	3 × 3 × 256 *dw*	28 × 28 × 256
Conv /s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw /s2	3 × 3 × 256 *dw*	28 × 28 × 256
Conv /s1	1 × 1 × 256 × 512	14 × 14 × 256
Conv dw /s1 5 × Conv /s1	3 × 3 × 512 *dw* 1 × 1 × 512 × 512	14 × 14 × 512 14 × 14 × 512
Conv dw /s2	3 × 3 × 512 *dw*	14 × 14 × 512
Conv /s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw /s2	3 × 3 × 1024 *dw*	7 × 7 × 1024
Conv /s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax /s1	Classifier	1 × 1 × 1000

Figure 2.11: MobileNet architecture [58]

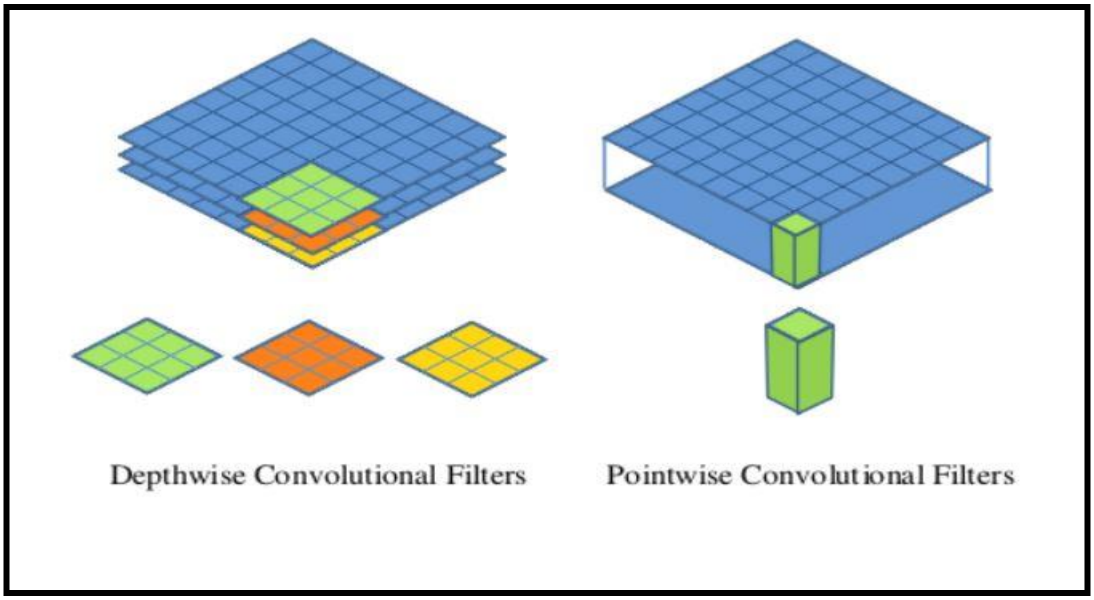
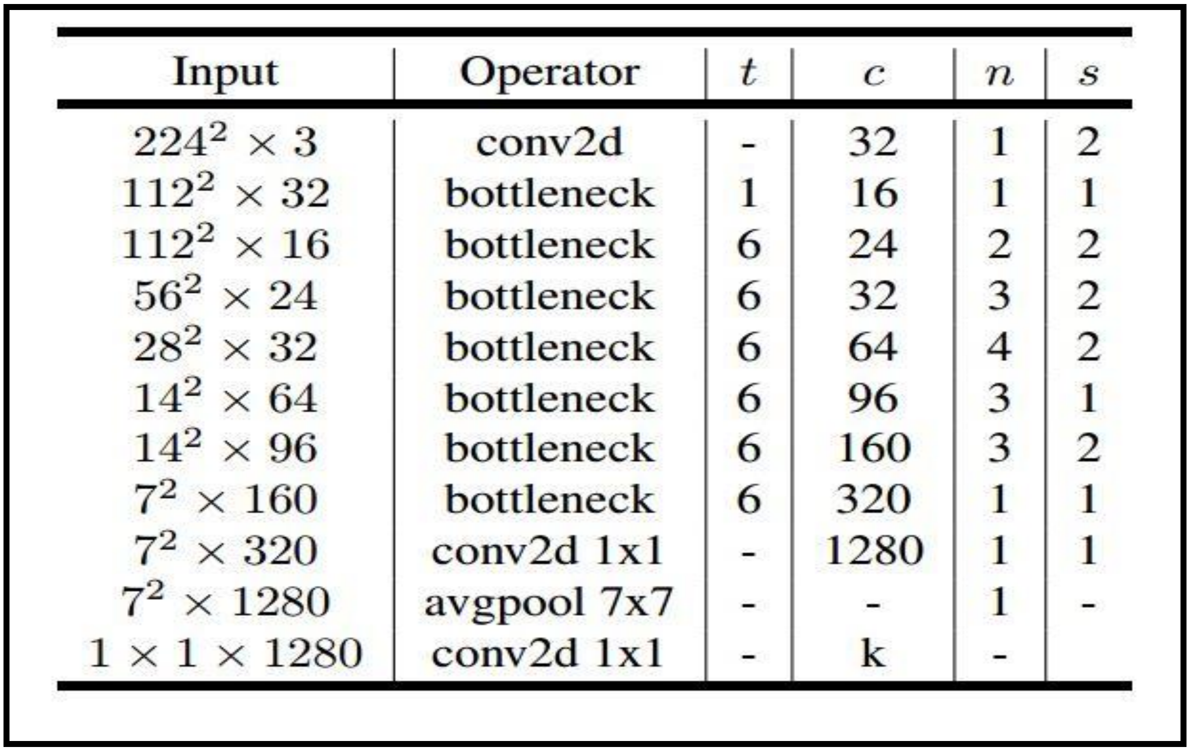

Depthwise Convolutional Filters
Pointwise Convolutional Filters

Figure 2.12: Depthwise and pointwise Convolution visualization [58]

MobileNetV2 [61] is very similar to the original MobileNet, except that it uses inverted residual blocks with bottlenecking features. It has a drastically lower parameter counts than the original MobileNet. It is a significant improvement over MobileNetV1 and pushes the state of the art for mobile visual recognition including classification, object detection, and semantic segmentation.

MobileNetV2 is an improvement upon MobileNetV1 [58], that makes use of depthwise separable convolution as a core technological component. Linear bottlenecks between layers and shortcut connections between them are two new additions to the architecture in V2. A bottleneck depth-separable convolution with residuals is the fundamental building piece. In Figure 2.13, we can see the in-depth composition of this block. MobileNetV2's architecture is made up of 32-filter fully convolution and 19-filter residual bottleneck layers.



Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Figure 2.13: MobileNetV2 structure [61]

Each line in Figure 2.13 describes a sequence of 1 or more identical (modulo stride) layers, repeated n times. All layers in the same sequence have the same number c of output channels. The first layer of each sequence has a stride s and all others use stride 1. All spatial convolutions use 3 × 3 kernels. The expansion factor t is always applied to the input size that controls the number of channels in the output of the first convolutional layer in a residual bottleneck block. It is typically set to 6, this means that the output of the first convolutional layer will have 6 times as many channels as the input [61].

The MobileNetV2 models are much faster in comparison to MobileNetV1. It uses 2 times fewer operations, and has higher accuracy. The principle of MobileNetV2 is to have blocked as in residual networks where the input of a block is added to its output. But instead of the usual convolutional layers in the block, they use depthwise convolutions.

Moreover, the number of channels at the input and the output of the blocks (in the trunk) is much smaller than the number of channels for the depthwise convolutions in the block [62].

MobileNetV2 architecture has 53 convolution layers and one AvgPool. It has two main components:

- ✓ Inverted Residual Block
- ✓ Bottleneck Residual Block

There are two types of Convolution layers in MobileNetV2 architecture:

- ✓ 1x1 Convolution
- ✓ 3x3 Depthwise Convolution

There are Stride one Blocks and Stride two Blocks. The internal components of the two blocks are shown in Figure 2.14 [61].

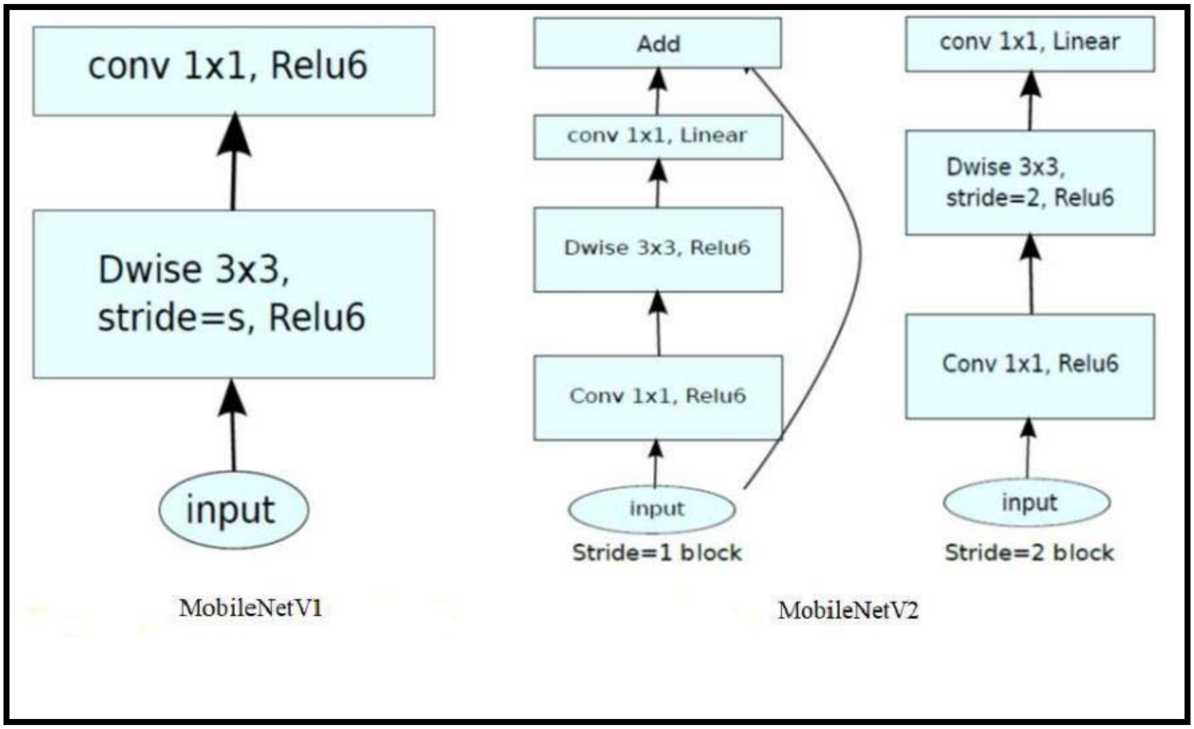


Figure 2.14: Convolutional blocks for MobileNetV1, MobileNetV2 architectures [61]

C.EfficientNet

There are three scaling dimensions of a CNN: depth, width, and resolution. Depth simply means how deep the network is which is equivalent to the number of layers. Width simply means how wide the network is. One measure of width, for example, is the number of channels in a Conv layer whereas Resolution is simply the image resolution that is being passed to a CNN. Figure 2.15 illustrates a clear idea of what scaling means across different dimensions [63].

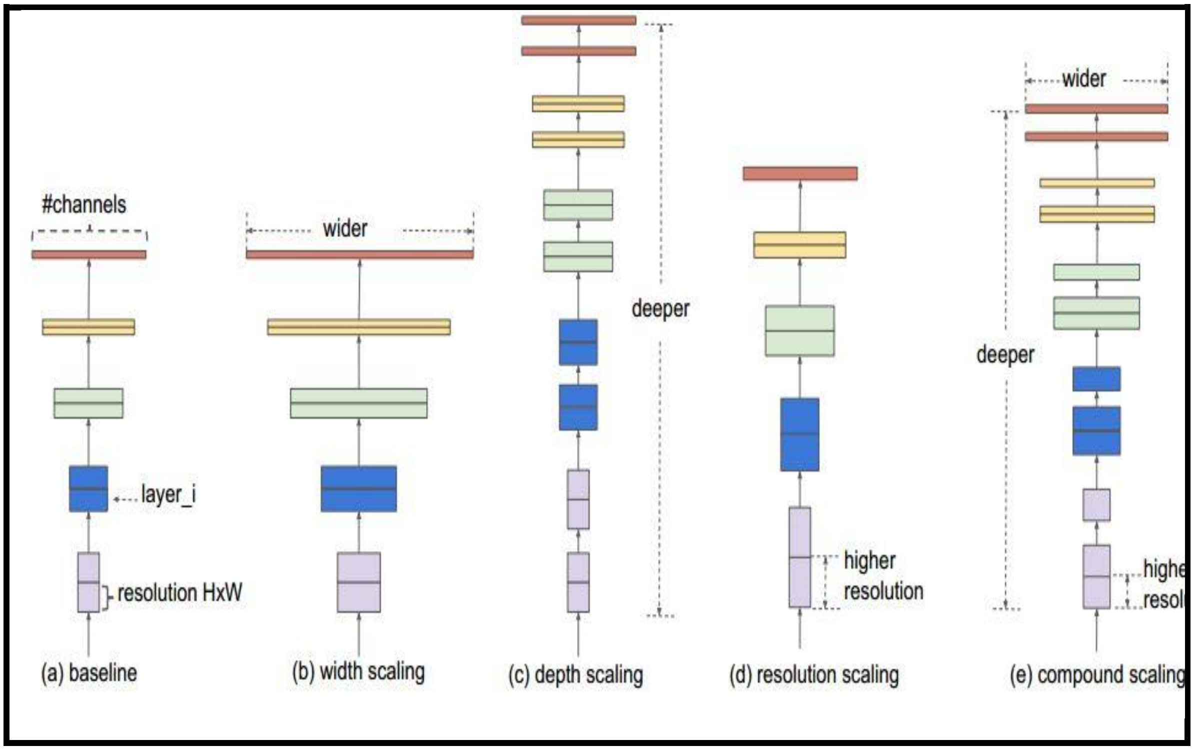
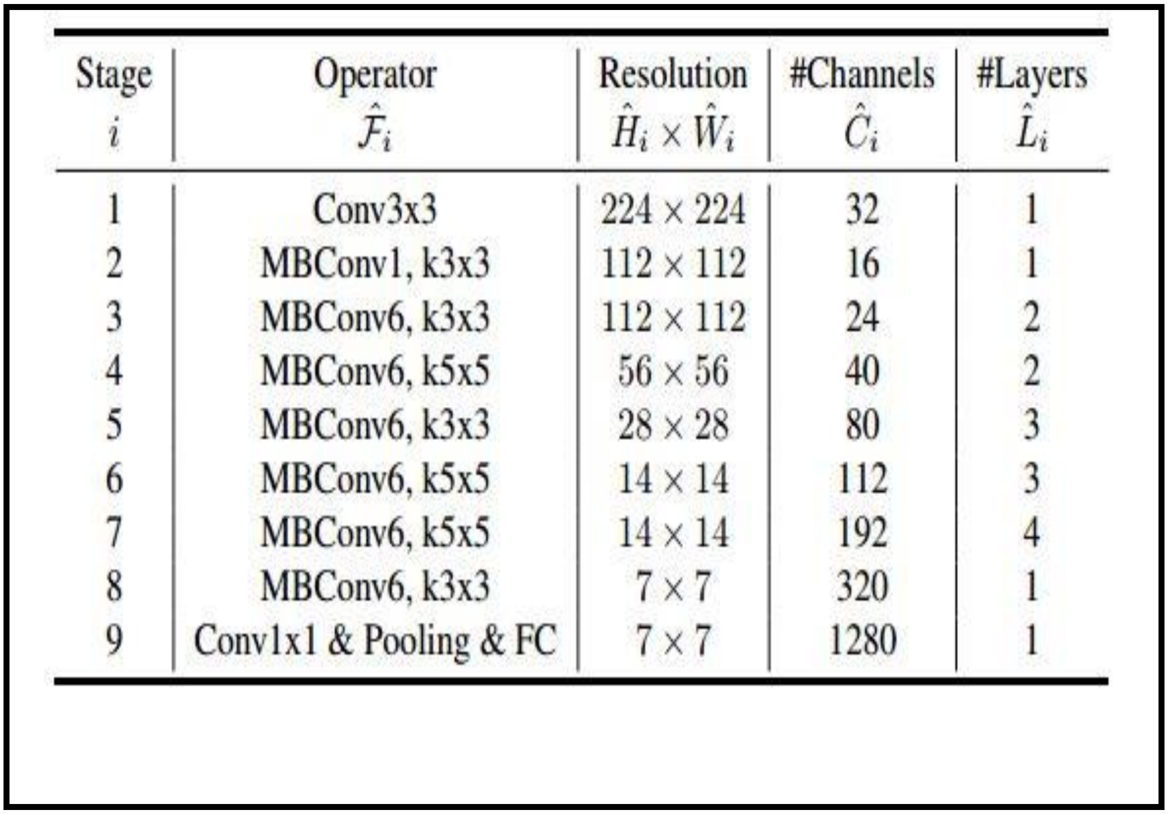


Figure 2.15: Model Scaling. (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) compound scaling method that uniformly scales all three dimensions with a fixed ratio [63]

As the resolution of the images is increased, the depth and width of the network should be increased as well. As the depth is increased, the larger receptive fields can capture similar features that include more pixels in an image. In addition, as the width is increased, more fine-grained features will be captured. It is critical to balance all dimensions of a network (width, depth, and resolution) during CNN's scaling for improved accuracy and efficiency. The conventional practice for model scaling is to arbitrarily increase the CNN depth or width or use a larger input image resolution for training and evaluation. While these methods do improve accuracy, they usually require tedious manual tuning, and still often yield suboptimal performance.

A model EfficientNet has proposed a scaling method that uses a simple yet highly effective compound coefficient to scale up CNNs in a more structured manner. Unlike conventional approaches that arbitrarily scale network dimensions, such as width, depth, and resolution, uniformly scales each dimension with a fixed set of scaling coefficients. Powered by this novel scaling method, a family of models called EfficientNets has been developed which super pass state-of-the-art accuracy with up to 10x better efficiency (smaller and faster) [63].

The resulting architecture uses mobile inverted bottleneck convolution (MBConv), similar to MobileNetV2, but is slightly larger. Then scale up the baseline network to obtain a family of models, called EfficientNets. Figure 2.16 shows the network layers/blocks and the architecture for baseline network EfficientNet-B0 is shown in Figure 2.17.



Stage	Operator	Resolution	#Channels	#Layers
i	$\hat{\mathcal{F}}_i$	$\hat{H}_i \times \hat{W}_i$	$\hat{C}_i$	$\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Figure 2.16: EfficientNet-B0 baseline network [63]

The MBConv block is nothing fancy but an Inverted Residual Block (used in MobileNetV2) with a Squeeze and Excite block injected sometimes [64].

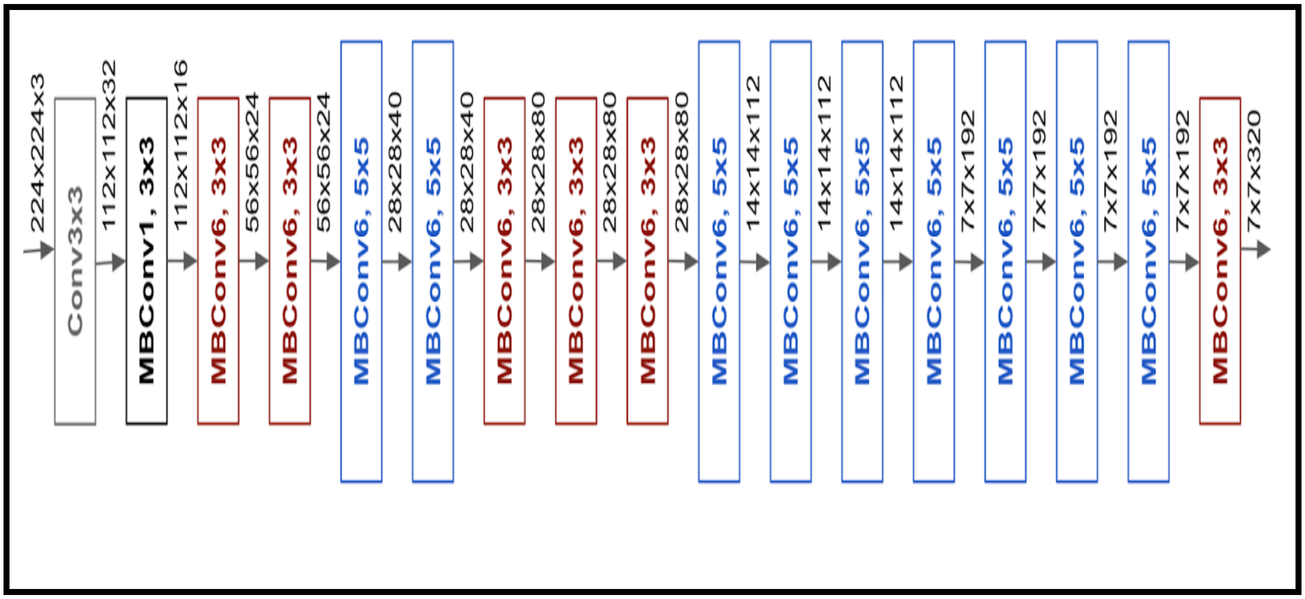


Figure 2.17: The architecture for baseline network EfficientNet-B0 [65]

2.3.2 Deep Learning Techniques

Many research subfields attempt to apply deep convolutional networks to Computer Vision problems to raise the bar on existing benchmarks. One of the most significant difficulties is enhancing these models' generalizability. In machine learning, generalizability is measured by comparing a model's results on familiar data (the training set) to those on novel data (the testing set). Overfitting the training data leads to models with low generalizability. Overfitting is depicted in Figure 2.18 by plotting these accuracies against the number of training iterations. Validation error needs to continue as training error goes down for Deep Learning models to be useful. The plot on the left shows an inflection point where the validation error starts to increase as the training rate continues to decrease. The increased training has caused the model to overfit the training data and perform poorly on the testing set relative to the training set. In contrast, the plot on the right shows a model with the desired relationship between training and testing error [66].

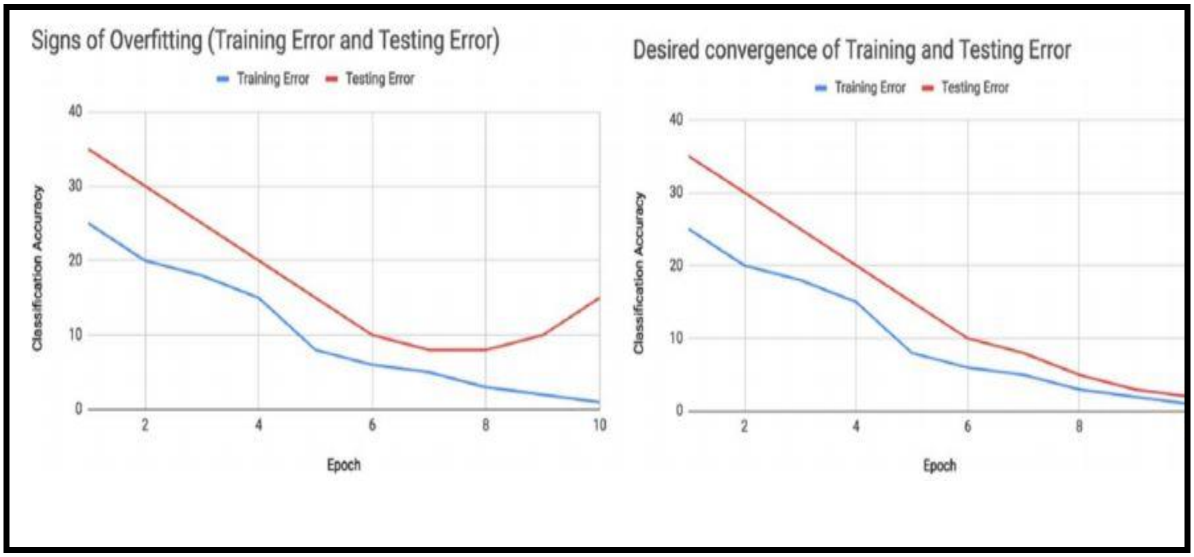


Figure 2.18: Overfitting and desired convergence of training [66]

Some of the powerful techniques that can be applied to deep learning algorithms to reduce overfitting, reduce the training time and to optimize the model are discussed in this section.

- **Backpropagation**: While solving an optimization problem using a gradient-based method, backpropagation can be used to calculate the gradient of the function for each iteration [67].
- **Stochastic Gradient Descent (SGD)**: This is an iterative optimization algorithm that is used to minimize a loss function. It works by repeatedly updating the model parameters in the direction of the negative gradient of the loss function. Using the convex function in gradient descent algorithms ensures finding an optimal minimum of the loss function without getting trapped in a local minimum. Depending upon the values of the function and learning rate or step size, it may arrive at the optimum value in different paths and manners [68].
- **Learning Rate Decay**: The widely used technique is to reduce the learning rate gradually, in which large changes can be made at the beginning and then reduce the learning rate gradually in the training process. Adjusting the learning rate increases the performance and reduces the training time of stochastic gradient descent algorithms. This allows for finetuning the weights in the later stages [69].
- **Dropout**: The overfitting problem in deep neural networks can be addressed using the dropout technique. This method is applied by randomly dropping units and their connections during training. Dropout over an effective regularization method to reduce overfitting and improve generalization error. Dropout gives an improved performance on

supervised learning tasks in computer vision, computational biology, document classification, and speech recognition [70].

- **Max-Pooling**: In max-pooling a filter is predefined, and this filter is then applied across the non-overlapping sub-regions of the input taking the max of the values contained in the window as the output. Dimensionality, as well as the computational cost of learning several parameters, can be reduced using max-pooling [45].
- **Batch Normalization**: Batch normalization reduces covariate shift, thereby accelerating the deep neural network. It normalizes the inputs to a layer, for each mini-batch when the weights are updated during the training. Normalization stabilizes learning and reduces the training epochs. The stability of a neural network can be increased by normalizing the output from the previous activation layer [60].
- **Transfer learning (TL):** In transfer learning, a model trained on a particular task is exploited on another related task. The knowledge obtained while solving a particular problem can be transferred to another network, which is to be trained on a related problem. This allows for rapid progress and enhances performance while solving the second problem [53].

DCNN features numerous hidden convolutional layers, and it works with images of large size, this makes inputs and training extremely difficult. Every DCNN model has distinct important layouts and interconnections [71].

Due to many parameters, training a large DCNN model is difficult undertaking. A huge network frequently necessitates a significant amount of training data. Due to the high cost of data collecting and costly annotation in some disciplines, such as bioinformatics and robotics, building a large-scale well-annotated dataset is extremely challenging, limiting its evolution. However, transfer learning TL[72], which is focused on knowledge transfer between domains, is a potential machine learning strategy for overcoming the above difficulty, and the researchers have demonstrated that TL [73][74] can even be highly effective in solving such issues. Deep learning has recently gotten a lot of interest from researchers as a modern categorization platform, and it has been effectively implemented in many different fields. Transfer learning is a useful approach for dealing with limited training examples. Instead of learning from the beginning, the model might begin utilizing the pre-trained weights [71].

It attempts to transfer the knowledge from the source to the target domain by loosening the requirement that the training set is independently and identically distributed as the test data. This will have a significant positive impact on a variety of areas that are hard to enhance due to a lack of training examples [75].

TL is a strategy that enables employing representations of knowledge obtained through a variety of tasks that have similar applicability. It has been observed that the TL performs better if the two tasks are identical. More recently, it has been explored on tasks that are not related to its training, also it has shown to be effective [73].

- **Fine-Tuning**: Its technique for training machine learning models is commonly used in deep learning that involves taking a pre-trained model and retraining it on a new dataset. Typically, the last fully connected layers, which can be viewed as classification layers, are reset and a smaller learning rate is applied to the pre-trained layers. By doing so, the goal is to adapt the features to the new dataset [76].

- **Data Augmentation**: is a technique that has been developed to reduce overfitting, thus increasing the generalization performance of the model. Data Augmentation is a useful oversampling solution to the problem of class imbalance. This is done under the assumption that more information can be extracted from the original dataset through augmentations. These augmentations artificially inflate the training dataset size by either data warping or oversampling [66].

  1- **Image Augmentation** is the process of generating new images for training our deep learning model. These new images are generated using the existing training images and hence there don't need to collect them manually. Different techniques can be used for image augmentation to feed input to the model such as Spatial augmentation (Scaling, Cropping, Flipping, Rotation, Translation) and Pixel augmentation (Brightness, Contrast, Saturation, Hue).

  - **Flipping**: Flipping Horizontal axis flipping is much more common than flipping the vertical axis.

  - **Cropping**: Cropping images can be used as a practical processing step for image data with mixed height and width dimensions by cropping a central patch of each image. Additionally, random cropping can also be used to provide an effect very similar to translations. The contrast between random cropping and translations is that cropping will reduce the size of the input such as (256,256) → (224, 224), whereas translations preserve the spatial dimensions of the image.

  - **Rotation:** Rotation augmentations are done by rotating the image right or left on an axis between 1° and 359°. The safety of rotation augmentations is heavily determined by the rotation degree parameter.

-Translation: Shifting images left, right, up, or down can be a very useful transformation to avoid positional bias in the data. For example, if all the images in a dataset are centered, which is common in face recognition datasets, this would require the model to be tested on perfectly centered images as well. As the original image is translated in a direction, the remaining space can be filled with either a constant value such as 0 s or 255 s, or it can be filled with random or Gaussian noise [66].

2- Audio augmentation

Audio data augmentation is a technique used in audio signal processing to extend and diversify the training data set. It consists of applying controlled transformations and modifications to existing audio samples, intending to create new data instances that retain the essential characteristics of the original audio, but with variations or perturbations. There are some techniques used in audio data augmentation:

- **Time Stretching:** you can increase or decrease the speed of audio playback without changing its pitch or duration. This allows you to simulate variations in the rhythm or speed of speech.
- **Pitch Shifting:** the pitch or frequency of the audio is altered while maintaining its duration. This technique can simulate variations of the voice or musical notes.
- **Noise Additio**n: random noise is added to the audio signal. This can help the model to be more robust against environmental noise or variations in recording conditions [77].

2.4 Data Loading and Reading Speed Bottlenecks

In Deep learning, the biggest obstacle regarding the resources is the Bottleneck scenario, when one component of a system is less capable than the rest, making all other components run slower; this can resemble a water stream, when a tight spot reduces the whole flow and creates a buildup, in the current case, the data flow as follow: Data is read from Storage disk (HDD, SSD, etc.) to Random Access Memory (RAM), then preprocessed on the Central Processing Unit (CPU) and loaded to the Graphical Processing Unit (GPU) to be ready for training. in most cases, the bottleneck is the read speed of storage disks, which reads data at a slower speed compared to RAM speed, the problem is that the RAM size is not as big as the HDD or even SDD, and more expensive [78].

The bigger the data, the harder it is to load it all at once on RAM. Optimizing the model training is an active search topic that focuses on decreasing the total training time and increasing the data ingestion rate of the training process. There are many methods of training performance optimization and model optimization for both single GPU and multi-GPU setups which allow for horizontal scaling with more hardware. Recent hardware innovations help with improved model performance. Therefore, Preprocessing the full dataset once before training is viable if one wants to avoid the processing overhead in every iteration. If there is a long sequence of preprocessing functions that slow down the process, you can perform these operations on the original dataset once and write it to disk, then read the preprocessed files and directly feed it to the model [78].

2.5 Multimodal Emotion Recognition

According to Mehrabian's study [6], only 7% of emotions can be approximated by linguistic language, 38% through paralanguage, and 55% through face-to-face encounters. However, when applied to social activities, information gleaned from a single source may need more accurate. Emotional state can be better determined using multiple modalities, such as body posture, gestures, facial expressions, and speech [79].

Multimodal emotion recognition gathers data from several sources. However, it is necessary to specify where the various channels' data will be fused. Data fusion, feature fusion, and decision fusion are the three most common approaches to combining disparate data sets for analysis. Combining similar physical signals (such as two speech signals, two videos from two cameras, etc.) is one way to perform data-level fusion. Data fusion is impractical for multimodal fusion because it necessitates that all modalities have the same captors and signal characteristics all the time. Data-level fusion is depicted in Figure 2.19 [80].

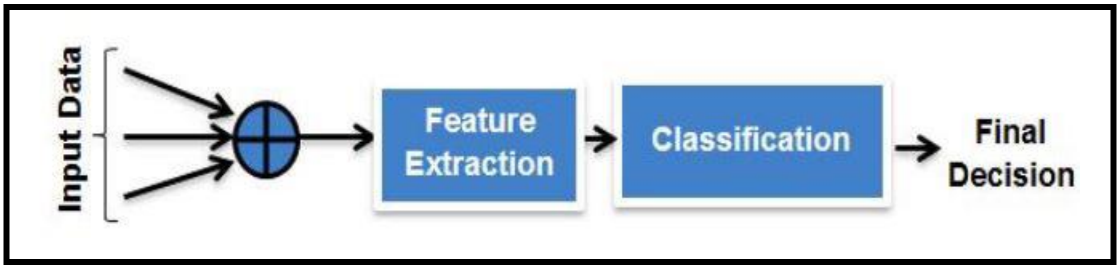


Figure 2.19: Data-level fusion method [80]

In feature-level fusion, all features gathered from all channels are fed into a single classifier, which then uses decision algorithms to determine the user's emotional state. The correlation between the collected features across modalities is beneficial for feature-level fusion. Disadvantages include failing to account for variations in the temporal organization, spatial detail, and quantitative significance. It also calls for coordinated use of different technologies. In addition to being computationally more expensive than combining at the decision level, the complexity of such a combination system is another drawback. In contrast, feature-level fusion can negatively affect the overall framework performance due to the increased dimensionality of the final feature vector. The concept of a feature-level fusion approach is depicted in Figure 2.20 [81].

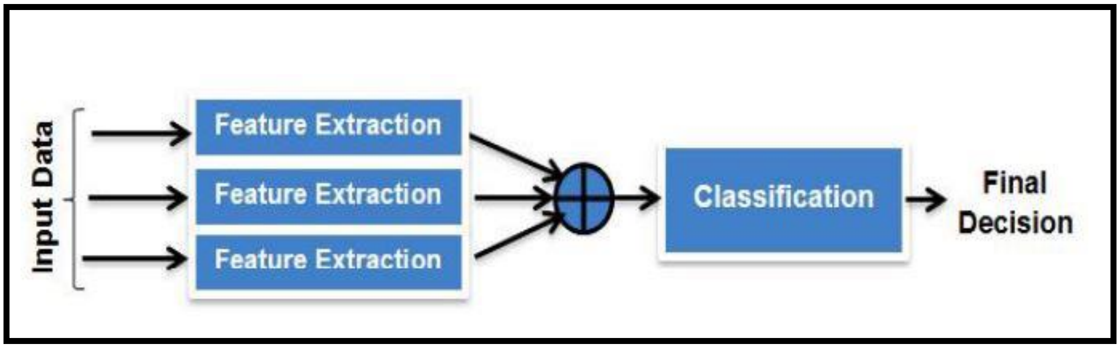


Figure 2.20: Feature-level fusion method [80]

In the decision-level fusion approach, each channel has its classifier that uses just its data to determine the emotional state of the person being observed. In this approach, a decision rule is critical in making a last-minute call on feelings. The law applied must consider the reliability of each channel's determination [81]. The benefits of fusing information at the decision level include:

- Not needing to synchronize across different modalities.
- Being able to use a relatively simple method.
- It doesn’t require high computational resources.

Because of this, most researchers involving multimodal emotion recognition employ fusion techniques at the decision-making level. Decision-level fusion is depicted in Figure 2.21 [80].

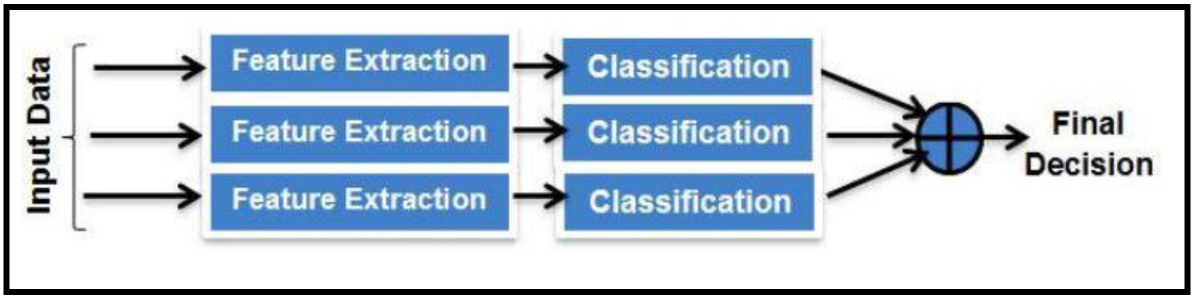
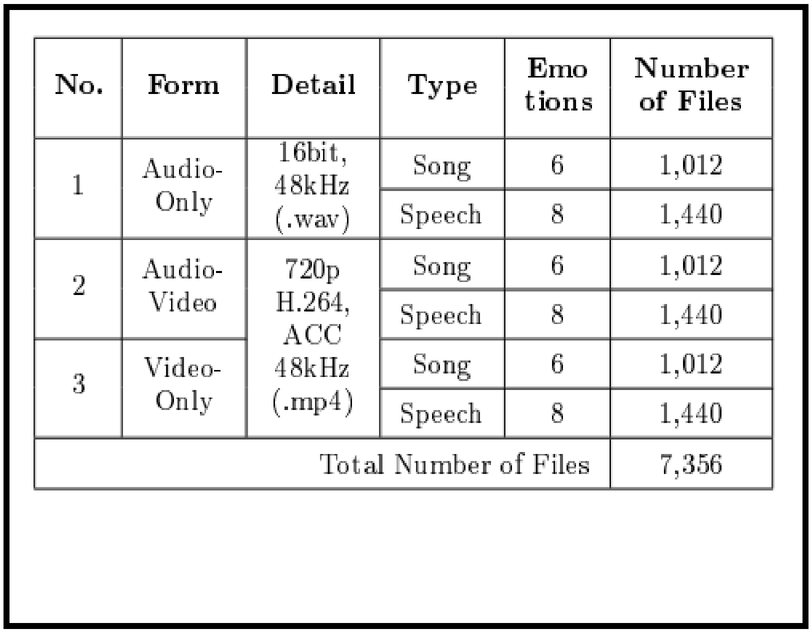


Figure 2.21: Decision-level fusion method [80]

2.6 Databases

2.6.1 The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)

The (RAVDESS) is a relatively new database, published on May 16, 2018. It consists of two main parts: facial and speech expression. The RAVDESS database has a total size of 24.8 GB. There are three forms of data (Audio-Only, Audio-Video, and Video-Only), consisting of 7,356 files. Each file is approximately three seconds long. All three forms are divided into two types. The first type is the song. There is a dynamic emotion expression of six basic emotions, such as neutral, calm, happy, sad, angry, and fearful. All six basic emotions have the intensity level of different emotions with two levels (normal and strong level) by using 23 professional actors (11 females and 12 males) vocalizing two lexically matched statements in a neutral North American accent. The age of the actors was between 21-33 years. The second type is speech. There is a dynamic emotion expression of eight basic emotions such as neutral, calm, happy, sad, angry, fearful, disgust, and surprised. All eight basic emotions have the intensity level of different emotions with two levels (normal and strong levels) by using 24 professional actors (12 females and 12 males). The age of the actors was between 21-33 years. Details of the database are shown in Figure 2.22.



No.	Form	Detail	Type	Emo tions	Number of Files
1	Audio-Only	16bit, 48kHz (.wav)	Song	6	1,012
			Speech	8	1,440
2	Audio-Video	720p H.264, ACC 48kHz (.mp4)	Song	6	1,012
			Speech	8	1,440
3	Video-Only		Song	6	1,012
			Speech	8	1,440
Total Number of Files					7,356

Figure 2.22: RAVDESS database

The sample images showing different emotional states from the RAVDESS database are shown in Figure 2.23 [24] [82].

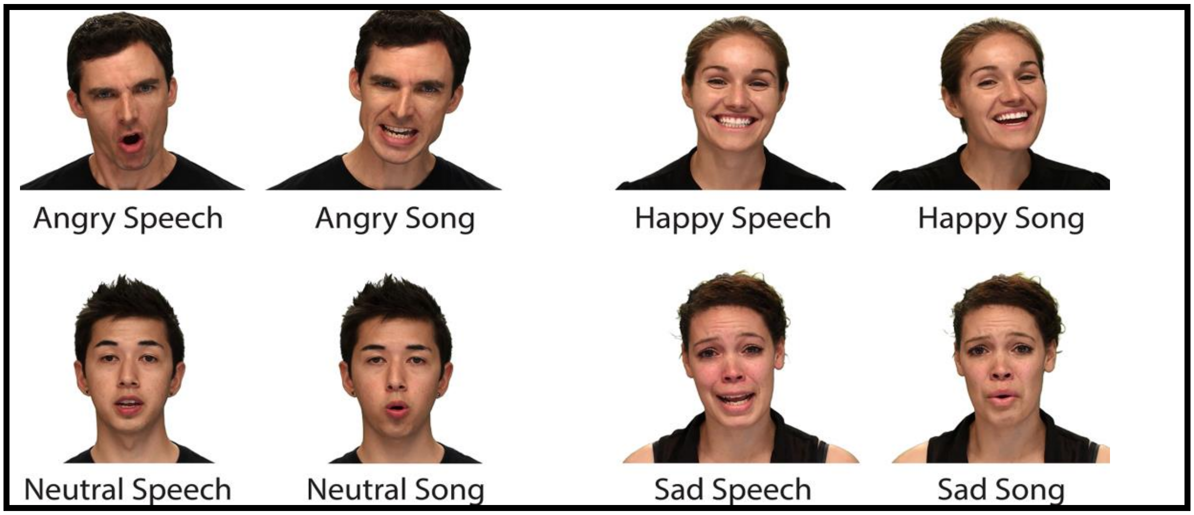
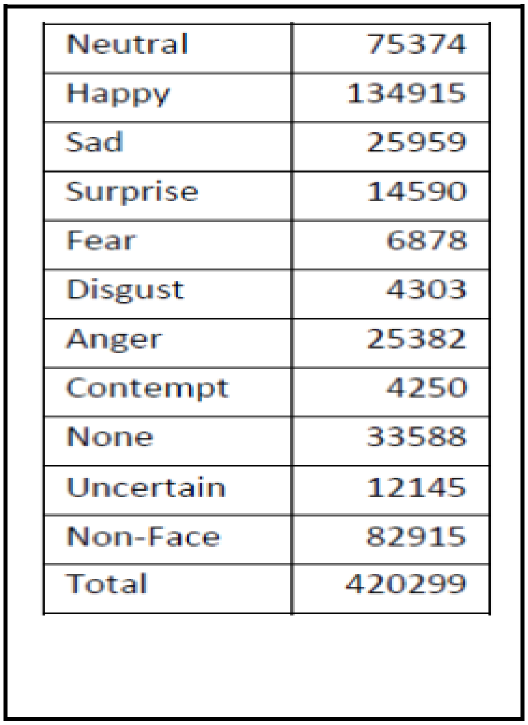


Figure 2.23 : Examples of the RAVDESS emotions

2.6.2 AffectNet

AffectNet (Affect from the InterNet) is the largest database of the categorical and dimensional models of effect in the wild. The AffectNet dataset contains more than 1,000,000 images from the Internet that were obtained by querying different search engines using emotion-related tags. AffectNet is by far the largest database that provides facial expressions in two different emotion models (a categorical model and a dimensional model), which can be used for studies in automated recognition of facial expressions, valences, and arousal in real-world scenarios. About 450,000 images already have manually annotated labels for eight basic expressions which are neutral, happy, sad, surprise, fear, disgust, anger, and contempt, as well as some non-emotion related image classes such as none, uncertain, and non-face. Figure 2.24 shows the number of annotated images in each category.



Neutral	75374
Happy	134915
Sad	25959
Surprise	14590
Fear	6878
Disgust	4303
Anger	25382
Contempt	4250
None	33588
Uncertain	12145
Non-Face	82915
Total	420299

Figure 2.24: Number of Annotated Images in each category

This database is highly imbalanced, the number of images in the largest category (happy with 134,915 images) is approximately 30 times larger than the smallest category (contempt, with 4250 images). Figure 2.25 shows some sample images from the dataset [83] [84].

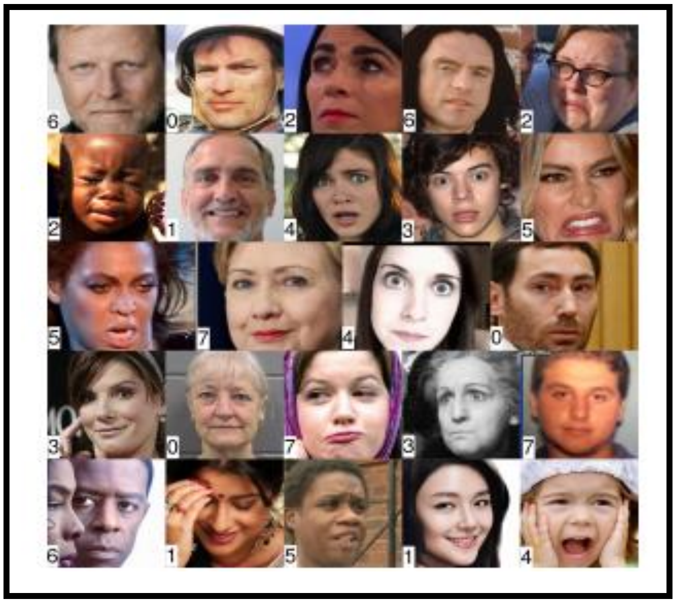


Figure 2.25: Sample of images from the AffectNet dataset (0-neutral, 1- happy, 2-sad, 3-surprise, 4-fear, 5-disgust, 6-anger,7-contempt)

2.7 Performance Evaluation Metrics and Confusion Matrix

Evaluation metrics are essential in calculating a model's performance. A confusion matrix is a tool for assessing a classification model's effectiveness. It is a simple crossbar showing the number of correct and incorrect predictions (or actual and predicted values) yielded by a classifier (or classification model) [85]. By visualizing the confusion matrix, an individual could determine the accuracy of the model by observing the diagonal values for measuring the number of accurate classifications. In Python, a confusion matrix can be obtained using the “confusion_matrix()” function which is a part of “sklearn” library. This function can be imported into Python using “from sklearn.metrics import confusion_matrix”[86]. To obtain a confusion matrix, users need to provide actual values and predicted values to the function. An example of a confusion matrix for two classes is shown in Figure 2.26. The metrics that are derived using the confusion matrix are explained in Table 2.1 [85].

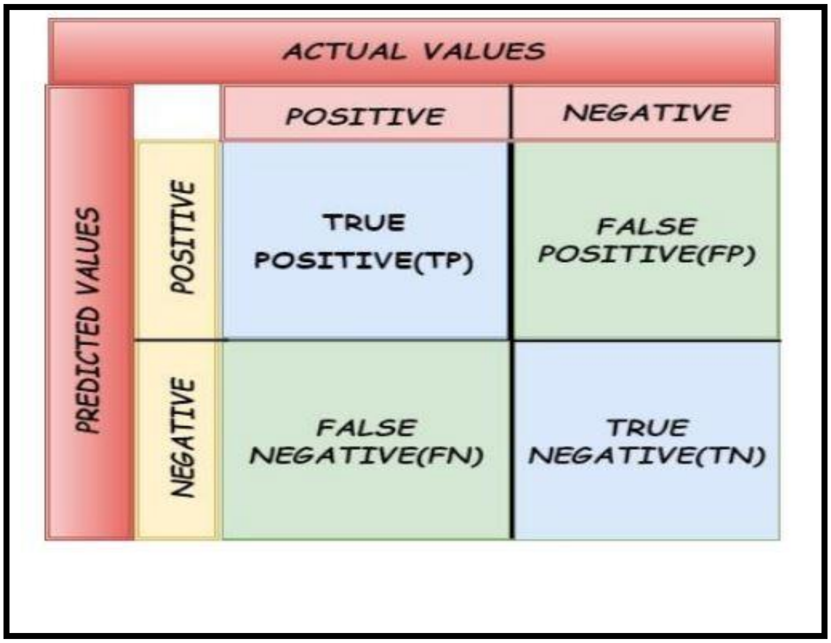


Figure 2.26: Confusion matrix

Table 2.1: Performance Evaluation Metrics

Metrics	**Definition**	**Calculation**
Accuracy	Correct-to-total-predictions ratio.	= (TP+TN) / (TP+FP+FN+TN)
Sensitivity/ Recall	True positives to total (real) positives ratio.	= TP/(TP+FN)
Specificity.	True negatives as a percentage of overall negatives	= TN/(TN+FP)
Precision	True positives to total expected positives ratio.	= TP/(TP+FP)
F1-Score	The recall and precision's harmonic mean.	=2*(Recall*Precision) / (Recall + Precision)

Chapter Three

The Proposed System

3.1 Overview

This chapter describes the proposed multimodal emotion recognition system. It benefits from the fusion of the outputs of many classifiers. Each one focuses on a distinct set of features from each model. The proposed models which include training cases and algorithms are also illustrated in detail in this chapter.

3.2 The Proposed Multimodal Emotion Recognition System

The proposed multimodal emotion recognition system consists of two phases: an audio phase and a visual phase. The visual phase uses the visual data to recognize the emotional state. The audio phase used the techniques needed to determine the emotional state of speech. The final stage of the multimodal emotion prediction is achieved using decision-level fusion, which uses the outputs of both visual and auditory classifiers as input to show the final emotion class. Figure 3.1 presents the component of the proposed system.

The methods used by the visual and audio phases are described in the following sections. The last part which is fusing-based emotion recognition discusses the fusing of classifier outputs to produce the final emotion prediction.

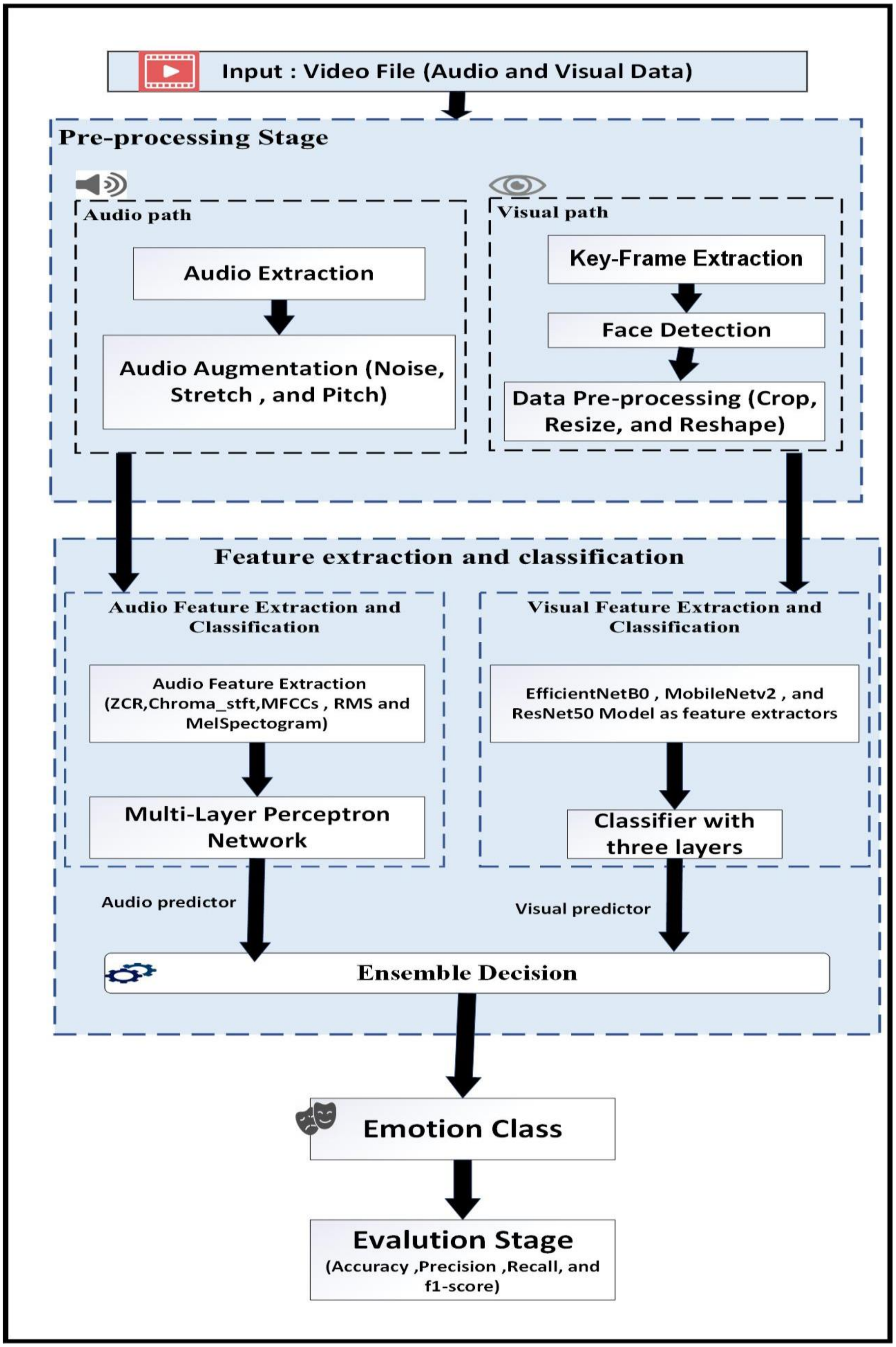

Input : Video File (Audio and Visual Data)
Pre-processing Stage
Audio path
Audio Extraction
Audio Augmentation (Noise, Stretch , and Pitch)
Visual path
Key-Frame Extraction
Face Detection
Data Pre-processing (Crop, Resize, and Reshape)
Feature extraction and classification
Audio Feature Extraction and Classification
Audio Feature Extraction (ZCR,Chroma_stft,MFCCs , RMS and MelSpectogram)
Multi-Layer Perceptron Network
Visual Feature Extraction and Classification
EfficientNetB0 , MobileNetv2 , and ResNet50 Model as feature extractors
Classifier with three layers
Audio predictor
Visual predictor
Ensemble Decision
Emotion Class
Evalution Stage
(Accuracy ,Precision ,Recall, and f1-score)

Figure 3.1: General Framework of Proposed System

3.3 Preprocessing Steps

The preprocessing stage includes a set of operations as follows:

A. Augmentation

Augmentation is usually used when the dataset is small for deep learning and it helps to increase sample size by producing changeful data with minor modifications. This can assist the model to reduce overfitting and increase generalization. This process is done in video by some preprocessing steps on images such as flip, shift, rotate and zoom. Data augmentation steps are described in Algorithm 3.1 for frame and Algorithm 3.2 for audio.

Algorithm 3.1*: Data Augmentation for frame*	
Input **:**	Frame
Output **:**	Augmented frame
Steps	**// Assume that the flip1 variable controls whether or not the flip frame is completed.** Set flip1 = random number between 0 and 1. **If** (flip1 < 0.5) **then** Flip the frame from the left to right **// With a 50% probability, the frame is flipped horizontally** **End if** **// Assume that the shift1 variable controls whether or not the shift frame is completed.** Set shift1= random number between 0 and 1 **If** (shift1<0.5) **then** Shift the frame vertically and horizontally by Max ±10%. **// With a 50% probability, the frame is randomly shifted vertically and horizontally by up to 10% of its width and height.** **Endif**

	// **Assume that the rotate1 variable controls whether or not the rotate image is completed.** Set rotate1 = random number between 0 and 1. **If** (rotate1 <0.5) **then** rotate the frame, Max ±10%. **// With a 50% probability, the frame is randomly rotated by an angle between -10 and 10 degrees.** **endif** // **Assume that the zoom1 variable controls whether or not the zoom frame is completed.** Set zoom1= random number between 0 and 1. **If** (zoom1< 0.5) **then** The frame is zoomed in or out by a factor between 0.9 and 1. **endif** Resize frame if necessary, according the selected model. Return Augmented frame **End**

The preprocessing on audio could include noise, stretch and pitch that help to introduce variations and make the model more robust to noise to recognize patterns at different speeds and pitches and enhance its generalization ability. Data augmentation steps are described in Algorithm 3.2 for audio.

Algorithm 3.2*: Data Augmentation for audio*	
Input **:**	Audio data
Output **:**	Augmented audio data
	// Assume that the noise variable controls whether or not the noise rate is completed. Set noise = random number between 0 and 1. **If** (noise < 0.5) **then**

Steps	Gaussian noise with an amplitude of 3.5% of the maximum amplitude of the audio signal is added to the audio data. // With a 50% probability. **End if** **// Assume that the stretch variable controls whether or not the stretch rate is completed.** Set stretch = random number between 0 and 1. **If** (stretch <0.5) **then** The audio data is stretched or compressed by a factor between 0.8 and 1.2 **// With a 50% probability** // **Assume that the pitch variable controls whether or not the pitch rate is completed.** Set pitch = random number between 0 and 1. **If** (pitch <0.5) **then** The pitch of the audio data is shifted by a factor between -1 and 1 **// with a 50% probability** **End if** Return Augmented audio data. **End**

B. Dataset Caching

The slowdown in the training process is caused by the bottleneck in data reading. Advanced server hardware is designed to function like a deep learning machine with a colossal RAM. Because we don't have access to such devices, therefore, we need to prepare the data in advance. Algorithm 3.3 is used for data preparation such as 1) reading the data of video of the dataset from the disk, 2) extracting the frames of images as in Algorithm 3.4, 3) extracting audio and labels from each video, and finally 4) taking all the steps that are explained regarding the frames of images and audio according to the Algorithms 3.3 and 3.4. These steps are carried out for all datasets, resulting in a single file in a serialized structure that is simpler to read and process. This file is named "Cached Dataset" and may be loaded into RAM for instant access.

Algorithm 3.3*: Dataset caching*

***Input* :**	Dataset path
***Output* :**	Cached Dataset file

Steps

```
Set NewClass as Empty Class to save the frames, audio and labels of each video in the dataset path.
For each video in the dataset do
    Call the Algorithm 3.4 to get frames
    Extract Audio from the video file.      // Using MoviePy library.
    Get a video label from each video in the dataset.
    Add the label, frames and audio of each video to NewClass.
Next For
Save the NewClass file as a pickle file.      // This is a cached dataset file
End
```

Each video with emotion requires several hundred frames to be processed, which increases processing time. As a result, the emotion video is summed by a few frames.

To achieve this purpose, several approaches can be utilized, including the motion analysis-based method, shot boundary-based method, visual content-based approach, and clustering-based approach. The shot boundary-based approach, which selects the start, middle, and end frames of every video as key frames, is used in this work. Therefore, the appropriate skip number, which was chosen in Algorithm 3.4, is ten (i.e., for each video in the dataset, we take one frame every ten frames) for the following reasons:

1. Typical expression development includes the stages of onset, apex, and offset. As a result, this approach has been utilized to collect expressions at three stages.
2. For each video in the dataset, one frame is extracted every ten frames, so that the number of frames extracted does not become too huge or small, and it is adequate for the training process and changes in facial expressions. Therefore, the skip number equal to ten is selected.
3. To avert the extra time overhead caused by finding the ideal selection of frames, this time will then raise the proposed system's response time. Thus, the suggested visual path employs the simplest and quickest approach (i.e., the shot boundary method).

Algorithm 3.4: ***Extracting frames from video***

***Input* :**	Video_path , skip_no=10
***Output* :**	frames

Steps

```
Set the frames array as an empty array to save frames for each video.
Open video_path file
Set frame_count = 0                // Keep track of frame number
While (frame_count  Mod  skip_no = 0) do
   Read frame
   Detection of the face.
   Draw a rectangle to show the detected area.
   Crop the area of the face.
   Resize this Cropped frame image.
   Add the frame to frames array
   If (Video_path is empty) then
        Break
   Endif
   Set frame_count = frame_count+1
```

```
End while
Close the video_path file
Return frames array
End
```

Algorithm 3.5 is used to prepare a cache frames of images file which takes the frames of images only from the cached dataset file and makes resizing and reshaping of each frame and creates two files which are the training file and testing file as a pickle file (data serialization).

Algorithm 3.5*: Cache dataset image frames*

***Input* :**	Cached Dataset file
***Output* :**	Training file, Testing file as pickle

Steps

```
Load cached dataset to read data frames.
Set List as Empty List.
For each image frame in the cached dataset do
   Read imageframe.
   Resize the image frame to the required dimension (Weight*Height).
   Reshape the image frame to the required channel.
   Get the label of the image frame.
   Save the image frame and its label to List. //The List contains all frames and its labels.
Next for.
Get the Training file and Testing file using the (train_test_split) function from the sklearn package according to the appropriate test ratio.
Save Training and Testing as a pickle file to the required location.
End
```

To cache the audio file only, Algorithm 3.6 is used. It includes the following steps 1) reading all data of audio from the disk, 2) adding an augmentation stage to increase the sample data of audio three times, 3) extracting features of audio data, 4) normalization of data, and 5) splitting the data for training file and testing file combining them into a single file, simpler-to-read and process in serialized form as pickle type files. The feature normalization process is crucial to correct differences between speech cases to ensure that all feature values fall between 0 and 1.

Algorithm 3.6*: Cache dataset audio files*

***Input* :**	Cached Dataset file , multiplay _no=3 //(for augmentation stage)
***Output* :**	Training file, Testing file as pickle

Steps

```
Load cached dataset to read audio data.
Set List as Empty List.
For in-range multiplay _no do
    For each data audio in the cached dataset file do
        Read data audio.
        Get sampling_rate of data audio.
        Get the label of data audio.
        Set audioaugment = Algorithm (3.2). // add augmentation of data audio
        Set Features = Algorithm (3.7).     // find features of data audio.
        Save Features and their labels to List. // List1 contain all features of data
                                  audio and its labels.
    Next for.
Next for
Normalize the numbers in the List.
Get the Training file and Testing file using the (train_test_split) function from
the sklearn package according to appropriate test ratio.
Save Training and Testing as a pickle file to the required location.
End
```

In this dissertation, the features are used for each audio file to avoid the problem of using one feature as shown in Algorithm 3.7 using librosa library. The used features are Zero Crossing Rate (ZCR), chroma and Short-Time Fourier Transform (Chroma_stft), Mel-frequency cepstral coefficients (MFCC), Root Mean Square (RMS) , and MelSpectogram. This Algorithm is called in Algorithm 3.6 to obtain the features of all audio files.

Algorithm 3.7*: Extracting Audio Features*

***Input* :**	audio data, sampling rate
***Output* :**	result array.
Steps	Set result as an empty array Set zcr = zero_crossing_rate(audio data) Set result =zcr Set stft = stft(audio data) Set chroma_stft (stft, sampling rate). Set result = chroma_stft. Set mfcc = mfcc (audio data, sampling rate). Set result = mfcc. Set rms = rms (audio data). Set result = rms Set Mel = MelSpectogram (audio data, sampling rate). Set result = Mel. Return result. **// This array contains the features of an audio file** **End**

Algorithm 3.8 explains the process of generating the batches of training video frames for the main training process, where it uses the cached training files as a list, performs random permutation and augmentation as in Algorithm 3.1 to prevent overfitting and achieve generality of the model,

and then the pair (image and label) are fed to the GPU as batches according to an appropriate size. The appropriate batch size used in this algorithm is 16.

Algorithm 3.8*: Cached data Batches Generator*

***Input* :**	Cached training list, BatchSize=16
***Output* :**	(Image, label) pairs Batches.

Steps

```
Set Batch_list as an empty list.      // To save the (image, label) pairs for each batch.
Work randomly permutation of entered training data.
While Training is Run Do        // iterate for each epoch
    BatchList_size = 0              // number of pairs in Batch_List
    Set Batch_List as Empty List
    For Each Pair in Permutated Data Do
        Call Augmentation Algorithm    // Algorithm 3.1
        Add (image, Label) pair to Batch_List.
        BatchList_size = BatchList_size + 1.
        If BatchList_size = BatchSize then
            Yield Batch_List             // Feed the Batch_List to training
            Set Batch_List as Empty List
            BatchList_size = 0
        End if
    Next for
End while
End
```

3.4 Designing the Models

This section includes two phases, the visual phase and the audio phase.

A. Visual phase

To create an acceptable network for performing visual emotion recognition both fast and accurately, several models must be designed, tested, and analyzed. The defects are discovered and conclusions are drawn on how to develop a better scheme. In general, the architecture of the proposed model is divided into two parts:

- The first part is the feature extractor, which consists of image processing layers such as 2D convolution layers and 2D pooling layers grouped into various sizes and many layers. The outcome is a feature vector in one dimension.
- The second part is the classifier, which takes the feature vector that was extracted in the first part and runs it through several dense layers with RELU as an activation function. The output layer then has seven neurons with the SoftMax activation function, producing seven values that each represent the probability that an image belongs to each class.

The training speed of transfer learning approaches can be enhanced by using previously trained networks. Alternately, instead of training the entire model from scratch, the model that was trained to classify some irrelevant dataset may be reused by using only the weights of the layers used for feature extraction (without the classification component) to initialize the feature extraction layers that are being utilized.

The models of transfer learning have already been trained using the ImageNet dataset, and fine-tuning techniques are utilized to improve the accuracy and generalizability of these models. As a result, the proposed models based on pre-trained models as feature extractors are being trained for two cases:

- **Case 1:** Fine-tuning the classification part and freezing only the feature extraction part. This process enables the classification layer to learn what features to search for in the output. Because few parameters require training, it requires less training time than other cases.
- **Case 2:** Fine-tuning both the classification and extraction parts so that the entire model is trained without any parts being frozen. This will improve learning accuracy, but it requires additional training time.

i. The EfficientNetB0 Model

In this model, the weights from the pre-trained EfficientNetB0 model have pertained using ImageNet Dataset. It is transferred to the newly built model by replacing its old classifier with the new classifier. The Global Average Pooling2D is computed for the features, then flattened to obtain the final features that are fed to the new classifier.

The newly added classifier consists of a dropout layer followed by two dense dropout layers with (2048 and 1024) neurons respectively and Rectified Linear Unit (RELU) activation function. Also, the last layer is the Dense layer with seven neurons (because the model will be utilized to classify an image into one of seven emotions) and the Softmax activation function. The structure of this model is shown in Figure 3.2.

ii. The MobileNetV2 Model

This model has been trained using the same strategies, techniques and structure as the previous model except that the pre-trained model weights are replaced by MobileNetV2, the implementation details described in Chapter 2. The schematic of this model is shown in Figure 3.2.

iii. The ResNet50 Model

This model has been trained using the same strategies, techniques and structure as the previous model except that the pre-trained model weights are replaced by ResNet50, the implementation details described in Chapter 2. The shape of this model is shown in Figure 3.2.

The features are extracted from the proposed models using pre-trained models such as EfficientNetB0, MobileNetV2, and ResNet50. The weights from the single pre-trained models such as (EfficientNetB0, MobileNetV2, and ResNet50) are transferred to the newly built classifier by taking only the features extraction part from them, leaving its old classifier, and replacing it with a new classifier. However, each proposed model has been trained for two cases:

A. Freezing the feature extraction component and fine-tuning the classification part only and unfreezing it and fine-tuning the whole model, this method is used with feature extraction parts that have already been trained on the ImageNet dataset and are now being trained again on the AffectNet dataset.

B. Using feature extraction parts from step A to train again separately on the RAVDESS dataset by fine-tuning the whole model.

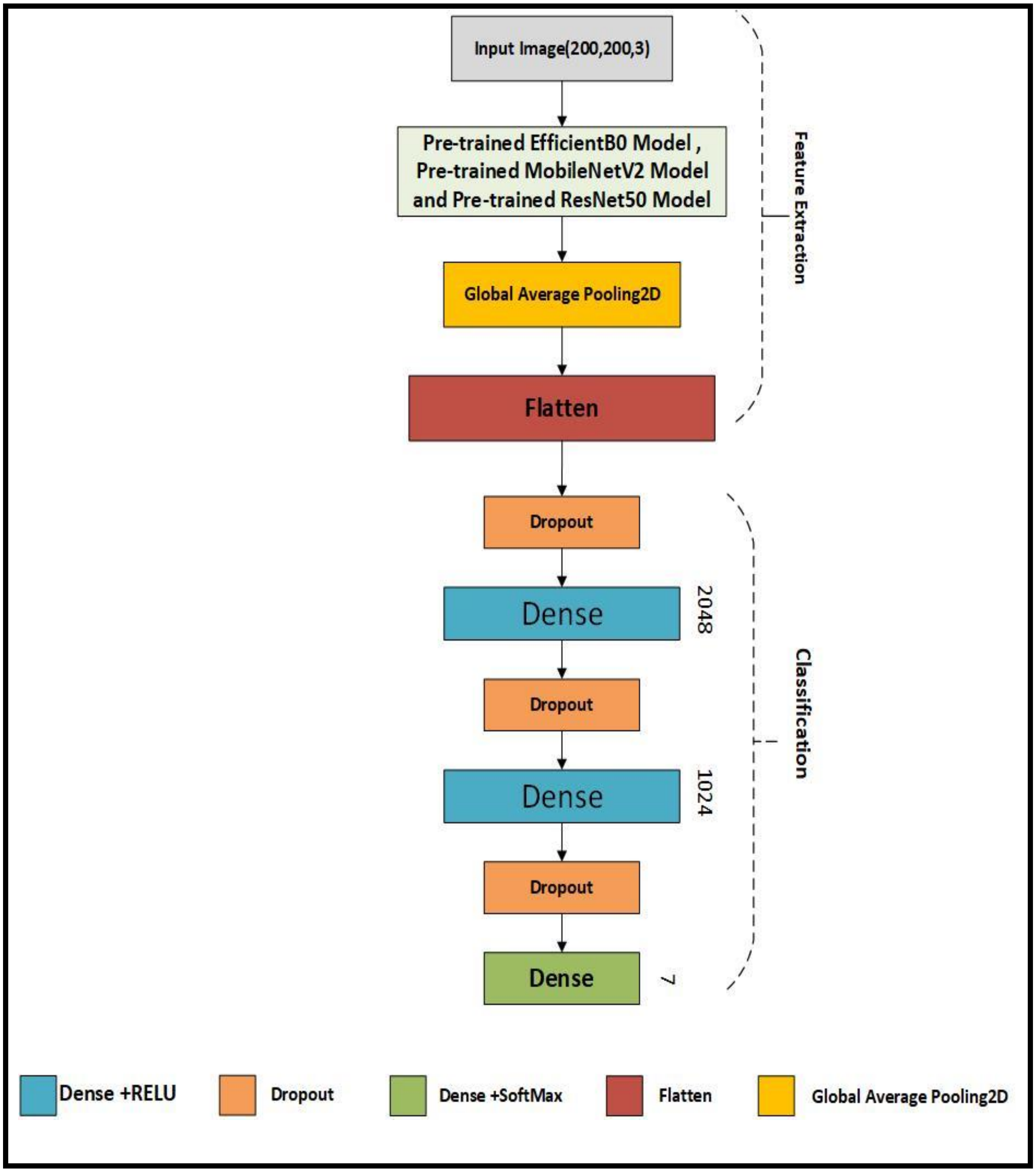


Figure 3.2: The structure of the proposed models

Many experiments were conducted to get the best model regarding accuracy and size. Such as changing the classifier layers with deeper layers and reducing and changing the learning rate and dropout rate for the best model.

When the model training is completed, the training and validation accuracy are obtained and accordingly, the best model is selected. The resulting model can then be used to predict the emotion of frames in the visual path in video.

B. Audio Path

The proposed Multi-Layer Perceptron Network (MLP) for the audio model consists of 162 neurons for the input layer, two dense dropout layers with (256 and 128) neurons and three dense layers with (128, 64, and 32) neurons, respectively, with a Rectified Linear Unit (RELU) activation function. Finally, a last dense layer with seven neurons and the Softmax activation function (because the model is used to classify audio into one of seven predictions). The structure of this model is illustrated in Figure 3.3.

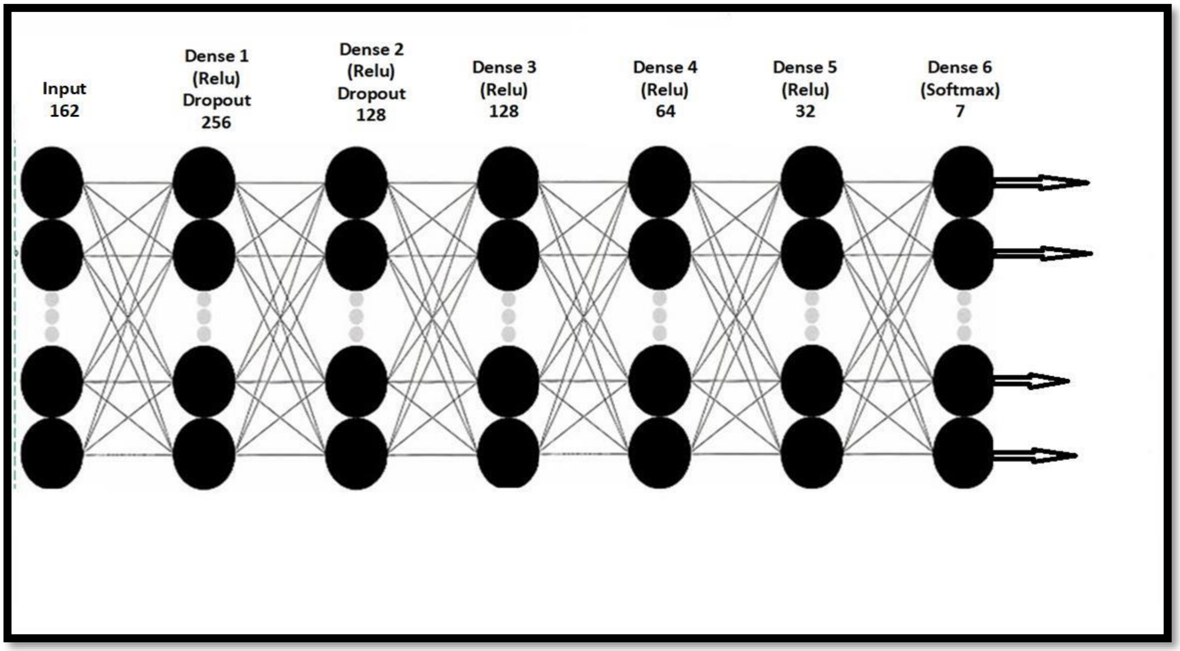


Figure 3.3: Proposed MLP model for the audio path

3.5 Final Prediction

Each video contains one output from the audio path and a set of outputs from the visual path corresponding to the skip number, as in Algorithm 3.4. To fuse the visual and audio prediction paths, decision-level fusion is utilized as shown in Figure 3.4. Decision-level fusion is often used in scenarios where multiple sources provide diverse information, and combining their decisions can lead to improving the overall performance. The proposed system treats audio samples and video frames asynchronously. Therefore, it is suitable for decision-level fusion. The ensemble decision is used to obtain the final prediction with higher accuracy as in Algorithm 3.9.

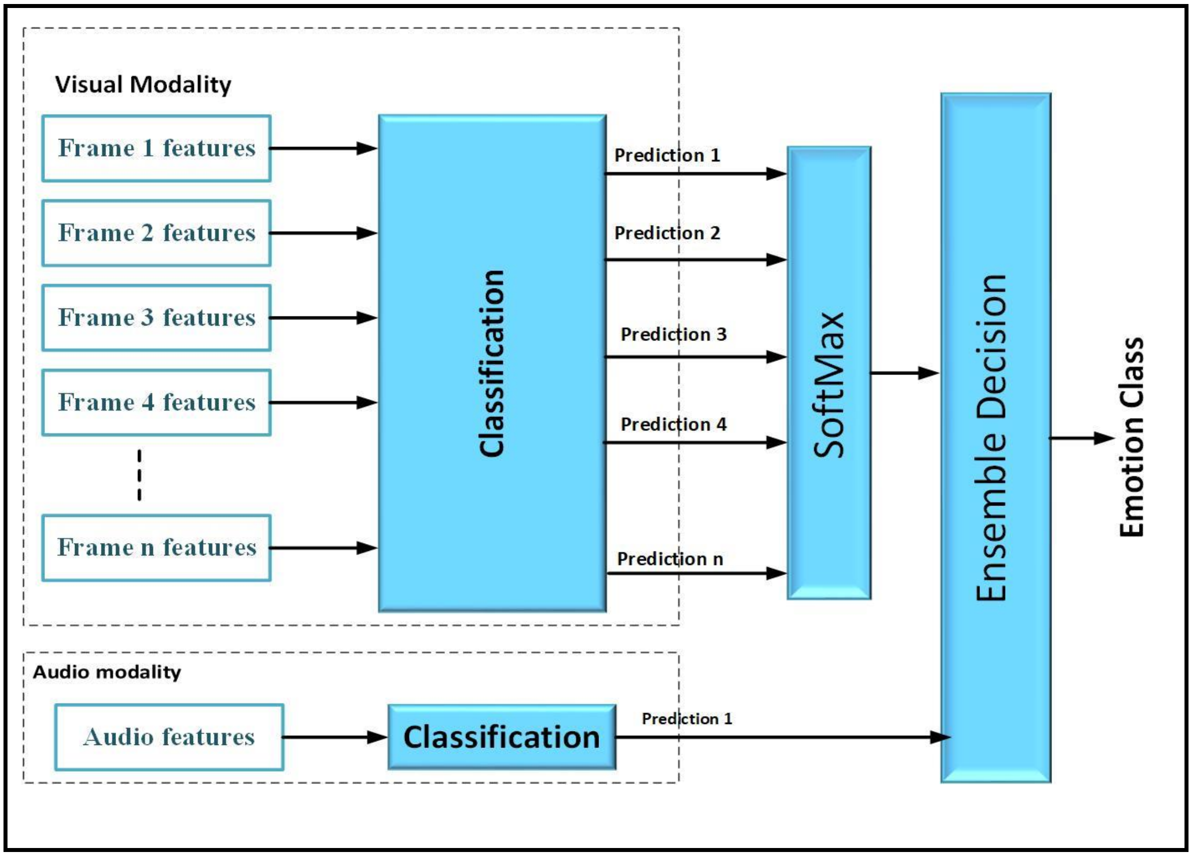


Figure 3.4: Decision-level fusion

Algorithm 3.9*: Visual /Audio Final Prediction Decision Level Fusion*

***Input* :** video, image_model, audio_model , image_path_weight=0.5 , audio_path_weight =0.5

***Output* :** Final prediction

Steps

```
Load any video in the database.
// Extract the frames of the video
Set frames = Extract frames for each video by using Algorithm 3.4
//Get prediction of the frames in video using image_model
Set frame_predictions = image_model (frames).

// Calculate the average of all arrays.
Set frame_predictions = mean of frame_predictions

//Get Audio from Video.
Set audio = Extract Audio From video
// Get Audio Features
features = Extract Audio Features by using Algorithm 3.7

// Get predictions to Audio by using audio_model
Set audio_predictions = audio_model (features)

//Use weights to ensemble the final prediction to find the maximum emotion class value
final prediction = maximum of (frame_prediction * image_path_weight + audio_predictions * audio_path_weight )

Return final prediction

End
```

Chapter Four

Results and Discussion

4.1 Overview

This chapter describes the specifications of the used system, the preprocessing results include feeder testing, the result of training and testing, all single models. The proposed models-based transfer learning for visual path and the proposed MLP model for audio path then that results from the experiments for enhancement the models, comparing all the results with each other and with other researches.

4.2 System Environment and Specifications

All the experiments are performed on the system shown in Table 4.1. This information is crucial when taking several of the testing results, the speed of preparing the data, and training the models, which depend highly on the system's performance.

Table 4.1: System environment Specifications

Operating System	Windows 10 Home 64-bit
CPU	11th Gen Intel(R) Core (TM) i7-11800H @ 2.30GHz
GPU	NVIDIA GeForce RTX 3070 8 GB
RAM	32GB 3200Mhz
Python Version	3.8.15
Conda Environment	conda 23.1.0
Nvidia Driver Version	511.65
TensorFlow Version	2.9.0
CUDNN	8.1.0
Cuda Toolkit	11.6
Visual Studio Code	1.80.0

4.3 Preparing Databases and Data Feeder

4.3.1 AffectNet Database

As shown in Chapter 2, the images in the first database used have a set of challenges, such as varying head positions, light changes, occlusion, poor resolution of some images, and imbalance because there are different numbers of images for each emotion. This makes it too hard to obtain a model that works well and accurately for all emotions. Thus, seven emotions have been balanced, and all sets are prepared and cached in one file with an HDF5 (Hierarchical Data Format version 5) type file. The seven basic emotions used are "angry, disgust, fear, happy, neutral, sad, and surprise". Figure 4.1 shows the number of images for each emotion after balancing.

In the training stage, 80631 images were used, but in the testing stage 3500 images are used. This database has been used for the purpose of the training process in scenarios 1 and 2 by feeding data using the Algorithm 3.8.

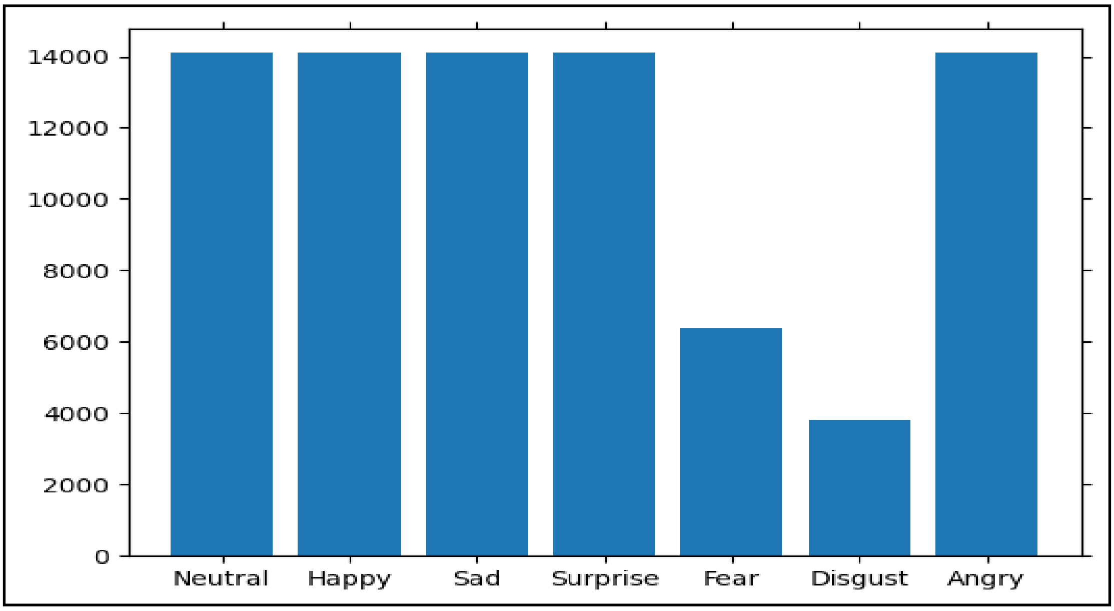


Figure 4.1: AffectNet database after balancing

4.3.2 Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS Database)

The second database, which has been used and described in Chapter 2, contains 1440 video (Audio -Visual video files) for the speech files and 1012 for the song files.

All videos (Audio -Visual video files) in Speech and Song files are prepared to obtain the image frames, audio, and labels as in the cache dataset Algorithm 3.3 and they are mixed them in one file as a pickle type file. The image frames are extracted from each video by calling the extract frames Algorithm 3.4 within the previous algorithm, where one frame is taken every ten frames, and faces are detected and cropped.

After that, cache dataset image frames Algorithm 3.5 takes the mixing file and prepares the 28,874 image frames only, divided into seven basic emotions only by resizing and reshaping each frame to (200 * 200 * 3) size and dividing it into 80% for training and 20% for testing, and save it in a pickle type file.

The cache audio files Algorithm 3.6 takes the cached mixing and prepares the audio only by finding the features, such as Zero Crossing Rate (ZCR), chroma and Short-Time Fourier Transform (Chroma_stft), Mel-frequency cepstral coefficients (MFCC), Root Mean Square (RMS), and MelSpectogram, by calling the Algorithm 3.7. Since the audio data are few and to avoid the occurrence of overfitting, we use the data augmentation Algorithm 3.2 to increase the number of data three times so that the data becomes 6374 and it is divided into 80% training and 20% testing and save it in pickle type file. This database has been used for the purpose of the training process in case 2 by feeding data using the Algorithm 3.8.

4.4 Training and Testing the Proposed Models Based on Transfer Learning Models for Visual Path

The training is carried out in two cases for each proposed model. First, it freezed the feature extraction part, fine-tuned the classification part only. Fine-tuned, and feature extraction and classification parts using the AffectNet database. Second, the feature extraction and classification parts were fine-tuned using the RAVDESS database. Table 4.2 shows the parameters used to train the proposed models.

Table 4.2: The parameters for all the proposed models

Parameter	Value
Input Image Size	200*200 *3 (color image)
No of Epochs	20 (First and Second cases) or 50 (Third case)
Batch Size	16
Dropout Rate	0.2
Learning Rate	0.001
Optimization function	ADAM

4.4.1 The EfficientNetB0 Model

Two cases were used to train the model. The outcomes of each case are presented here.

- **Case 1:** Freezing the feature extraction part and fine-tuning the classification part only during training by using AffectNet database. Model accuracy and loss in this situation is depicted in Figure 4.2. At epoch 20, the accuracy is 60%. It is shown that the loss error grows after a few epochs, but the training error continues to decrease until the last epoch. This indicates that the model is overfitting.

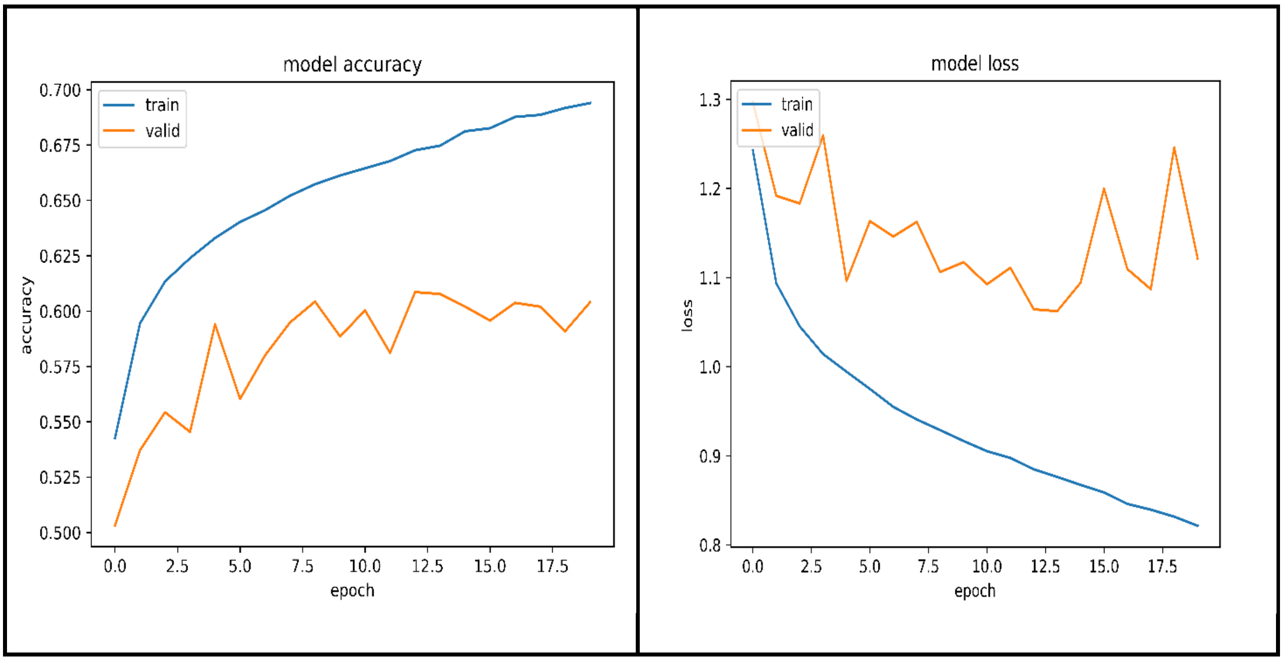
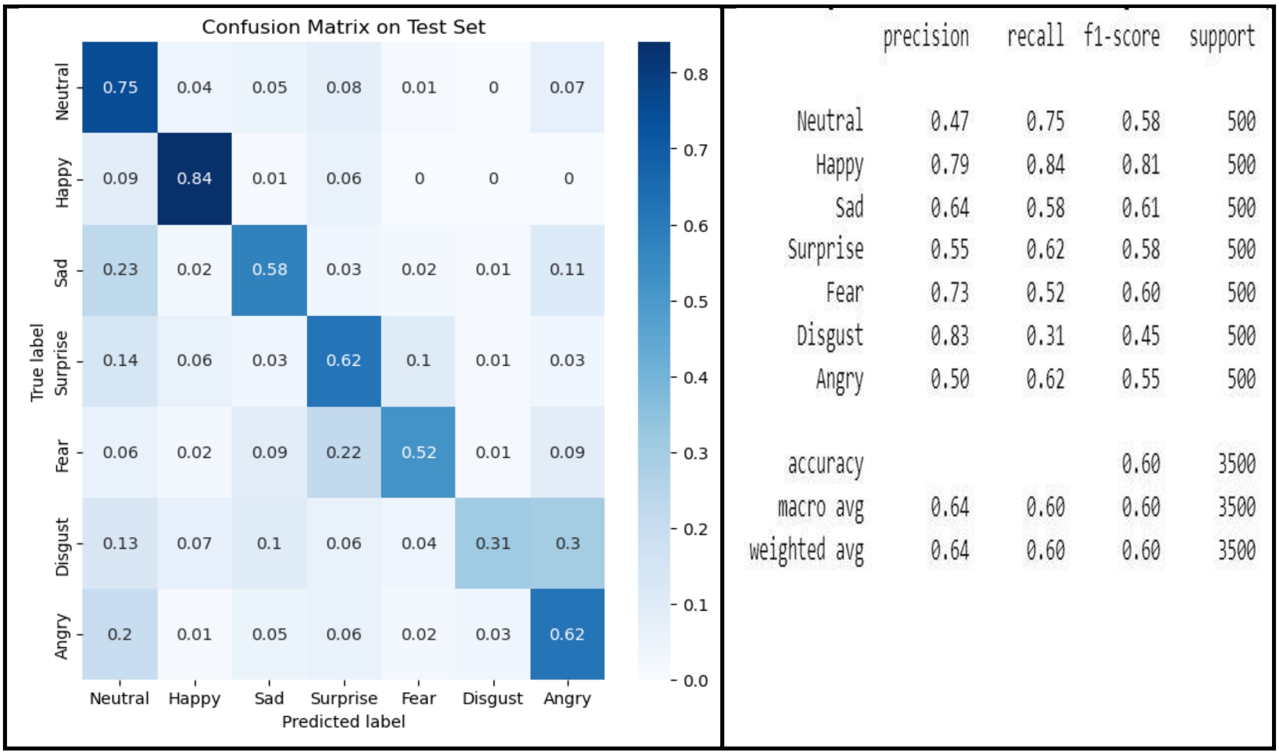


Figure 4.2: Model accuracy and loss after freezing the feature extraction part

The classification report and confusion matrix produced by testing the final model are illustrated in Figure 4.3.



```
              precision    recall  f1-score   support

     Neutral       0.47      0.75      0.58       500
       Happy       0.79      0.84      0.81       500
         Sad       0.64      0.58      0.61       500
    Surprise       0.55      0.62      0.58       500
        Fear       0.73      0.52      0.60       500
     Disgust       0.83      0.31      0.45       500
       Angry       0.50      0.62      0.55       500

    accuracy                           0.60      3500
   macro avg       0.64      0.60      0.60      3500
weighted avg       0.64      0.60      0.60      3500
```

Figure 4.3: The classification report and confusion matrix after freezing

The model accuracy and loss for training by fine-tuning both the feature extraction and classification parts by using the AffectNet database (Unfreeze of parts) are demonstrates in Figure 4.4. The accuracy is 61% at epoch 20. Figure 4.5 presents the classification report and confusion matrix generated by testing the final model.

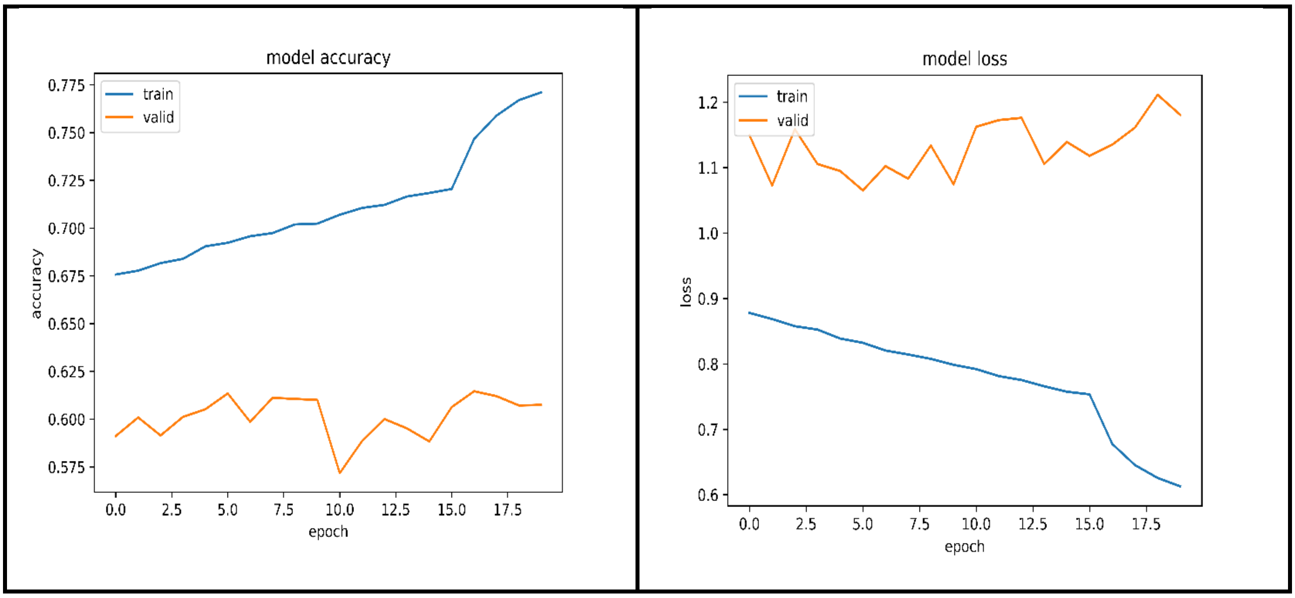
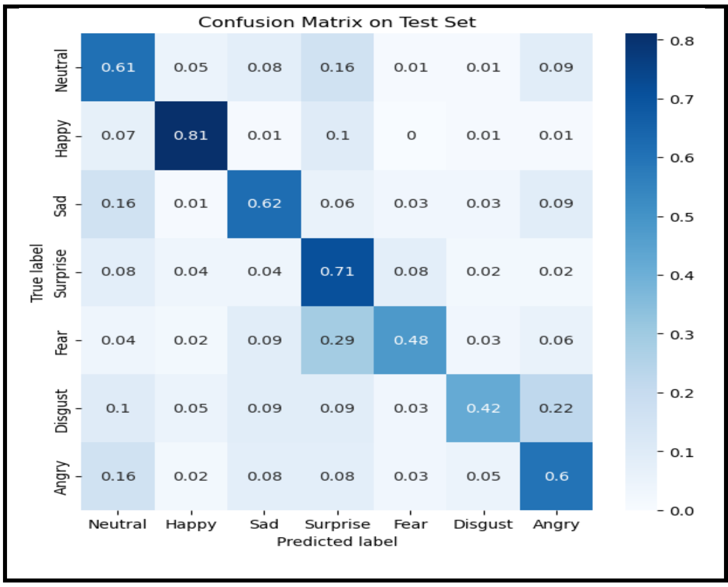
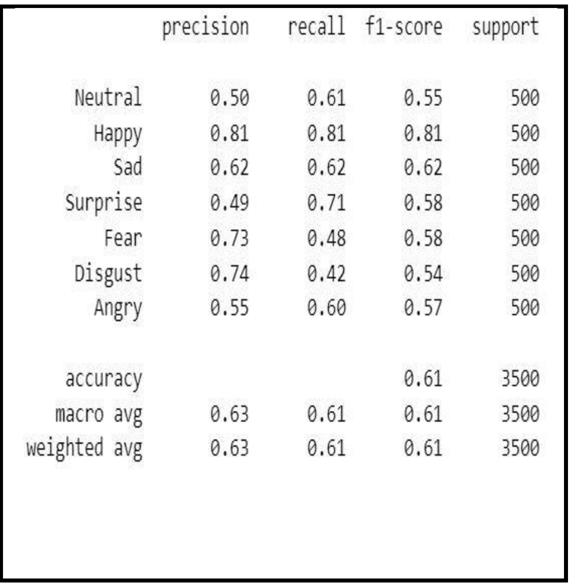


Figure 4.4: Accuracy and loss after fine-tuning the feature extraction and classification parts



	Neutral	Happy	Sad	Surprise	Fear	Disgust	Angry
Neutral	0.61	0.05	0.08	0.16	0.01	0.01	0.09
Happy	0.07	0.81	0.01	0.1	0	0.01	0.01
Sad	0.16	0.01	0.62	0.06	0.03	0.03	0.09
Surprise	0.08	0.04	0.04	0.71	0.08	0.02	0.02
Fear	0.04	0.02	0.09	0.29	0.48	0.03	0.06
Disgust	0.1	0.05	0.09	0.09	0.03	0.42	0.22
Angry	0.16	0.02	0.08	0.08	0.03	0.05	0.6



```
              precision    recall  f1-score   support

     Neutral       0.50      0.61      0.55       500
       Happy       0.81      0.81      0.81       500
         Sad       0.62      0.62      0.62       500
    Surprise       0.49      0.71      0.58       500
        Fear       0.73      0.48      0.58       500
     Disgust       0.74      0.42      0.54       500
       Angry       0.55      0.60      0.57       500

    accuracy                           0.61      3500
   macro avg       0.63      0.61      0.61      3500
weighted avg       0.63      0.61      0.61      3500
```

Figure 4.5: The classification report and confusion matrix after fine-tuning the whole model

- **Case 2:** Training by fine-tuning both the feature extraction and classification parts by using the RAVDESS database (Unfreeze of parts). Figure 4.6 demonstrates the model accuracy and loss for this case. The accuracy is 98 % at epoch 50. Figure 4.7 depicts the classification report and confusion matrix generated by testing the final model.

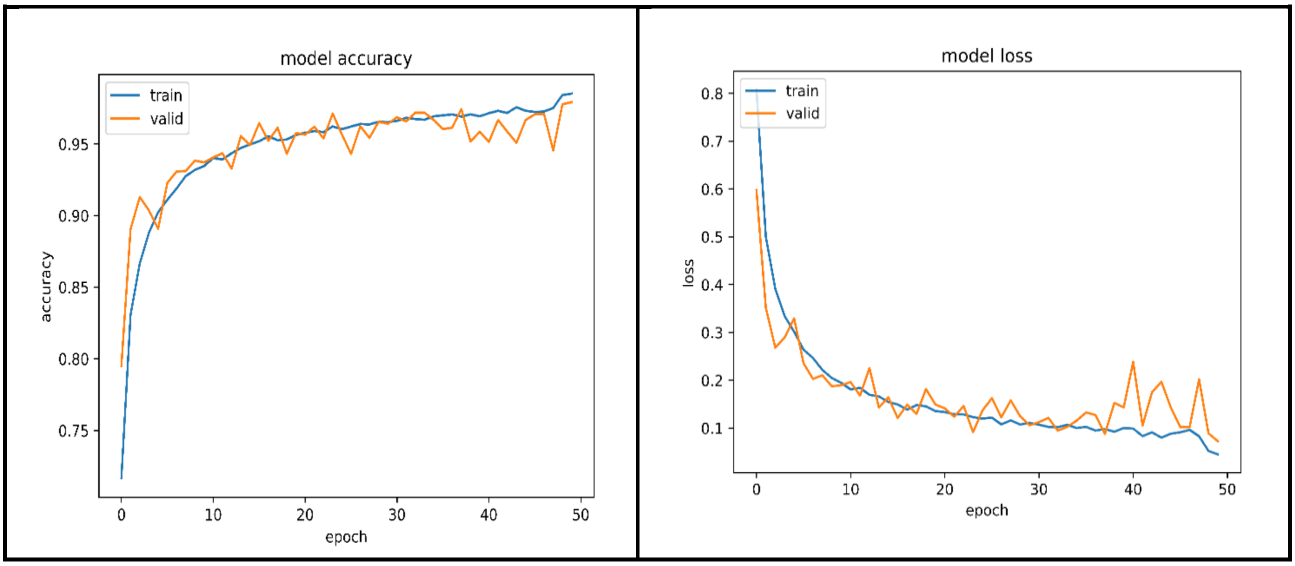
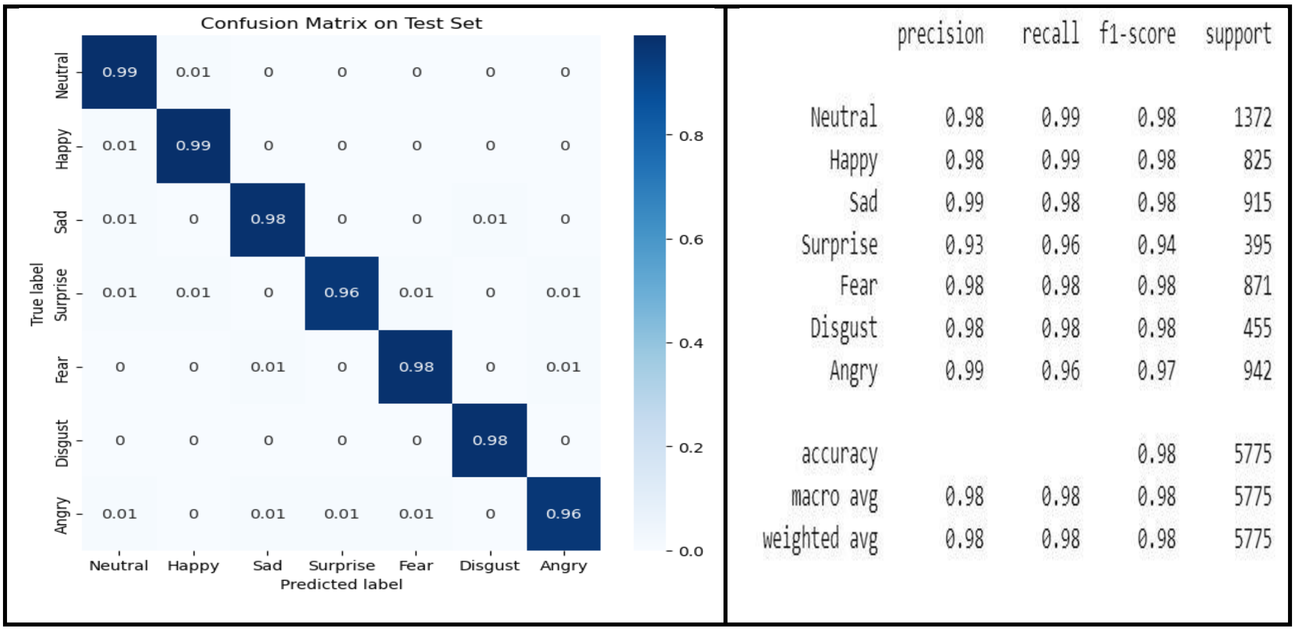


Figure 4.6: Accuracy and loss after fine-tuning the feature extraction and classification parts for this case



	Neutral	Happy	Sad	Surprise	Fear	Disgust	Angry
Neutral	0.99	0.01	0	0	0	0	0
Happy	0.01	0.99	0	0	0	0	0
Sad	0.01	0	0.98	0	0	0.01	0
Surprise	0.01	0.01	0	0.96	0.01	0	0.01
Fear	0	0	0.01	0	0.98	0	0.01
Disgust	0	0	0	0	0	0.98	0
Angry	0.01	0	0.01	0.01	0.01	0	0.96

```
              precision    recall  f1-score   support

     Neutral       0.98      0.99      0.98      1372
       Happy       0.98      0.99      0.98       825
         Sad       0.99      0.98      0.98       915
    Surprise       0.93      0.96      0.94       395
        Fear       0.98      0.98      0.98       871
     Disgust       0.98      0.98      0.98       455
       Angry       0.99      0.96      0.97       942

    accuracy                           0.98      5775
   macro avg       0.98      0.98      0.98      5775
weighted avg       0.98      0.98      0.98      5775
```

Figure 4.7: The classification report and confusion matrix for this case

Figure 4.8 shows the model accuracy and loss when one frame is taken every twenty frames (skip_no=20). Figure 4.9 shows the classification report and confusion matrix of this case with accuracy 96%. It is less when we take skip_no =10.

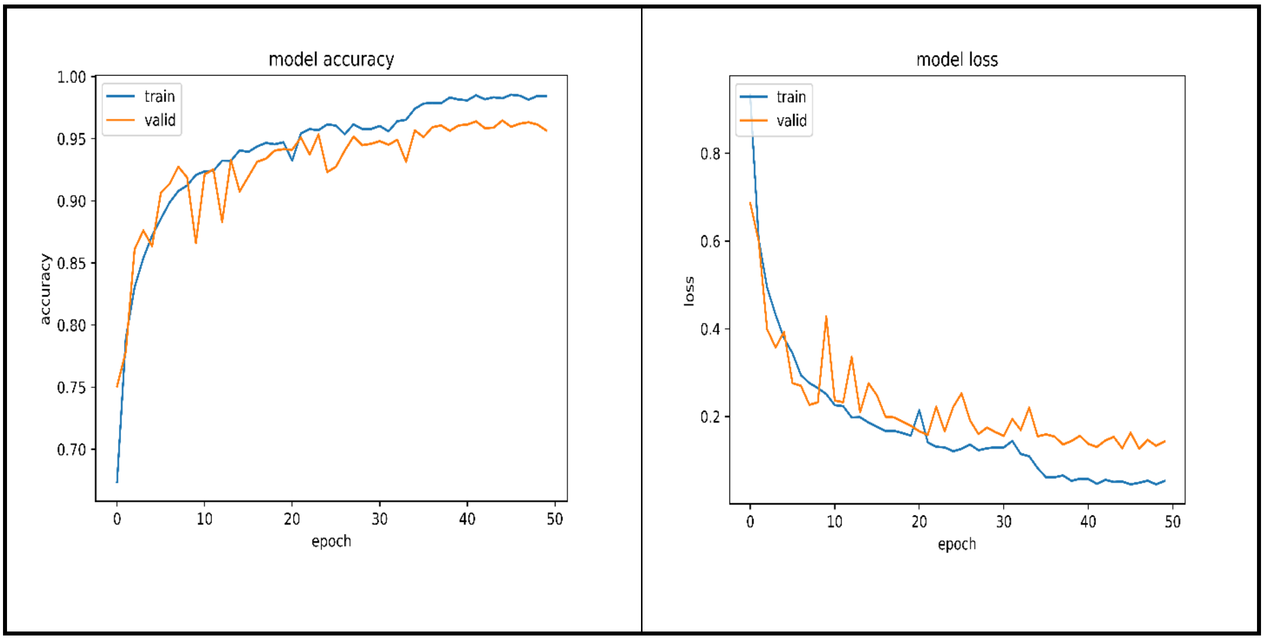
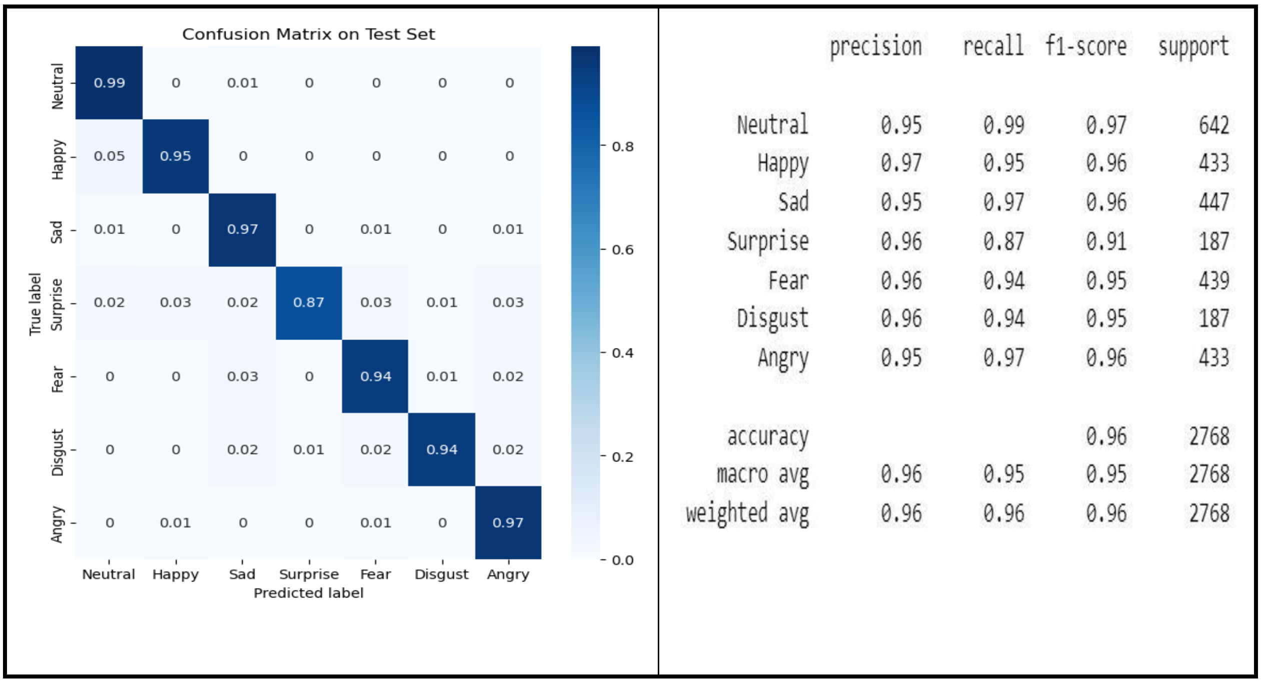


Figure 4.8: Model accuracy and loss when one frame is taken every twenty frames



Confusion Matrix on Test Set

True label \ Predicted label	Neutral	Happy	Sad	Surprise	Fear	Disgust	Angry
Neutral	0.99	0	0.01	0	0	0	0
Happy	0.05	0.95	0	0	0	0	0
Sad	0.01	0	0.97	0	0.01	0	0.01
Surprise	0.02	0.03	0.02	0.87	0.03	0.01	0.03
Fear	0	0	0.03	0	0.94	0.01	0.02
Disgust	0	0	0.02	0.01	0.02	0.94	0.02
Angry	0	0.01	0	0	0.01	0	0.97

```
              precision    recall  f1-score   support

     Neutral       0.95      0.99      0.97       642
       Happy       0.97      0.95      0.96       433
         Sad       0.95      0.97      0.96       447
    Surprise       0.96      0.87      0.91       187
        Fear       0.96      0.94      0.95       439
     Disgust       0.96      0.94      0.95       187
       Angry       0.95      0.97      0.96       433

    accuracy                           0.96      2768
   macro avg       0.96      0.95      0.95      2768
weighted avg       0.96      0.96      0.96      2768
```

Figure 4.9: The classification report and confusion matrix of (skip_no=20)

4.4.2 The MobilenetV2 Model

The model has been trained by two cases. In each case, the following results are achieved:

- **Case 1:** Freezing the feature extraction part and fine-tuning the classification part only during training by using the AffectNet database. Figure 4.10 illustrates the accuracy and loss of this case. The accuracy is 59% at epoch 20. The classification report and confusion matrix produced by testing the final model are illustrated in Figure 4.11.

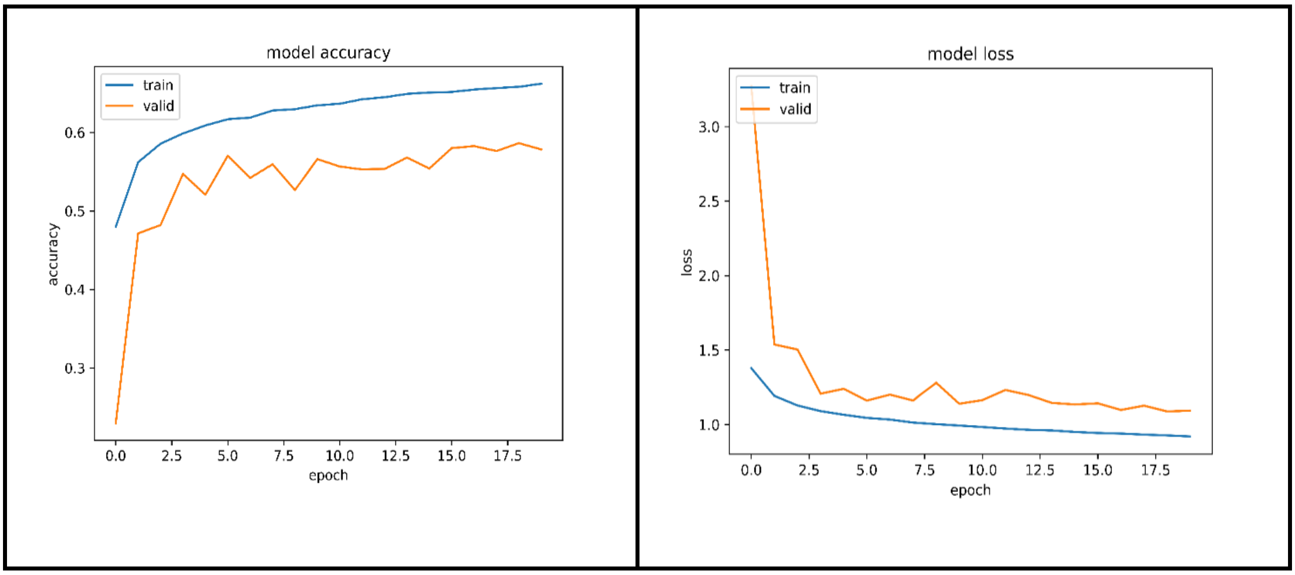
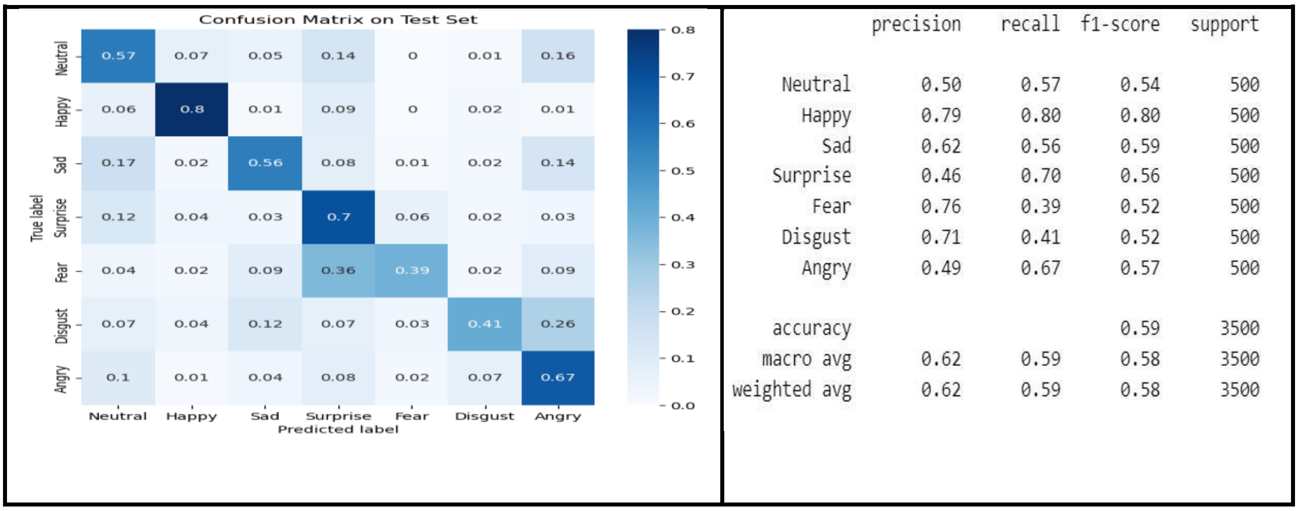


Figure 4.10: Accuracy and loss after freezing the feature extraction part in this case


Confusion Matrix on Test Set

True label \ Predicted label	Neutral	Happy	Sad	Surprise	Fear	Disgust	Angry
Neutral	0.57	0.07	0.05	0.14	0	0.01	0.16
Happy	0.06	0.8	0.01	0.09	0	0.02	0.01
Sad	0.17	0.02	0.56	0.08	0.01	0.02	0.14
Surprise	0.12	0.04	0.03	0.7	0.06	0.02	0.03
Fear	0.04	0.02	0.09	0.36	0.39	0.02	0.09
Disgust	0.07	0.04	0.12	0.07	0.03	0.41	0.26
Angry	0.1	0.01	0.04	0.08	0.02	0.07	0.67

```
              precision    recall  f1-score   support

     Neutral       0.50      0.57      0.54       500
       Happy       0.79      0.80      0.80       500
         Sad       0.62      0.56      0.59       500
    Surprise       0.46      0.70      0.56       500
        Fear       0.76      0.39      0.52       500
     Disgust       0.71      0.41      0.52       500
       Angry       0.49      0.67      0.57       500

    accuracy                           0.59      3500
   macro avg       0.62      0.59      0.58      3500
weighted avg       0.62      0.59      0.58      3500
```

Figure 4.11: The classification report and confusion matrix after freezing in this case

The model accuracy and loss for training by fine-tuning both the feature extraction and classification parts by using the AffectNet database (Unfreeze of parts) are demonstrates in Figure 4.12. The accuracy is 62% at epoch 20. Figure 4.13 depicts the classification report and confusion matrix generated by testing the final model.

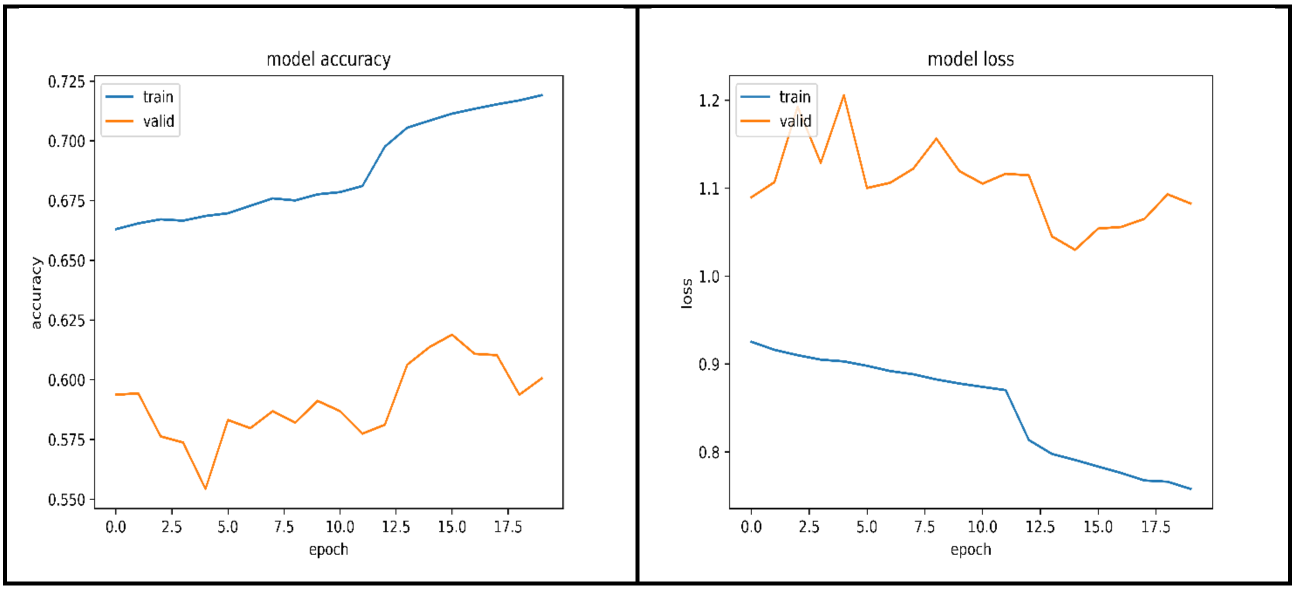
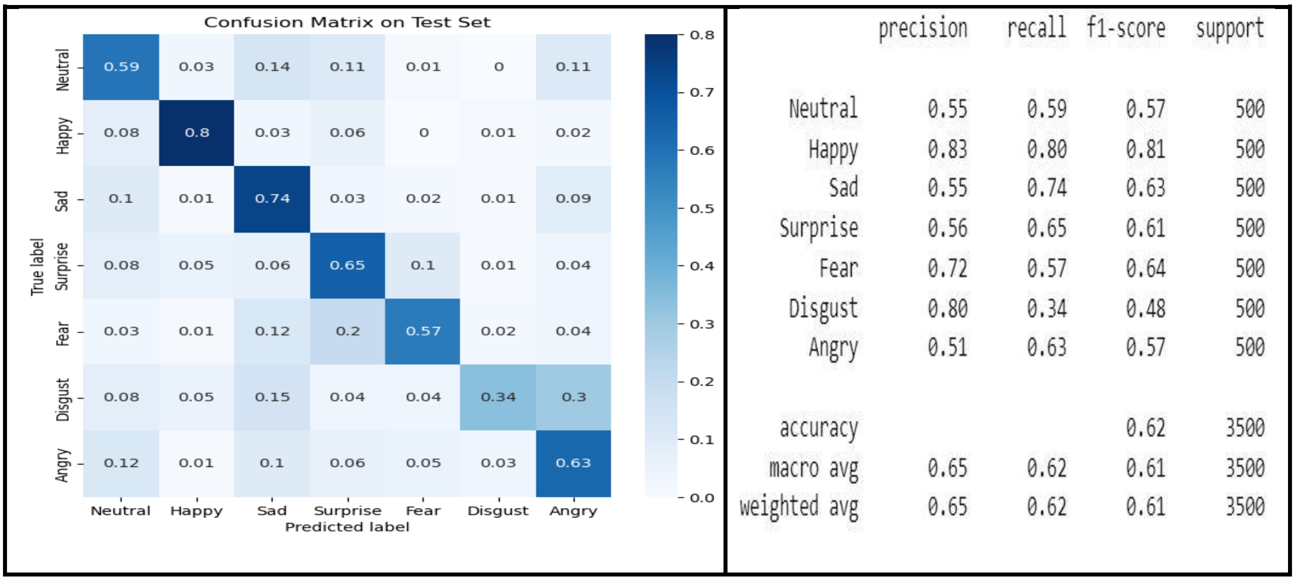


Figure 4.12: Accuracy and loss after fine-tuning the feature extraction and classification part



```
              precision    recall  f1-score   support

     Neutral       0.55      0.59      0.57       500
       Happy       0.83      0.80      0.81       500
         Sad       0.55      0.74      0.63       500
    Surprise       0.56      0.65      0.61       500
        Fear       0.72      0.57      0.64       500
     Disgust       0.80      0.34      0.48       500
       Angry       0.51      0.63      0.57       500

    accuracy                           0.62      3500
   macro avg       0.65      0.62      0.61      3500
weighted avg       0.65      0.62      0.61      3500
```

Figure 4.13: The classification report and confusion matrix

- **Case 2:** Training by fine-tuning both the feature extraction and classification parts by using the RAVDESS database (Unfreeze of parts). Figure 4.14 demonstrates the model accuracy and loss for this case. The accuracy is 95 % at epoch 50. Figure 4.15 demonstrates the classification report and confusion matrix generated by testing the final model.

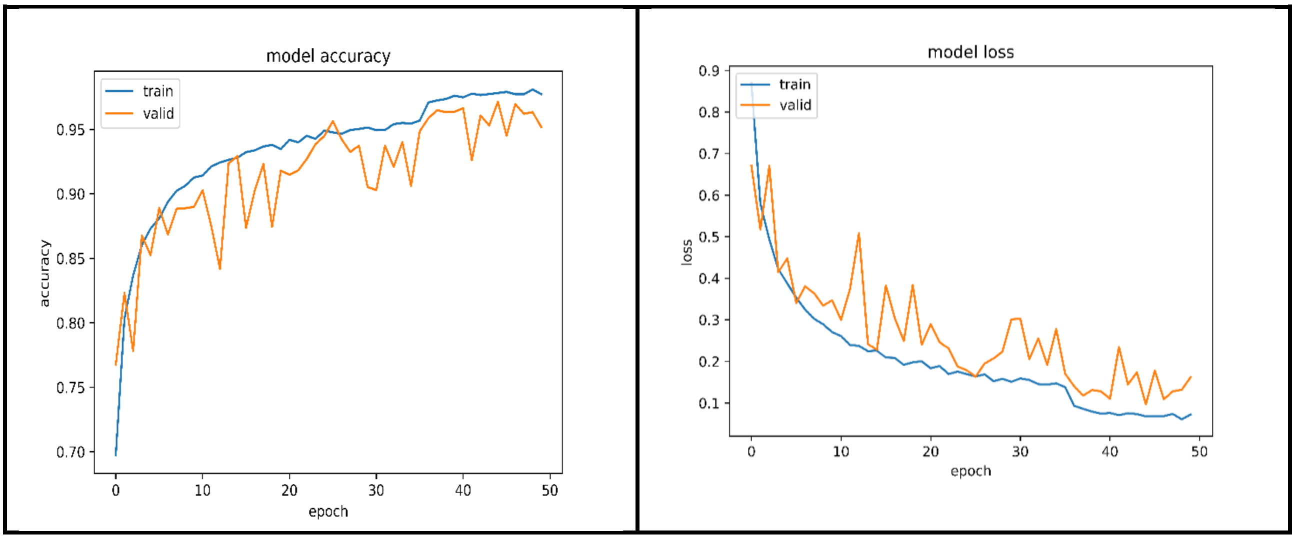
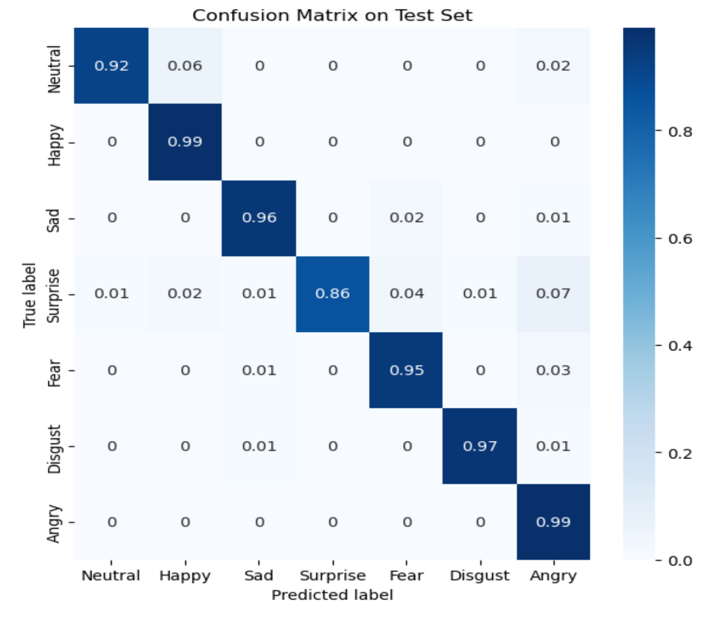
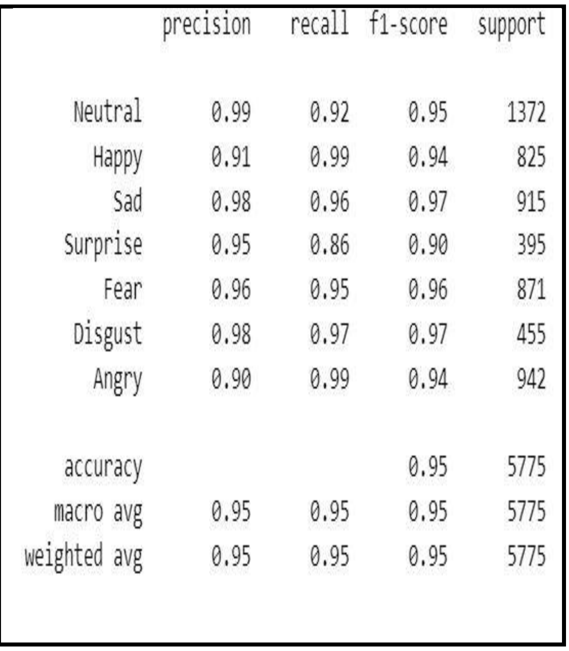


Figure 4.14: Accuracy and loss after fine-tuning the feature extraction and classification parts in this case



	Neutral	Happy	Sad	Surprise	Fear	Disgust	Angry
Neutral	0.92	0.06	0	0	0	0	0.02
Happy	0	0.99	0	0	0	0	0
Sad	0	0	0.96	0	0.02	0	0.01
Surprise	0.01	0.02	0.01	0.86	0.04	0.01	0.07
Fear	0	0	0.01	0	0.95	0	0.03
Disgust	0	0	0.01	0	0	0.97	0.01
Angry	0	0	0	0	0	0	0.99



```
              precision    recall  f1-score   support

     Neutral       0.99      0.92      0.95      1372
       Happy       0.91      0.99      0.94       825
         Sad       0.98      0.96      0.97       915
    Surprise       0.95      0.86      0.90       395
        Fear       0.96      0.95      0.96       871
     Disgust       0.98      0.97      0.97       455
       Angry       0.90      0.99      0.94       942

    accuracy                           0.95      5775
   macro avg       0.95      0.95      0.95      5775
weighted avg       0.95      0.95      0.95      5775
```

Figure 4.15: The classification report and confusion matrix of this case

4.4.3 The ResNet50 Model

The model has been trained by three cases. In each case, the following results are achieved:

- **Case 1:** Freezing the feature extraction part and fine-tuning the classification part only during training by using the AffectNet database. Figure 4.16 illustrates the accuracy and loss of this case. The accuracy is 54% at epoch 20. The classification report and confusion matrix produced by testing the final model are illustrated in Figure 4.17.

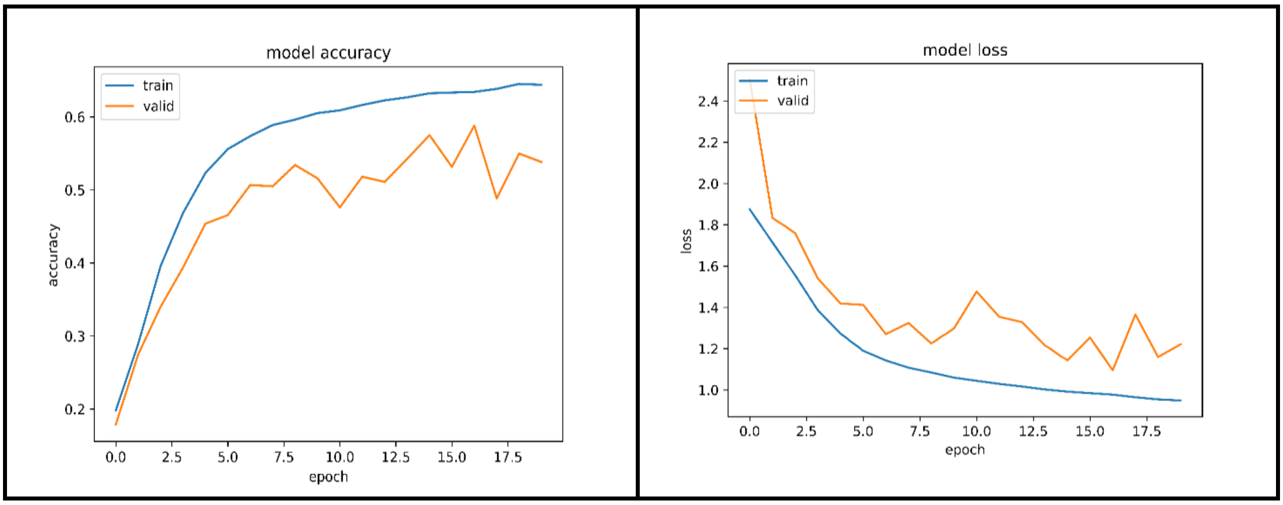
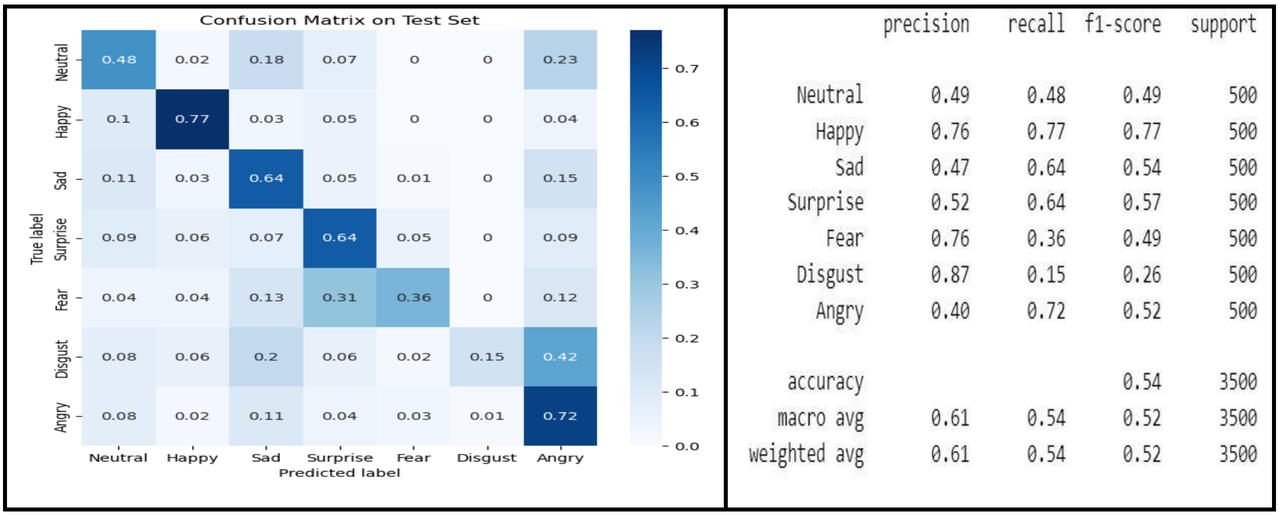


Figure 4.16: Accuracy and loss after freezing the feature extraction part in this case



	Neutral	Happy	Sad	Surprise	Fear	Disgust	Angry
Neutral	0.48	0.02	0.18	0.07	0	0	0.23
Happy	0.1	0.77	0.03	0.05	0	0	0.04
Sad	0.11	0.03	0.64	0.05	0.01	0	0.15
Surprise	0.09	0.06	0.07	0.64	0.05	0	0.09
Fear	0.04	0.04	0.13	0.31	0.36	0	0.12
Disgust	0.08	0.06	0.2	0.06	0.02	0.15	0.42
Angry	0.08	0.02	0.11	0.04	0.03	0.01	0.72

```
              precision    recall  f1-score   support

     Neutral       0.49      0.48      0.49       500
       Happy       0.76      0.77      0.77       500
         Sad       0.47      0.64      0.54       500
    Surprise       0.52      0.64      0.57       500
        Fear       0.76      0.36      0.49       500
     Disgust       0.87      0.15      0.26       500
       Angry       0.40      0.72      0.52       500

    accuracy                           0.54      3500
   macro avg       0.61      0.54      0.52      3500
weighted avg       0.61      0.54      0.52      3500
```

Figure 4.17: The classification report and confusion matrix after freezing in this case

The model accuracy and loss for training by fine-tuning both the feature extraction and classification parts by using the AffectNet database (Unfreeze of parts) are demonstrates in Figure 4.18. The accuracy is 58% at epoch 20. Figure 4.19 presents the classification report and confusion matrix generated by testing the final model.

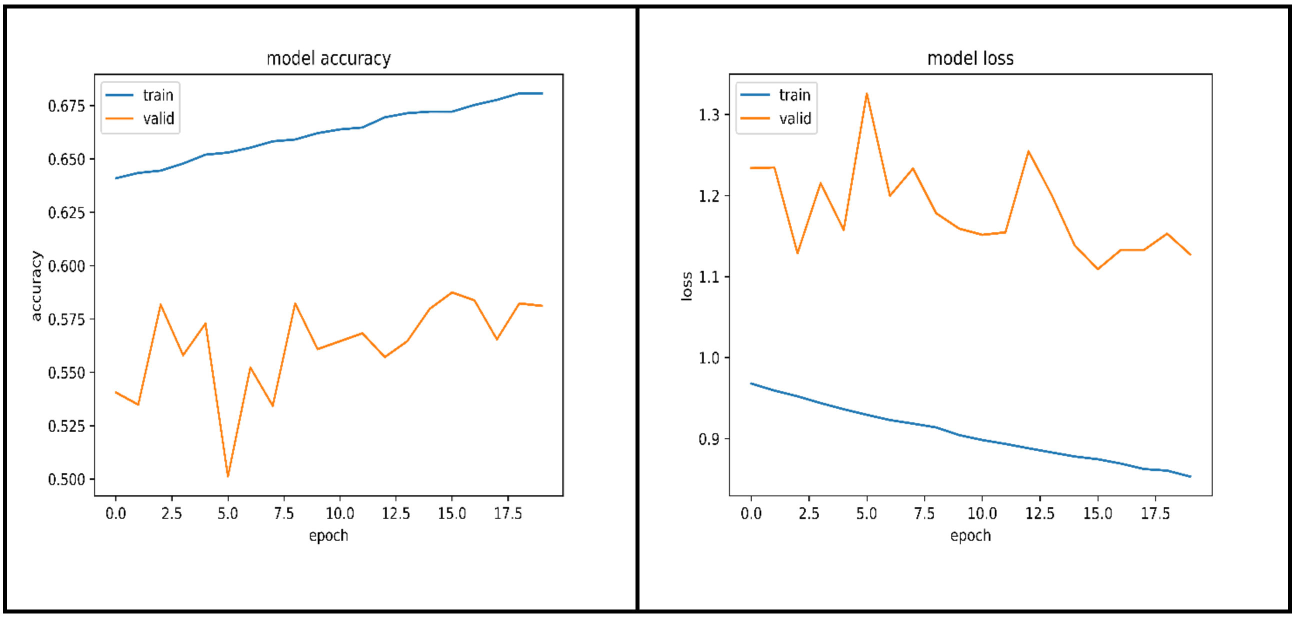
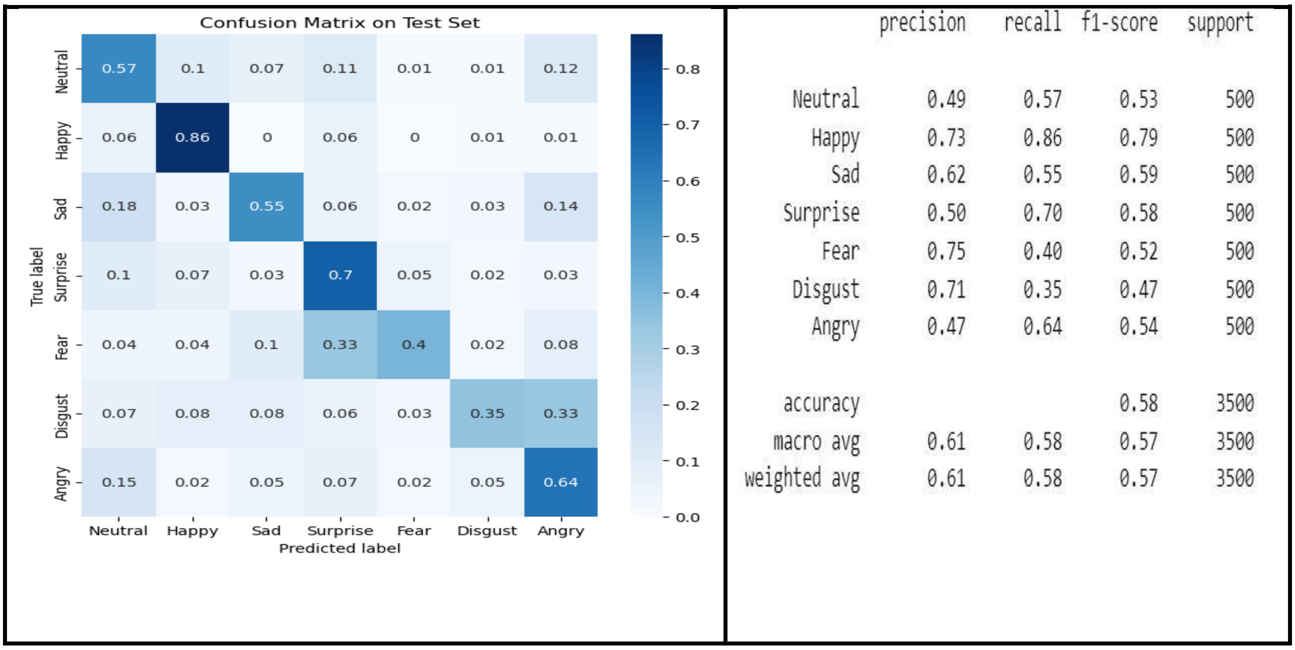


Figure 4.18: Accuracy and loss after fine-tuning the feature extraction and classification parts



	Neutral	Happy	Sad	Surprise	Fear	Disgust	Angry
Neutral	0.57	0.1	0.07	0.11	0.01	0.01	0.12
Happy	0.06	0.86	0	0.06	0	0.01	0.01
Sad	0.18	0.03	0.55	0.06	0.02	0.03	0.14
Surprise	0.1	0.07	0.03	0.7	0.05	0.02	0.03
Fear	0.04	0.04	0.1	0.33	0.4	0.02	0.08
Disgust	0.07	0.08	0.08	0.06	0.03	0.35	0.33
Angry	0.15	0.02	0.05	0.07	0.02	0.05	0.64

```
              precision    recall  f1-score   support

     Neutral       0.49      0.57      0.53       500
       Happy       0.73      0.86      0.79       500
         Sad       0.62      0.55      0.59       500
    Surprise       0.50      0.70      0.58       500
        Fear       0.75      0.40      0.52       500
     Disgust       0.71      0.35      0.47       500
       Angry       0.47      0.64      0.54       500

    accuracy                           0.58      3500
   macro avg       0.61      0.58      0.57      3500
weighted avg       0.61      0.58      0.57      3500
```

Figure 4.19: The classification report and confusion matrix

- **Case 2:** Training by fine-tuning both the feature extraction and classification parts by using the RAVDESS database (Unfreeze of parts). Figure 4.20 demonstrates the model accuracy and loss for this case. The accuracy is 96 % at epoch 50. Figure 4.21 depicts the classification report and confusion matrix generated by testing the final model.

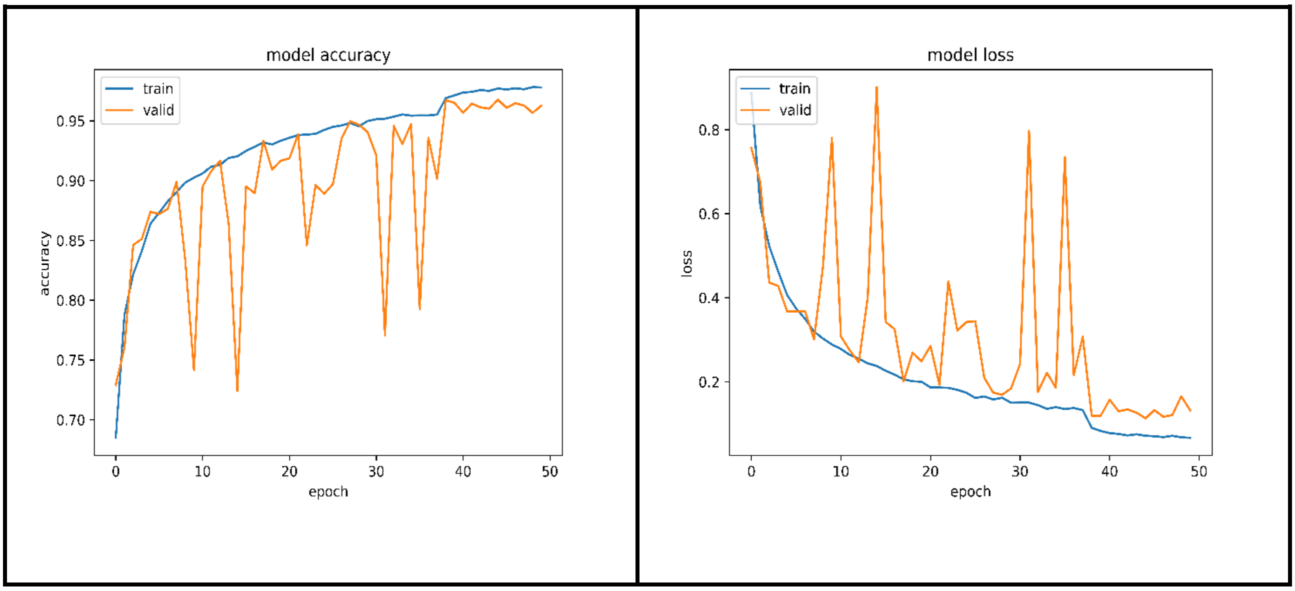
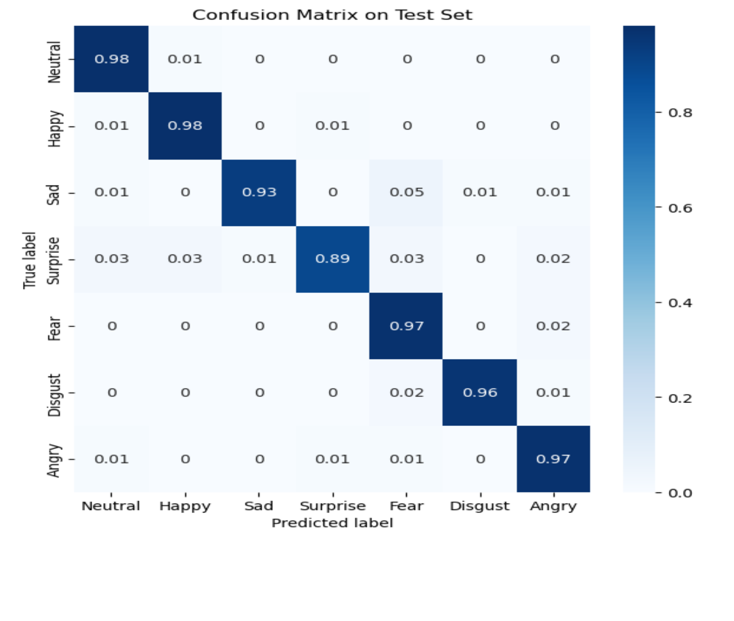
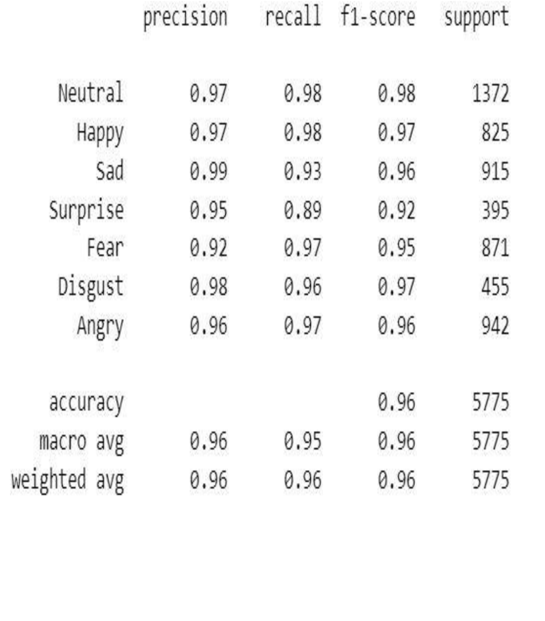


Figure 4.20: Accuracy and loss after fine-tuning the feature extraction and classification parts in this case



	Neutral	Happy	Sad	Surprise	Fear	Disgust	Angry
Neutral	0.98	0.01	0	0	0	0	0
Happy	0.01	0.98	0	0.01	0	0	0
Sad	0.01	0	0.93	0	0.05	0.01	0.01
Surprise	0.03	0.03	0.01	0.89	0.03	0	0.02
Fear	0	0	0	0	0.97	0	0.02
Disgust	0	0	0	0	0.02	0.96	0.01
Angry	0.01	0	0	0.01	0.01	0	0.97



```
              precision    recall  f1-score   support

     Neutral       0.97      0.98      0.98      1372
       Happy       0.97      0.98      0.97       825
         Sad       0.99      0.93      0.96       915
    Surprise       0.95      0.89      0.92       395
        Fear       0.92      0.97      0.95       871
     Disgust       0.98      0.96      0.97       455
       Angry       0.96      0.97      0.96       942

    accuracy                           0.96      5775
   macro avg       0.96      0.95      0.96      5775
weighted avg       0.96      0.96      0.96      5775
```

Figure 4.21: The classification report and confusion matrix of this case

From the previous results of the visual path, we conclude that the best proposed model based on transfer learning is the first model (EfficientNetB0) because of its high-performance metrics, as shown in Figure 4.22.

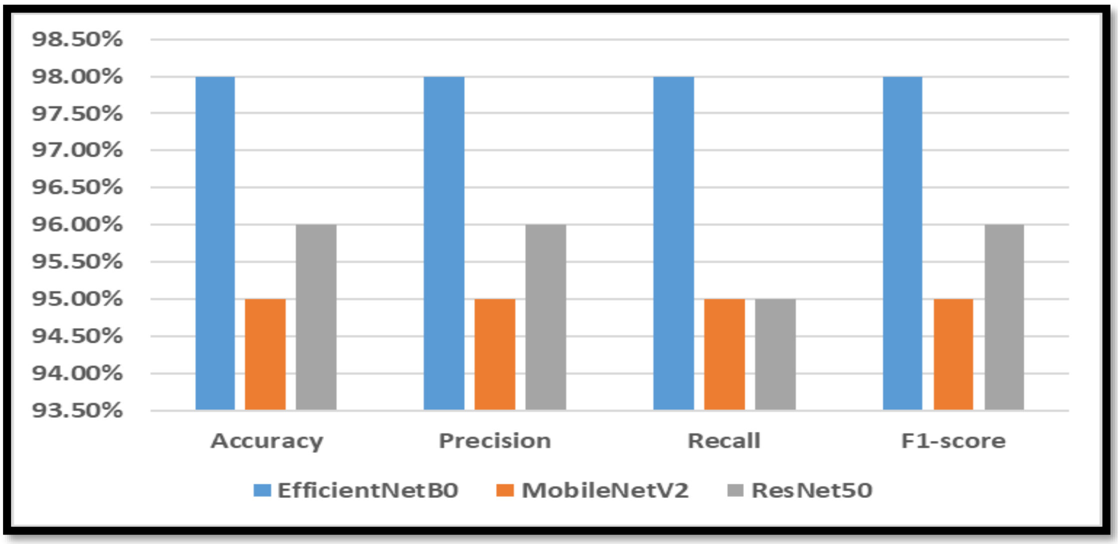


Figure 4.22: The performance metrics of the proposed models for the visual path

4.5 Training and Testing the Proposed Model for Audio Path

The parameters used for the purpose of training the proposed MLP model on the audio data are shown in Table 4.3 and Table 4.4 shows a summary of the architecture of the proposed model of audio with the number of neurons and parameters in each layer.

Table 4.3: The parameters for training the proposed model of audio

Parameter	Value
Input shape	(162,)
No of Epochs	800
Batch Size	32
Dropout Rate	0.2
Learning Rate	0.001
Optimization function	ADAM

Table 4.4 : Summary of the architecture of the proposed model of audio with the number of neurons and parameters in each layer

Layer name	Output Shape	Number of Parameters
Input layer	(None, 162)	0
Dense 1	(None, 256)	41728
Dropout	(None, 256)	0
Dense 2	(None, 128)	32896
Dropout	(None, 128)	0
Dense 3	(None, 128)	16512
Dense 4	(None, 64)	8256
Dense 5	(None, 32)	2080
Output layer	(None, 7)	231

The Training is done by fine-tuning both the feature extraction and classification parts by using the RAVDESS database (Unfreeze of parts). Figure 4.23 demonstrates the model accuracy and loss. The accuracy is 69 % at epoch 800. The classification report and confusion matrix generated by testing the final model are displayed in Figure 4.24.

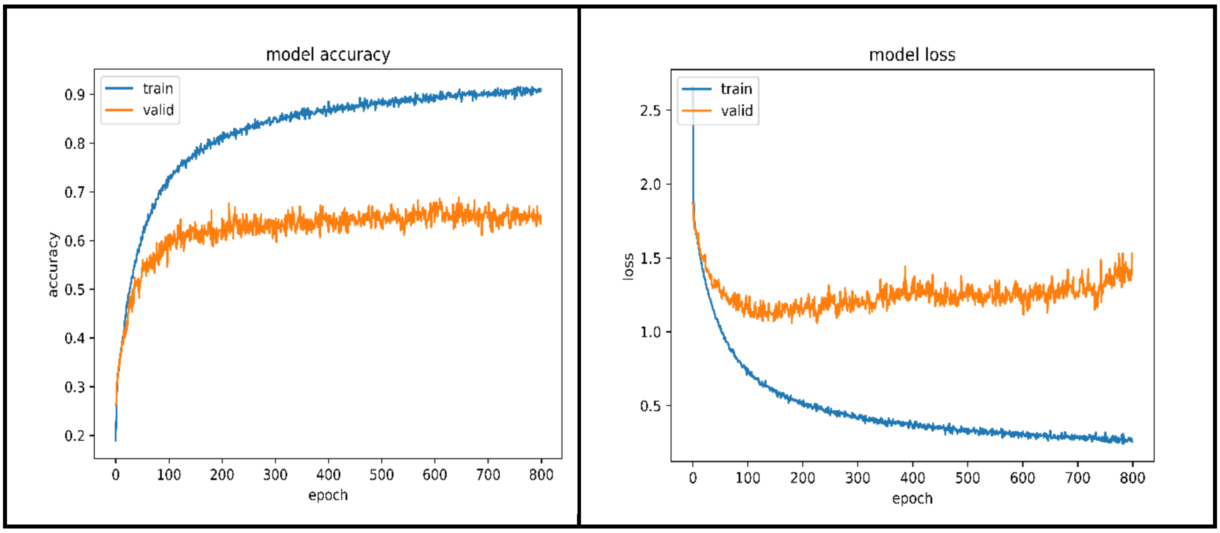
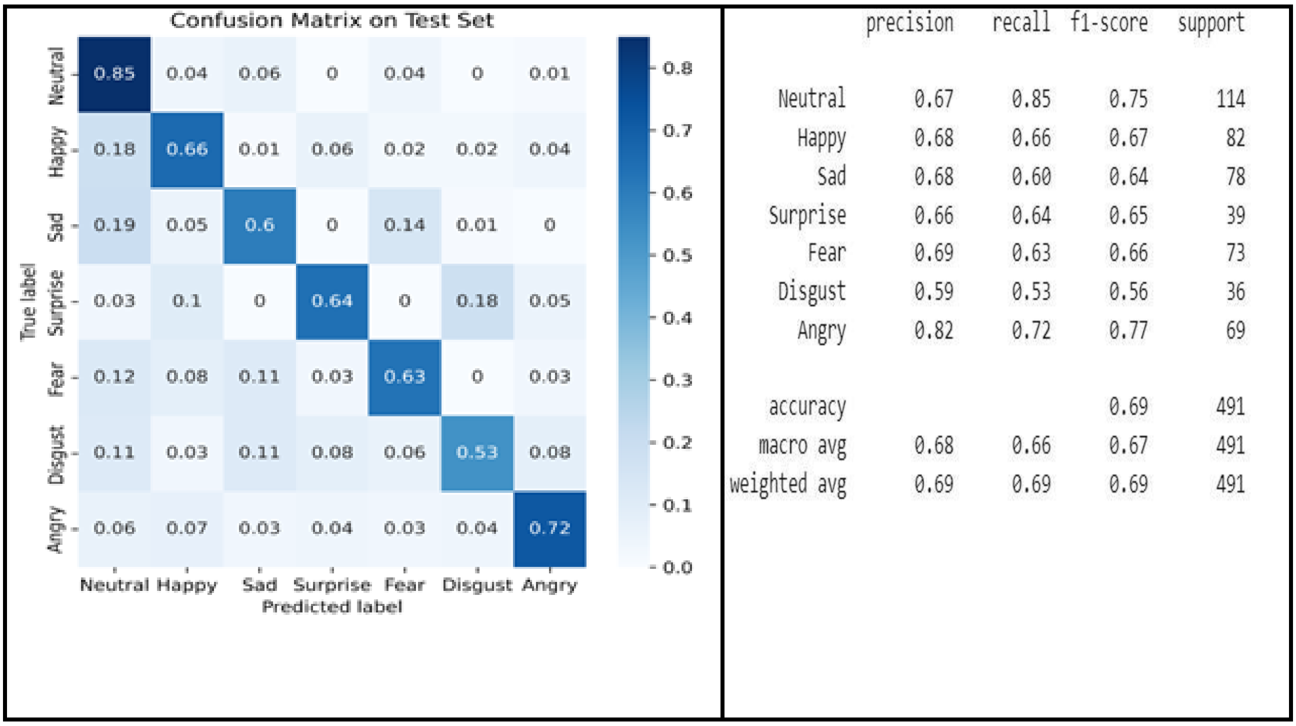


Figure 4.23: Accuracy and loss after finetuning the whole proposed audio model



Confusion Matrix on Test Set

True label \ Predicted label	Neutral	Happy	Sad	Surprise	Fear	Disgust	Angry
Neutral	0.85	0.04	0.06	0	0.04	0	0.01
Happy	0.18	0.66	0.01	0.06	0.02	0.02	0.04
Sad	0.19	0.05	0.6	0	0.14	0.01	0
Surprise	0.03	0.1	0	0.64	0	0.18	0.05
Fear	0.12	0.08	0.11	0.03	0.63	0	0.03
Disgust	0.11	0.03	0.11	0.08	0.06	0.53	0.08
Angry	0.06	0.07	0.03	0.04	0.03	0.04	0.72

```
              precision    recall  f1-score   support

     Neutral       0.67      0.85      0.75       114
       Happy       0.68      0.66      0.67        82
         Sad       0.68      0.60      0.64        78
    Surprise       0.66      0.64      0.65        39
        Fear       0.69      0.63      0.66        73
     Disgust       0.59      0.53      0.56        36
       Angry       0.82      0.72      0.77        69

    accuracy                           0.69       491
   macro avg       0.68      0.66      0.67       491
weighted avg       0.69      0.69      0.69       491
```

Figure 4.24: The classification report and confusion matrix of audio model

Figure 4.25 shows the accuracy and loss without using augmentation process of audio data and Figure 4.26 demonstrates the classification report and confusion matrix generated by testing the final model without using augmentation process which shows that the accuracy has become 38% due to the lack of trained data.

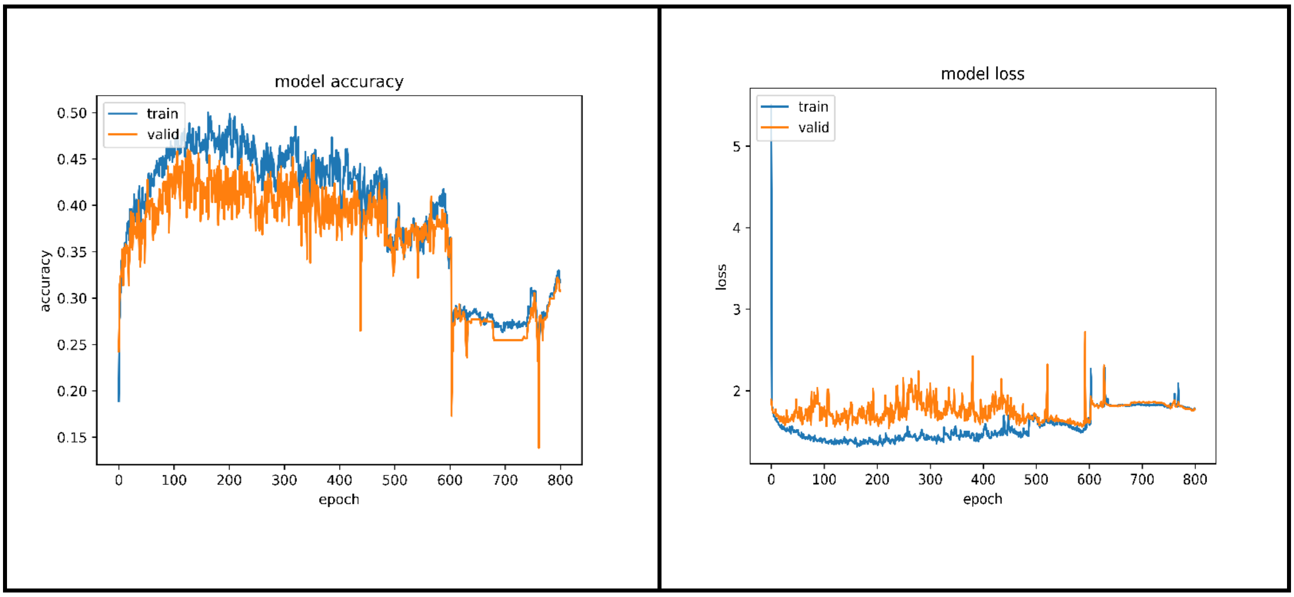
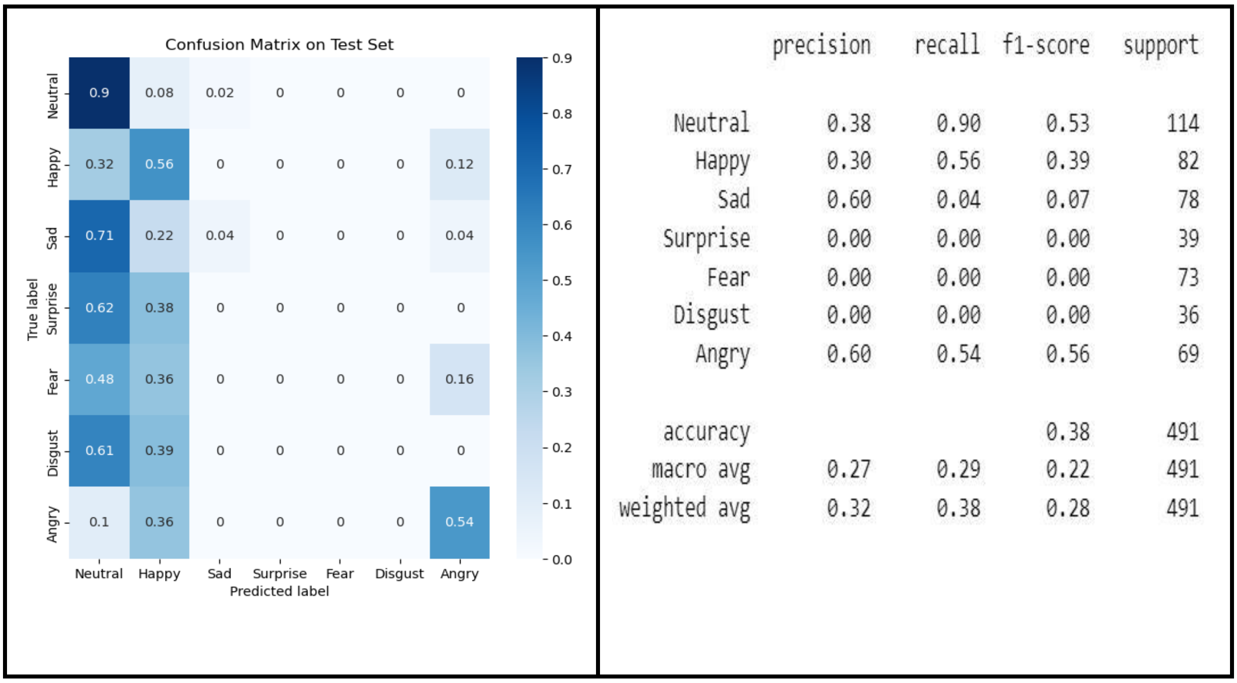


Figure 4.25: The accuracy and loss without using augmentation process



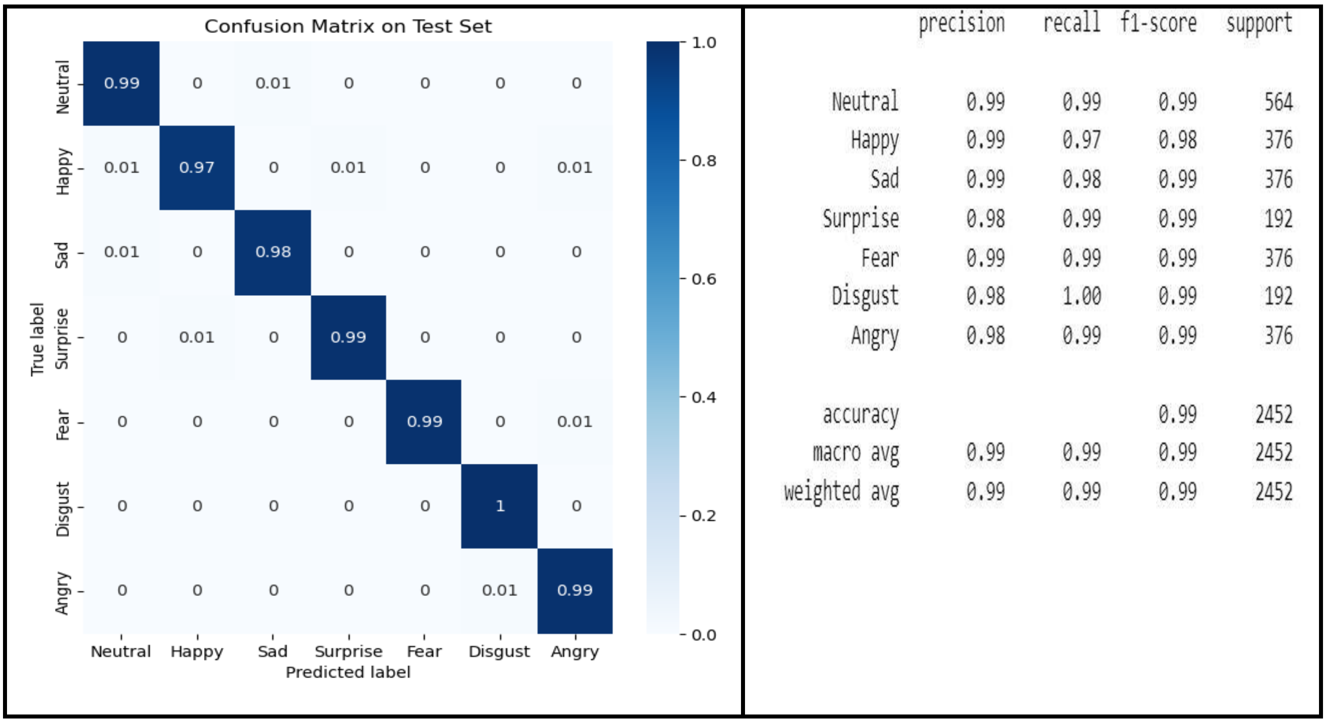
```
              precision    recall  f1-score   support

     Neutral       0.38      0.90      0.53       114
       Happy       0.30      0.56      0.39        82
         Sad       0.60      0.04      0.07        78
    Surprise       0.00      0.00      0.00        39
        Fear       0.00      0.00      0.00        73
     Disgust       0.00      0.00      0.00        36
       Angry       0.60      0.54      0.56        69

    accuracy                           0.38       491
   macro avg       0.27      0.29      0.22       491
weighted avg       0.32      0.38      0.28       491
```

Figure 4.26: The classification report and confusion matrix

4.6 Final Visual -Audio Fusion Prediction Results

At the decision-level, the visual and audio results are combined. A weighted average majority voting method is used to fuse the results of the visual and audio predictions. The results from the visual and audio predictors complement each other. Thus, the fused results are considered better than the individual results. The fusion using ensemble decision method gives an accuracy of 99% for the RAVDESS database by using Algorithm 3.9.

A classification report and confusion matrix were obtained when a weight of 0.5 was assumed for both visual and auditory cases, as shown in Figure 4.27.



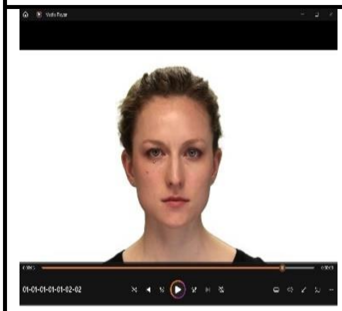
	Neutral	Happy	Sad	Surprise	Fear	Disgust	Angry
Neutral	0.99	0	0.01	0	0	0	0
Happy	0.01	0.97	0	0.01	0	0	0.01
Sad	0.01	0	0.98	0	0	0	0
Surprise	0	0.01	0	0.99	0	0	0
Fear	0	0	0	0	0.99	0	0.01
Disgust	0	0	0	0	0	1	0
Angry	0	0	0	0	0	0.01	0.99

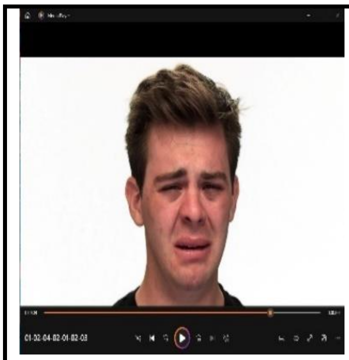
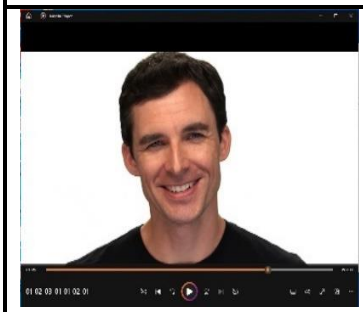
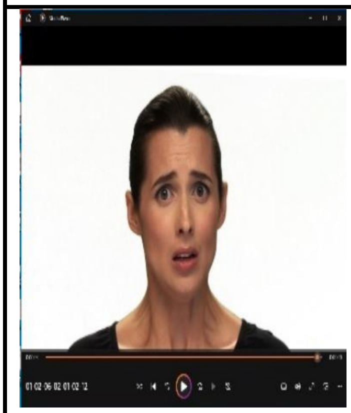
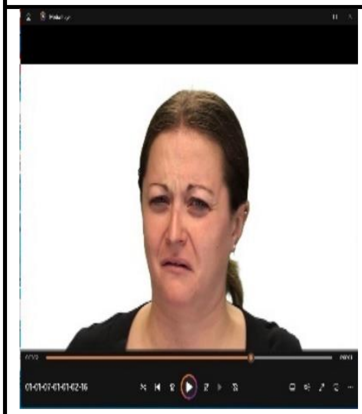
```
              precision    recall  f1-score   support

     Neutral       0.99      0.99      0.99       564
       Happy       0.99      0.97      0.98       376
         Sad       0.99      0.98      0.99       376
    Surprise       0.98      0.99      0.99       192
        Fear       0.99      0.99      0.99       376
     Disgust       0.98      1.00      0.99       192
       Angry       0.98      0.99      0.99       376

    accuracy                           0.99      2452
   macro avg       0.99      0.99      0.99      2452
weighted avg       0.99      0.99      0.99      2452
```

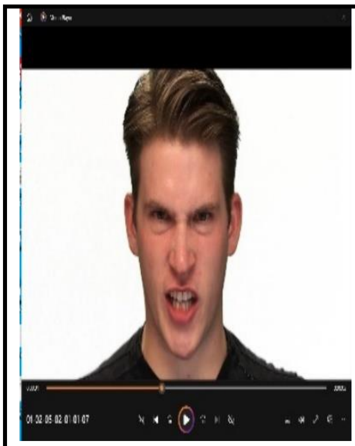
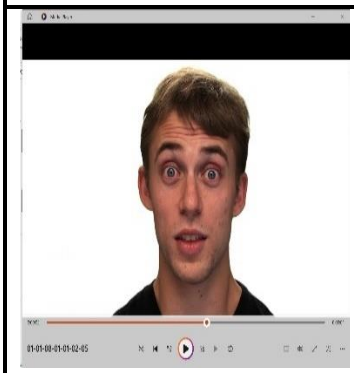
Figure 4.27: The classification report and confusion matrix for fusion by the ensemble decision method

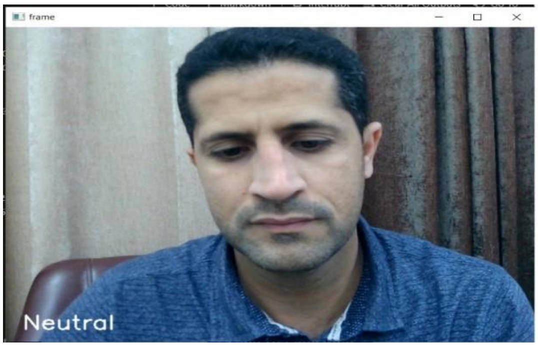
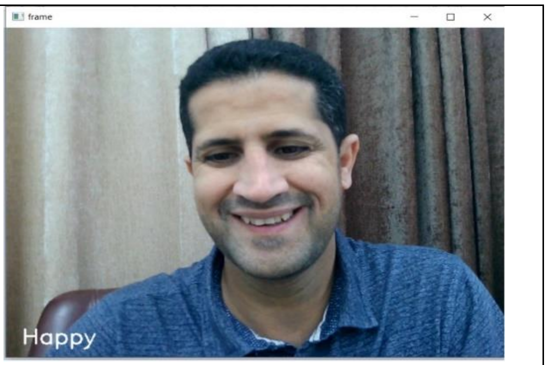
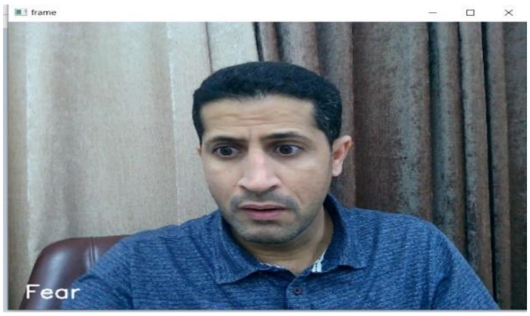
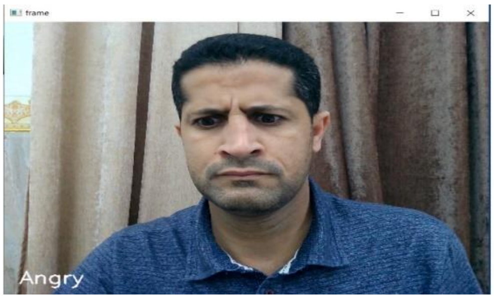
Table 4.5 demonstrates the samples of testing some of the videos of emotions from the RAVDESS database. The Figure 4.28 demonstrated some results of the real-time environment.

Table 4.5: The samples of testing some of the videos of emotions

Videos	Predicted emotion for Visual path	Predicted emotion for Audio path	Predicted emotion for fusion
	Neutral	Neutral	Neutral

	Sad	Sad	Sad Low- Arousal Emotion
	Happy	Happy	Happy High-Arousal Emotion
	Fear	Fear	Fear High-Arousal Emotion
	Disgust	Disgust	Disgust Low-Arousal Emotion

	Angry	Angry	Angry High-Arousal Emotion
	Surprise	Surprise	Surprise High-Arousal Emotion

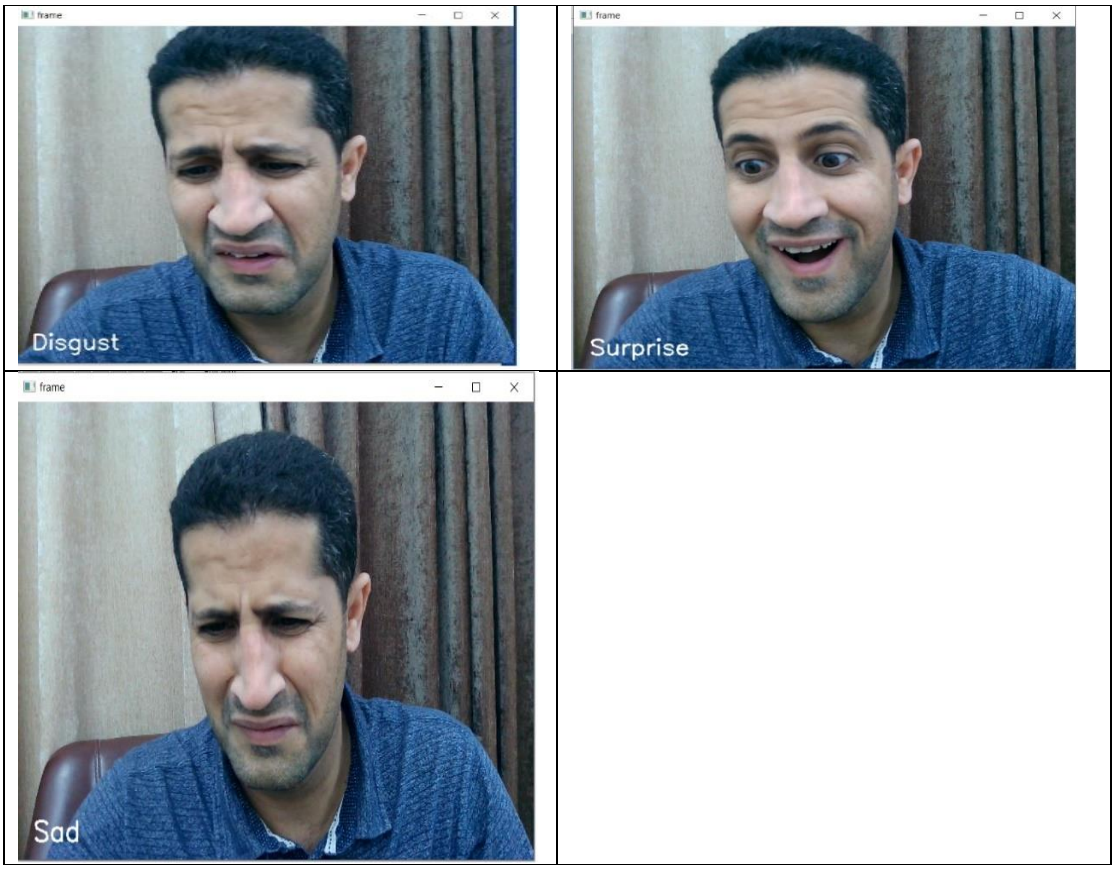


Figure 4.28: Some of real-time results

4.7 Comparison between our model and state of the art

The comparison of the achieved results by the proposed system and the state-of-art works for the database is provided in Table 4.6. As clearly shown in the comparison table, the proposed system outperformed the results achieved by the state-of-the-arts with the following points:

1. The achieved accuracy is enhanced for RAVDESS database Comparing all the works mentioned in the Table 4.6.
2. The proposed system has been used on two different databases with different sizes and conditions. This reflects the flexibility of the proposed system and the reliability of the extracted audio and visual features that can recognize the emotion regardless to the intensity of expressions, the presence of noise and backgrounds as in Figure 4.28.

3. The proposed system is based on lightweight features that takes few seconds to be extracted which makes the response time as small as possible.

Table 4.6: A comparison between current research works and the proposed system based on RAVDESS database

Authers	Methodology	Database	Accuracy
Jaratrotkamjorn and Choksuriwong [24].	Gabor filter bank to extract 68 facial features, they also used the pyAudioAnalysis library to extract 34 speech features, feature-level fusion to audio& facial features, Deep belief network for the classification.	RAVDESS Database	97.92%.
Siddiqui and Javaid [26].	Two CNNs were trained using visible and infrared images, Transfer Learning to extract features from the image, and feature-level fusion to fuse the features for classification. Third CNN extracts audio spectrogram features, SVM and third CNN fed to decision-level fusion.	RAVDESS Database	86.36%.
Luna-Jiménez et al. [28]	CNN-14 of the Large-Scale Pretrained Audio Neural Networks (PANNs) framework for speech path. Pre-trained Spatial Transformer Network on saliency maps and facial images followed by an LSTM with an attention mechanism for facial emotion. The late fusion strategy to fuse the two stages.	RAVDESS Database	80.08%
Middya et al. [30].	CNN as video features extractor,1-D CNN for audio mfccs, melspectrogram, spectral contrast and tonnetz features extractor, fusing audio and video features at the model level.	RAVDESS & SAVEE Database	86 % 99%
Puri et al. [29].	CNN contain 8 layers to classify Log Mel Spectrogram and Mel-Frequency Cepstral Coefficients (MFCCs) audio features	RAVDESS Database	98%
Oud Model	**CNN to extract the features with a new classifier for the visual path, MLP to classify 5 types of features for the audio path, and the decision-level fusion strategy to fuse the two paths.**	**RAVDESS Database**	**99%**

4.8 Estimated Execution Time Computation

Table 4.7 demonstrate the sample of estimated execution time consumed by the proposed system in visual path, audio path, and fusion path.

Table 4.7: The sample of estimated execution time consumed by the proposed system

Visual-path (sec)	Audio-path (sec)	Fusion (sec)
0.236	0.039	0.453

Chapter Five

Conclusions and Future Works

5.1 Conclusions

Due to the performance of our system, we can conclude the following points:

1. Deep learning required massive amounts of resources. Algorithms 3.3 and Algorithm 3.8 were employed to decrease the training time and the required resources. There is no need to read all image frames and audio directly. This is because reading one large file is faster than reading several files.
2. If the model is already trained on a database similar or close to the one on which the training is required, the performance will become more effective because the weights are set correctly to extract the best features. This is proved by using the results from training model using weights from network pre-trained on ImageNet dataset and those obtained from models trained separately on AffectNet dataset as in Figure 4.2, Figure 4.10 and Figure 4.16.
3. It is necessary to fine-tune the part of feature extraction of pre-trained models to obtain good accuracy when using them for different dataset.
4. The best accuracy was obtained in visual path by using the EfficientNetB0 model as feature extractor which is accuracy 98 %.
5. Using information sources is better than a single source to obtain high accuracy. This is proved by combing the visual and audio sources to obtain the best performance with recognition rates equal to 99 % in ensemble decision method as Figure 4.27.
6. The suggested approach can identify emotions despite of noise, expression strength and other effects as in Figure 4.28.

5.2 Suggestions for future works

1. The number of key-frames can be enlarged to cover more characteristics of the emotion by combining content-based and clustering-based strategies.
2. 3D facial expression analysis can be utilized for more closely examine the fine changes in the structure of the face that occur as a result of the expressed emotion.
3. A real-world database with more than seven emotions can be collected and used to train and test system in conditions that are similar to those in the real world.
4. A multi-agent environment can be used with more than one resources can be handled as a perception time new remote sensing acquiring for visual and audio.
5. It is also recommended to find the best training parameters for the models such as batch size, training epochs, optimization functions, and so on.

References

[1] M. Spezialetti, G. Placidi, and S. Rossi, “Emotion Recognition for Human-Robot Interaction: Recent Advances and Future Perspectives,” *Frontiers in Robotics and AI*, vol. 7. Frontiers Media S.A., Dec. 21, 2020. doi: 10.3389/frobt.2020.532279.

[2] H. Bansal and R. Khan, “A Review Paper on Human Computer Interaction,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 8, no. 4, p. 53, Apr. 2018, doi: 10.23956/ijarcsse.v8i4.630.

[3] E. Politou, E. Alepis, and C. Patsakis, “A survey on mobile affective computing,” *Computer Science Review*, vol. 25. Elsevier Ireland Ltd, pp. 79–100, Aug. 01, 2017. doi: 10.1016/j.cosrev.2017.07.002.

[4] N. M. Seel, *Encyclopedia of the Sciences of Learning*. Springer Science & Business Media, 2011.

[5] E. Chandra and J. Y. Hsu, “Deep learning for multimodal emotion recognition-attentive residual disconnected RNN,” in *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, IEEE, 2019, pp. 1–8.

[6] A. Mehrabian, *Silent messages*, vol. 8, no. 152. Wadsworth Belmont, CA, 1971.

[7] D. Gharavian, M. Bejani, and M. Sheikhan, “Audio-visual emotion recognition using FCBF feature selection method and particle swarm optimization for fuzzy ARTMAP neural networks,” *Multimed Tools Appl*, vol. 76, no. 2, pp. 2331–2352, Jan. 2017, doi: 10.1007/s11042-015-3180-6.

[8] A. S. Hatem and A. M. Al-Bakry, “The Information Channels of Emotion Recognition: A Review,” *Webology*, vol. 19, no. 1, pp. 927–941, Jan. 2022, doi: 10.14704/web/v19i1/web19064.

[9] M. Ivanović *et al.*, “Emotional agents — state of the art and applications,” *Computer Science and Information Systems*, vol. 12, no. 4, pp. 1121–1148, Nov. 2015, doi: 10.2298/CSIS141026047I.

[10] D. Grandjean, D. Sander, and K. R. Scherer, “Conscious emotional experience emerges as a function of multilevel, appraisal-driven response

synchronization," *Conscious Cogn*, vol. 17, no. 2, pp. 484–495, Jun. 2008, doi: 10.1016/j.concog.2008.03.019.

[11] I. B. Mauss and M. D. Robinson, "Measures of emotion: A review," *Cognition and Emotion*, vol. 23, no. 2. pp. 209–237, 2009. doi: 10.1080/02699930802204677.

[12] T. Eerola and J. K. Vuoskoski, "A comparison of the discrete and dimensional models of emotion in music," *Psychol Music*, vol. 39, no. 1, pp. 18–49, 2011, doi: 10.1177/0305735610362821.

[13] J. Grekow, *From content-based music emotion recognition to emotion maps of musical pieces*, vol. 747. Springer, 2018.

[14] F. Alonso-Martín, M. Malfaz, J. Sequeira, J. F. Gorostiza, and M. A. Salichs, "A multimodal emotion detection system during human-robot interaction," *Sensors (Switzerland)*, vol. 13, no. 11, pp. 15549–15581, Nov. 2013, doi: 10.3390/s131115549.

[15] D. Kulíc and E. Croft, "Affective state estimation for human-robot interaction," in *IEEE Transactions on Robotics*, Oct. 2007, pp. 991–1000. doi: 10.1109/TRO.2007.904899.

[16] E. Martinez-Martin and A. P. Del Pobil, "Object detection and recognition for assistive robots: Experimentation and implementation," *IEEE Robot Autom Mag*, vol. 24, no. 3, pp. 123–138, Sep. 2017, doi: 10.1109/MRA.2016.2615329.

[17] A. Paiva, I. Leite, H. Boukricha, and I. Wachsmuth, "Empathy in virtual agents and robots: A survey," *ACM Trans Interact Intell Syst*, vol. 7, no. 3, Sep. 2017, doi: 10.1145/2912150.

[18] S. Li and W. Deng, "Deep Facial Expression Recognition: A Survey," *IEEE Trans Affect Comput*, vol. 13, no. 3, pp. 1195–1215, 2022, doi: 10.1109/TAFFC.2020.2981446.

[19] A. Savran, B. Sankur, and M. Taha Bilge, "Regression-based intensity estimation of facial action units," *Image Vis Comput*, vol. 30, no. 10, pp. 774–784, 2012, doi: 10.1016/j.imavis.2011.11.008.

[20] M. V. Mishra, S. B. Ray, and N. Srinivasan, “Cross-cultural emotion recognition and evaluation of Radboud faces database with an Indian sample,” *PLoS One*, vol. 13, no. 10, Oct. 2018, doi: 10.1371/journal.pone.0203959.

[21] G. Iatraki, “Emotional Facial Expression Recognition & Classification,” *Master Thesis, Delft University of Technology*, 2009.

[22] K. S. Song, Y. H. Nho, J. H. Seo, and D. S. Kwon, “Decision-Level Fusion Method for Emotion Recognition using Multimodal Emotion Recognition Information,” in *2018 15th International Conference on Ubiquitous Robots, UR 2018*, Institute of Electrical and Electronics Engineers Inc., Aug. 2018, pp. 472–476. doi: 10.1109/URAI.2018.8441795.

[23] S. Zhang, S. Zhang, T. Huang, W. Gao, and Q. Tian, “Learning Affective Features with a Hybrid Deep Model for Audio-Visual Emotion Recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 3030–3043, Oct. 2018, doi: 10.1109/TCSVT.2017.2719043.

[24] A. Jaratrotkamjorn and A. Choksuriwong, “Bimodal emotion recognition using deep belief network,” in *2019 23rd International Computer Science and Engineering Conference (ICSEC)*, IEEE, 2019, pp. 103–109.

[25] D. Issa, M. Fatih Demirci, and A. Yazici, “Speech emotion recognition with deep convolutional neural networks,” *Biomed Signal Process Control*, vol. 59, May 2020, doi: 10.1016/j.bspc.2020.101894.

[26] M. F. H. Siddiqui and A. Y. Javaid, “A multimodal facial emotion recognition framework through the fusion of speech with visible and infrared images,” *Multimodal Technologies and Interaction*, vol. 4, no. 3, pp. 1–21, Sep. 2020, doi: 10.3390/mti4030046.

[27] C. Luna-Jiménez, R. Kleinlein, D. Griol, Z. Callejas, J. M. Montero, and F. Fernández-Martínez, “A proposal for multimodal emotion recognition using aural transformers and action units on RAVDESS dataset,” *Applied Sciences*, vol. 12, no. 1, p. 327, 2021.

[28] C. Luna-Jiménez, D. Griol, Z. Callejas, R. Kleinlein, J. M. Montero, and F. Fernández-Martínez, “Multimodal emotion recognition on RAVDESS dataset using transfer learning,” *Sensors*, vol. 21, no. 22, Nov. 2021, doi: 10.3390/s21227665.

[29] T. Puri, M. Soni, G. Dhiman, O. Ibrahim Khalaf, M. alazzam, and I. Raza Khan, “Detection of Emotion of Speech for RAVDESS Audio Using Hybrid Convolution Neural Network,” *J Healthc Eng*, vol. 2022, 2022, doi: 10.1155/2022/8472947.

[30] A. I. Middya, B. Nag, and S. Roy, “Deep learning based multimodal emotion recognition using model-level fusion of audio–visual modalities,” *Knowl Based Syst*, vol. 244, May 2022, doi: 10.1016/j.knosys.2022.108580.

[31] K. Bhangale and M. Kothandaraman, “Speech Emotion Recognition Based on Multiple Acoustic Features and Deep Convolutional Neural Network,” *Electronics (Switzerland)*, vol. 12, no. 4, Feb. 2023, doi: 10.3390/electronics12040839.

[32] Y. A. Ibrahim, J. C. Odiketa, and T. S. Ibiyemi, “Preprocessing technique in automatic speech recognition for human computer interaction: an overview,” *Ann Comput Sci Ser*, vol. 15, no. 1, pp. 186–191, 2017.

[33] Witten, Frank, and Eibe, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Elsevier, 2005.

[34] M. El Ayadi, M. S. Kamel, and F. Karray, “Survey on speech emotion recognition: Features, classification schemes, and databases,” *Pattern Recognit*, vol. 44, no. 3, pp. 572–587, Mar. 2011, doi: 10.1016/j.patcog.2010.09.020.

[35] S. Bhadra, U. Sharma, and A. Choudhury, “Study on Feature Extraction of Speech Emotion Recognition,” *ADBU-Journal of Engineering Technology*, vol. 7, no. 1, p. 2016, doi: 10.7763/IPCSIT.2012.V49.27.

[36] Z. Yuxin and D. Yan, “A voice activity detection algorithm based on spectral entropy analysis of sub-frequency band,” *BioTechnol. Indian J*, vol. 10, pp. 12342–12348, 2014.

[37] K. Wang, N. An, B. N. Li, Y. Zhang, and L. Li, “Speech emotion recognition using Fourier parameters,” *IEEE Trans Affect Comput*, vol. 6, no. 1, pp. 69–75, Jan. 2015, doi: 10.1109/TAFFC.2015.2392101.

[38] C. H. Wu, J. C. Lin, and W. L. Wei, “Survey on audiovisual emotion recognition: Databases, features, and data fusion strategies,” *APSIPA Trans Signal Inf Process*, vol. 3, Nov. 2014, doi: 10.1017/ATSIP.2014.11.

[39] M. B. Er, “A Novel Approach for Classification of Speech Emotions Based on Deep and Acoustic Features,” *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.3043201.

[40] D. Bitouk, R. Verma, and A. Nenkova, “Class-level spectral features for emotion recognition,” *Speech Commun*, vol. 52, no. 7–8, pp. 613–625, 2010, doi: 10.1016/j.specom.2010.02.010.

[41] J. K. Das, A. Ghosh, A. K. Pal, S. Dutta, and A. Chakrabarty, “Urban Sound Classification Using Convolutional Neural Network and Long Short Term Memory Based on Multiple Features,” in *4th International Conference on Intelligent Computing in Data Sciences, ICDS 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020. doi: 10.1109/ICDS50568.2020.9268723.

[42] S. Paul, S. Bhattacharya, and S. Gupta, “Selection of keyframes for video colourization using steerable filtering,” *Sadhana - Academy Proceedings in Engineering Sciences*, vol. 42, no. 10, pp. 1685–1692, Oct. 2017, doi: 10.1007/s12046-017-0720-y.

[43] F. Noroozi, M. Marjanovic, A. Njegus, S. Escalera, and G. Anbarjafari, “Audio-Visual Emotion Recognition in Video Clips,” *IEEE Trans Affect Comput*, vol. 10, no. 1, pp. 60–75, Jan. 2019, doi: 10.1109/TAFFC.2017.2713783.

[44] I. H. Sarker, “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions,” *SN Computer Science*, vol. 2, no. 6. Springer, Nov. 01, 2021. doi: 10.1007/s42979-021-00815-1.

[45] A. Mathew, P. Amudha, and S. Sivakumari, “Deep learning techniques: an overview,” in *Advances in Intelligent Systems and Computing*, Springer, 2021, pp. 599–608. doi: 10.1007/978-981-15-3383-9_54.

[46] K. P. Murphy, *Machine learning: A probabilistic perspective*. MIT press, Cambridge, 2012.

[47] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognit*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/j.patcog.2017.10.013.

[48] W. Pedrycz and S.-M. Chen, *Deep learning: Concepts and architectures*. Springer, 2020.

[49] M. Sahu and R. Dash, “A survey on deep learning: Convolution neural network (cnn),” in *Smart Innovation, Systems and Technologies*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 317–325. doi: 10.1007/978-981-15-6202-0_32.

[50] Z. Şen, *Shallow and Deep Learning Principles*. Springer International Publishing, 2023. doi: 10.1007/978-3-031-29555-3.

[51] Y. Tian, “Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm,” *IEEE Access*, vol. 8, pp. 125731–125744, 2020, doi: 10.1109/ACCESS.2020.3006097.

[52] M. Gheisari *et al.*, “Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey,” *CAAI Transactions on Intelligence Technology*. John Wiley and Sons Inc, 2023. doi: 10.1049/cit2.12180.

[53] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights into Imaging*, vol. 9, no. 4. Springer Verlag, pp. 611–629, Aug. 01, 2018. doi: 10.1007/s13244-018-0639-9.

[54] J. Du, “Understanding of Object Detection Based on CNN Family and YOLO,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Apr. 2018. doi: 10.1088/1742-6596/1004/1/012029.

[55] H. Abdel-Jaber, D. Devassy, A. Al Salam, L. Hidaytallah, and M. El-Amir, “A Review of Deep Learning Algorithms and Their Applications in Healthcare,” *Algorithms*, vol. 15, no. 2. MDPI, Feb. 01, 2022. doi: 10.3390/a15020071.

[56] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[57] N. A. Al-Humaidan and M. Prince, “A Classification of Arab Ethnicity Based on Face Image Using Deep Learning Approach,” *IEEE Access*, vol. 9, pp. 50755–50766, 2021, doi: 10.1109/ACCESS.2021.3069022.

[58] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.04861

[59] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[60] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, pmlr, 2015, pp. 448–456.

[61] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Dec. 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[62] J. Jing, Z. Wang, M. Rätsch, and H. Zhang, “Mobile-Unet: An efficient convolutional neural network for fabric defect detection,” *Textile Research Journal*, vol. 92, no. 1–2, pp. 30–42, Jan. 2022, doi: 10.1177/0040517520928604.

[63] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*, PMLR, 2019, pp. 6105–6114.

[64] M. Tan *et al.*, “Mnasnet: Platform-aware neural architecture search for mobile,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2820–2828.

[65] M. Tan and Q. V Le, “Efficientnet: Improving accuracy and efficiency through automl and model scaling,” *arXiv preprint arXiv:1905.11946*, vol. 2, no. 5, 2019.

[66] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[67] A. Panigrahi, Y. Chen, and C.-C. J. Kuo, “Analysis on gradient propagation in batch normalized residual networks,” *arXiv preprint arXiv:1812.00342*, 2018.

[68] J. Lorraine and D. Duvenaud, “Stochastic hyperparameter optimization through hypernetworks,” *arXiv preprint arXiv:1802.09419*, 2018.

[69] K. You, M. Long, J. Wang, and M. I. Jordan, “How does learning rate decay help modern neural networks?,” *arXiv preprint arXiv:1908.01878*, 2019.

[70] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[71] M. Z. Alom *et al.*, “A state-of-the-art survey on deep learning theory and architectures,” *Electronics (Switzerland)*, vol. 8, no. 3. MDPI AG, Mar. 01, 2019. doi: 10.3390/electronics8030292.

[72] F. Zhuang *et al.*, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[73] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.

[74] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” *Adv Neural Inf Process Syst*, vol. 27, 2014.

[75] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*, Springer, 2018, pp. 270–279.

[76] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10. pp. 1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

[77] S. Wei, S. Zou, F. Liao, and W. Lang, “A Comparison on Data Augmentation Methods Based on Deep Learning for Audio Classification,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Mar. 2020. doi: 10.1088/1742-6596/1453/1/012085.

[78] A. Isenko, R. Mayer, J. Jedele, and H. A. Jacobsen, “Where Is My Training Bottleneck? Hidden Trade-Offs in Deep Learning Preprocessing Pipelines,” in *Proceedings of the ACM SIGMOD International Conference on*

Management of Data, Association for Computing Machinery, Jun. 2022, pp. 1825–1839. doi: 10.1145/3514221.3517848.

[79] M. Ghayoumi and A. K. Bansal, "Multimodal architecture for emotion in robots using deep learning," in *2016 Future Technologies Conference (FTC)*, IEEE, 2016, pp. 901–907.

[80] A. Rabie, "Audio-visual emotion recognition for natural human-robot interaction," *Ph.D. Dissertation, Bielefeld University*, 2010.

[81] F. Alonso-Martín, M. Malfaz, J. Sequeira, J. F. Gorostiza, and M. A. Salichs, "A multimodal emotion detection system during human-robot interaction," *Sensors (Switzerland)*, vol. 13, no. 11, pp. 15549–15581, Nov. 2013, doi: 10.3390/s131115549.

[82] S. R. Livingstone and F. A. Russo, "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, p. e0196391, 2018.

[83] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "Affectnet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Trans Affect Comput*, vol. 10, no. 1, pp. 18–31, 2017.

[84] Q. T. Ngo and S. Yoon, "Facial expression recognition based on weighted-cluster loss and deep transfer learning using a highly imbalanced dataset," *Sensors (Switzerland)*, vol. 20, no. 9, May 2020, doi: 10.3390/s20092639.

[85] Om B Khadka *et al.*, "Prediction Of Technical Feasibility For Building BridgesIn Nepal By Using Data Mining Technique," *international journal of engineering technology and management sciences*, vol. 7, no. 2, pp. 112–120, 2023, doi: 10.46647/ijetms.2023.v07i02.014.

[86] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

الخلاصة

تلعب العواطف دورًا رئيسيًا في العديد من جوانب حياتنا، وقد تؤثر أو حتى تحدد تفكيرنا وصنع القرار. يعد التعرف على المشاعر باستخدام البيانات متعددة الوسائط ، مثل الفيديو والصوت والنص وما إلى ذلك ، موضوعًا صعبًا ولكنه مجال بحث مهم استحوذ على الكثير من الاهتمام من الأكاديميين . في هذه الأطروحة تم تقديم نظام متعدد الوسائط للتعرف على المشاعر يعتمد على الطرائق المرئية والصوتية.

بالنسبة للمسار المرئي، يتم تحديد عدد الإطارات الرئيسية من كل فيديو بالتسلسل . حيث تم تدريب ثلاثة نماذج من الشبكات العصبية التلافيفية (CNN) المدربة مسبقًا من خلال تغيير مصنفاتها باستخدام مصنف مقترح جديد واختيار الأفضل . يقوم هذا النموذج باستخراج الميزات المناسبة من البيانات المرئية وتصنيفها وفقا لسبع فئات عاطفية مع حالتي تدريب باستخدام قاعدتي بيانات هما AffectNet وقاعدة بيانات (RAVDESS). اما في المسار الصوتي ، تم تحديد ميزات مختلفة من الأجزاء الصوتية لتصنيفها بواسطة متعدد الطبقات (Multi-layer Perceptron) المقترح إلى سبع فئات من المشاعر. تعتمد نتائج النظام المقترح على طريقة دمج مستوى القرار للحصول على نتيجة الإخراج لكل من الطرائق المرئية والصوتية. يتم استخدام طريقة Ensemble decision للحصول على التنبؤ النهائي من قيم مخرجات المتنبئين المختلفين.

تشير نتائج التجارب إلى أن الدقة في المسار البصري بلغت 98% لأفضل نموذج لقاعدة بيانات RAVDESS، في حين تم استخدام قاعدة بيانات AffectNet للتحقق من قابلية تعميم النموذج المقترح. وفي المسار الصوتي، حصل النموذج الأفضل على دقة قدرها 69% لقاعدة بيانات RAVDESS. تبلغ الدقة النهائية للنظام المقترح 99% بناءً على طريقة دمج قرارات المجموعة.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعـــة بـابــل

كلية تكنولوجيا المعلومات

قسم البرمجيات

تصنيف العواطف بالاعتماد على الانماط المرئية والمسموعه باستخدام شبكات التعلم العميق

اطروحة مقدمة إلى

مجلس كلية تكنولوجيا المعلومات / جامعة بابل كجزء من متطلبات

نيل درجة الدكتوراه فلسفة في تكنولوجيا المعلومات / برمجيات

من قبل

احمد صامت حاتم عبد الكريم

بإشـراف

أ.د. عباس محسن البكري

1445 هـ 2023 م