A Novel Deep Learning Approach for Emotion Classification

Satya Chandrashekhar Ayyalasomayajula

A thesis submitted in partial fulfillment of the requirements for the
Master of Applied Science
degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

"Learn how to see. Realize that everything connects to everything else."

Leonardo Da Vinci

Abstract

Neural Networks are at the core of computer vision solutions for various applications. With the advent of deep neural networks Facial Expression Recognition (FER) has been a very ineluctable and challenging task in the field of computer vision. Micro-expressions (ME) have been quite prominently used in security, psychotherapy, neuroscience and have a wide role in several related disciplines. However, due to the subtle movements of facial muscles, the micro-expressions are difficult to detect and identify. Due to the above, emotion detection and classification have always been hot research topics. The recently adopted networks to train FERs are yet to focus on issues caused due to overfitting, effectuated by insufficient data for training and expression unrelated variations like gender bias, face occlusions and others. Association of FER with the Speech Emotion Recognition (SER) triggered the development of multimodal neural networks for emotion classification in which the application of sensors played a significant role as they substantially increased the accuracy by providing high quality inputs, further elevating the efficiency of the system. This thesis relates to the exploration of different principles behind application of deep neural networks with a strong focus towards Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN) in regards to their applications to emotion recognition. A Motion Magnification algorithm for ME's detection and classification was implemented for applications requiring near real-time computations. A new and improved architecture using a Multimodal Network was implemented. In addition to the motion magnification technique for emotion classification and extraction, the Multimodal algorithm takes the audio-visual cues as inputs and reads the MEs on the real face of the participant. This feature of the above architecture can be deployed while administering interviews, or supervising ICU patients in hospitals, in the auto industry, and many others. The real-time emotion classifier based on state-of-the-art Image-Avatar Animation model was tested on simulated subjects. The salient features of the real-face are mapped on avatars that are build with a 3D scene generation platform. In pursuit of the goal of emotion classification, the Image Animation model outperforms all baselines and prior works. Extensive tests and results obtained demonstrate the validity of the approach.

Acknowledgements

This thesis was prepared at the Network Communications and Control Technologies (NCCT) Laboratory of the University of Ottawa. I am grateful for the openness and guidance of my supervisor, Dr. Dan Ionescu, who always encouraged me to tackle new challenges and never hesitated to offer help. His input and experience was invaluable throughout the realization of this work.

This thesis would not have been possible without the generous assistance of other members of the NCCT Lab. Their depth of knowledge and devotion to technical research created a positive working atmosphere that motivated me to learn about many new technologies. Bogdan Ionescu, Mircea Trifan, all contributed to the successful completion of this thesis.

Finally, I would like to thank my parents, my fiancee, my sister and my uncle for their continuous love, support and encouragement throughout this entire journey.

Publications

Content from this thesis has previously appeared in the following publication:

S.C.Ayyalasomayajula, B. Ionescu, D. Ionescu"A CNN Approach to Micro-Expressions Detection" *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), 2021, pp. 345-350.*

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

AAM	**A**ctive **A**ppearance **M**odel
A-EMM	**A**mplitude **E**ulerian **M**otion **M**agnification
ADAS	**A**dvanced **D**river **A**ssistance **S**ystems
ANN	**A**rtificial **N**eural **N**etwork
AU	**A**ction **U**nit
AWGN	**A**daptive **W**hite **G**aussian **N**oise
AUC	**A**rea **U**nder **C**urve
BEQ	**B**erkeley **E**xpressivity **Q**uestionnaire
Bi-WOOF	**Bi** **W**eighted **O**riented **O**ptical **F**low
BN	**B**atch **N**ormalization
BPM	**B**links **P**er **M**inute
CASME	**C**hineese **A**cademy of **S**ciences **M**icro **E**xpression
CEW	**C**losed **E**yes in the **W**ild
CNN	**C**onvolutional **N**eural **N**etwork
DTSCNN	**D**ual **T**emporal **S**cale **C**onvolutional **N**eural **N**etwork
EAR	**E**ye **A**spect **R**atio
EMM	**E**ulerian **M**otion **M**agnification
FACS	**F**ace **A**ction **C**oding **S**ystem
FER	**F**ace **E**xpression **R**ecognition
FERG-DB	**F**acial **E**xpression **R**esearch **G**roup **D**atabase
FLD	**F**ace **L**andmark **D**etection
FR	**F**ace **R**ecognition
GAN	**G**enerative **A**dversarial **N**etwork
GRN	**G**ate **R**ecurrent **U**nit

I2D	**I**ntrinsic **2D**imensional
LBP-TOP	**L**ocal **B**inary **P**attern **T**hree **O**rthogonal **P**lanes
LFPW	**L**abeled **F**ace **P**arts in the **W**ild
LLD	**L**ow **L**evel **D**escriptor
LReLU	**L**eaky **R**ectified **L**inear **U**nit
LSTM	**L**ong **S**hort **T**erm **M**emory
MAE	**M**ean **A**bsolute **E**rror
MAN	**M**ultimodal **A**ggregator **N**etwork
ME	**M**icro **E**xpression
METT	**M**icro **E**xpressions **T**raining **T**ool
MFCC	**M**el **F**requency **C**epstral **C**oefficients
MRI	**M**agnetic **R**esonane **I**maging
MRL	**M**edia **R**esearch **L**ab
NLM	**N**on **L**ocal **M**eans
RAF-DB	**R**eal-world **A**ffective **F**aces **D**atabase
RAVDSS	**R**yerson **A**udio **V**isual **D**atabase of **E**motional **S**peech and **S**ong
ReLU	**R**ectified **L**inear **U**nit
RNN	**R**ecurrent **N**eural **N**etwork
SER	**S**peech **E**motion **R**ecognition
SETT	**S**ubtle **E**xpressions **T**raining **T**ool
SMIC	**S**pontaneous **MIC**ro Expression Database
SMIC-NIR	**S**pontaneous **MIC**ro Expression Database - **N**early **I**nfered
SR	**S**peech **R**ecognition
SSD	**S**ingle **S**hot **D**etector
STTM	**S**patio **T**emporal **T**exture **M**ap
SVM	**S**upport **V**ector **M**achine
VLBP	**V**olume **L**ocal **B**inary **P**attern
WLD	**W**eber **L**ocal **D**escriptor
YorkDDT	**York D**eception **D**etection **T**est

Dedicated to my parents

Chapter 1

Introduction

1.1 Micro-Expressions and their Relationship to Emotions

Emotions play a vital role in our daily lives helping directly in revealing the feelings of an individual at a given time. Sigmund Freud theory states that a human being cannot have a neutral feeling/position in regards to a fact, idea, persons, etc. [1]. A person either likes or dislikes a fact, other person, situation,etc but cannot be neutral. Facial expressions are always showing the position of a person in regards to the situation or scene in which another person (or persons) is involved. A micro-expression is a subconscious flash of emotion across the face, lasting no longer than a quarter of a second. It is a generic reflection of what that person is feeling or thinking. However, it will quickly be adjusted, once the conscious mind kicks in. Facial features are defined by a combination of the movement and final position by 43 facial muscles giving rise to various expressions Fig. 1.2 [2]. These can be used to first identify the face expression and then predict the emotion at that state. These facial muscles are the primary source for communicating emotions but at the same time their movement makes the person trying to suppress them as well. The seven universal emotions [2] [3] namely anger, disgust, fear, happy, surprise, sadness and contempt are outlined in the Fig. 1.1.

Recent advents in using Artificial Intelligence developments in the area of Machine Learning (ML), made it possible to explore and automate the detection and interpretation of facts using special software like Artificial Neural Networks (ANN)

FIGURE 1.1: Example of seven universal emotions. From left to right: fear, contempt, sad, happy, surprise, anger, disgust

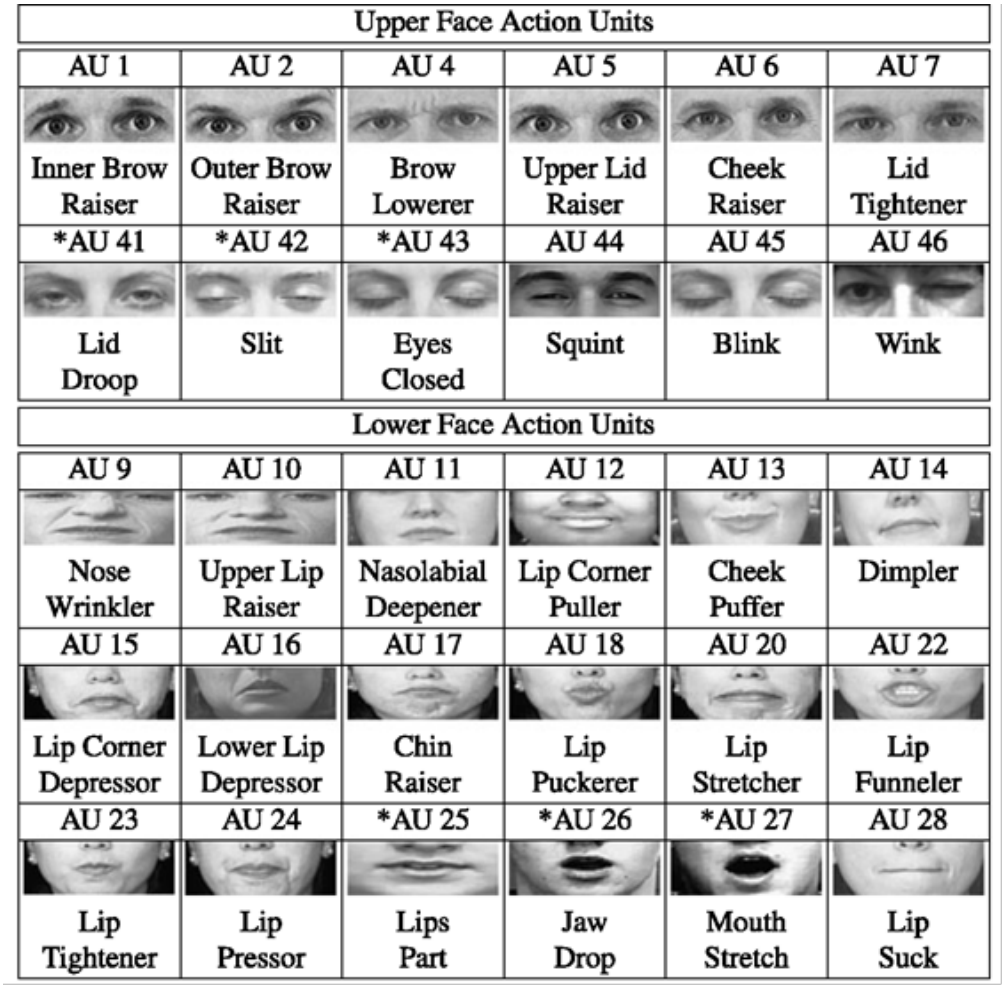

FIGURE 1.2: Upper and Lower Face Action Units

giving birth to the field called affective computing. This is presently a very hot research subject at the confluence of computer science, neuroscience, psychology and cognitive science with deep roots and applications in health. The affective computing methods and technologies can recognize, interpret and process human reactions to situation stimuli based on moods and emotions [4] [5] [6] [7]. A Micro-expression has a very short duration of 1/25–1/5 second and a light intensity making it quite difficult to perceive via the naked eye. Detecting emotions using facial

micro-expressions can be done as an application which uses the same technology like a "Face Recognition" application in which the detected Face Recognition template is further analyzed and classified, the results having different domain applications [8] [9] [10]. A taxonomy system of human facial movements named as Face Action Coding System [11] (FACS) which comprises of 44 Action Units (AU) is used to describe the different facial expressions. Two of the tools which were later devised and implemented to identify subtle emotions were Micro-Expressions Training Tool (METT) and Subtle Expression Training Tool (SETT). The seven universal emotions can be each represented by the following areas on the face: the left and right edges of the eyebrows, the left and right edges of the eyes, the left and right edges of the mouth and cheeks. Research on Emotion Detection and Recognition has been conducted for quite some time motivated by studying the reaction of humans to various stimuli and also for humans behavior during their interactions with a computer. The utilization of CNNs for speech and/or emotion detection and identification recognition is also gaining larger and larger attention as it has a double effect: it relates the Face Recognition to Natural Language Processing and improves the precision and the stability of the Face Recognition [12]. Yang et al. introduced a parallel model of a recurrent neural network (RNN) and a CNN to read the EEG signal in order to acquire meaningful features [13]. This creates a need to collect facial features of the person whose emotions are to be identified and associate it with the dynamics of the features matching one of the seven states of the emotions a human being can go through [2] [3].

Studies have shown that faces elicit activity in a specific region of the brain that includes the fusiform gyri which is associated with the face perception, amygdala which is associated with processing affect and the fronto-temporal regions associated with the knowledge of the individual [14] [15]. Researchers consider dynamic faces to be valid more than the static faces as when information from both the faces are available, people tend to use static faces for the face recognition whereas dynamic faces including motion, contributes to the quality of the structural information accessible that plays a unique role in social interaction, the evaluation of the mood etc [16]. Functional magnetic resonance imaging (fMRI) studies, investigate the face perception to typically reveal activation within the fusiform gyrus and occipital gyrus, areas part of the core regions of face processing, which mediate visual analysis of faces [17] [18].

The precision of detecting micro-expressions in a face data set depends very

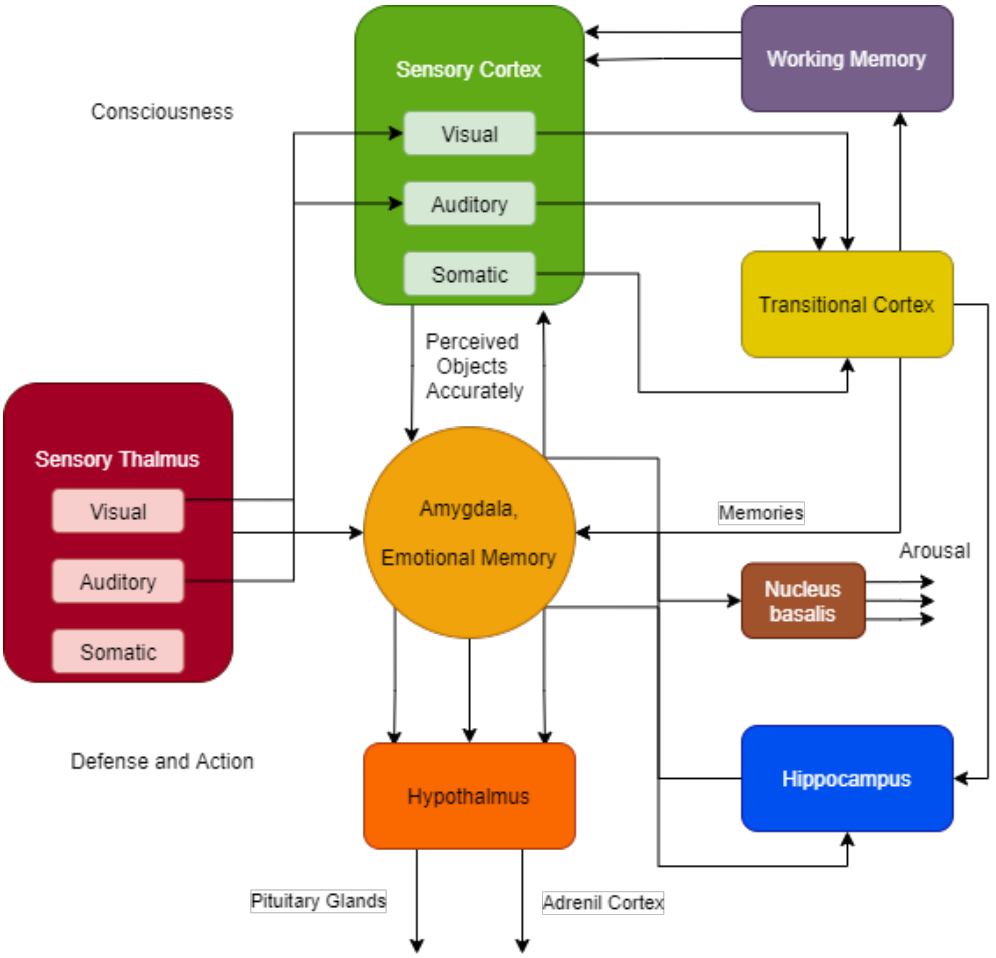

FIGURE 1.3: Emotional Pathway in the brain

much on the quality of the registered faces as these figures containing micro-expressions are used to train the CNN. CNN precision in determining the relevant micro-expressions requires training the model. However, this poses a significant challenge. Previously, micro-expressions detection was carried out using non-spontaneous datasets like the York Deception Detection Test (YorkDDT), a non spontaneous dataset which contained fewer data for ME's adopted by Warren et al. [19] in their study who had recorded a total of 20 video clips at 320x240 resolution and at a speed of 25 fps. Ongoing research in the field of micro-expression detection uses spontaneous datasets which provide comparatively better results than the former method as they are not posed which helps revealing the true emotion of the individual. These types of datasets are better for the micro-expression detection and interpretation as there is no posed data which makes it difficult to fake the expressions. CASME II [20], CAS(ME)^2 [21] , SAMM [22] are few such examples of the spontaneous datasets.

An issue while using the multimodal data processing is that the multisensory data is being processed separately and being combined at the end as people do display the different audio visual cues in a redundant manner. Chen et al. [23] devises an experiment to show human-like multimodal analysis of various input signals gathered by the sensors which are being processed in a joint featured space according to a context dependent model. A viable way to achieve the target tightly coupled multisensory data fusion is by developing context-dependent versions. A new method entitled Dual Temporal Scale Convolutional Neural Network (DTSCNN) was proposed by Peng et al. [24] consisting of only 4 convolutional layers which was fed with optical flow sequences achieving an accuracy of 66.67% on CASME II dataset. A Deep Belief Network developed by Hao and Tian [25] in the second stage to extract more features was associated with a Weber Local Descriptor (WLD) but the model was only evaluated on a non-spontaneous dataset making it difficult to adopt with the current literature. Training a ME based emotion classifier using CNN requires large amount of data and ME's being complex are difficult to categorize in different classes [26].

TABLE 1.1: Combination of Action units for the seven universal emotions

Emotion	Action Units
Happy	6+12
Sad	1+4+15
Surprise	1+2+5B+26
Fear	1+2+4+5+7+20+26
Anger	4+5+7+23
Disgust	9+15+17
Contempt	R12A+R14A

1.2 Thesis Objective

Enabling computers to understand human's affective states such as interests and emotions, requires building applications capable of encoding human feelings by blending computers and humans in a real-time feedback loop which operates on acoustic, visual and linguistics signals. The objective of this thesis is to devise a state of the art emotion classifier using facial micro-expressions in real time accompanied by a low computation model performing with a high accuracy rate. The problem of obtaining an emotion classifier with a high accuracy in inference

stage was solved in this thesis using different algorithms such as eulerian motion magnification, multimodal networks and an image to avatar animation model.

1.3 Thesis Contributions

This thesis brings the following contributions of this field:

1. (a) Design and Implementation of a new CNN architecture for the face detection using Single Shot Multi-Box Detector which surpasses the previous face detectors including Haar-Cascades, Dlib Frontal Face detector and OpenCV Deep Neural Network in terms of both speed and accuracy. The face detector designed achieved a high inference accuracy of 95.2%.

   (b) A novel CNN was designed and implemented for the facial landmarks detection and localization detecting the 68 points facial mesh in real time with an accuracy of 98.6%. The CNN implemented was suitable and performed remarkably even in noisy environments with face occlusions.

   (c) Devising a novel method of detecting micro-expressions and emotion classification using Eulerian Video Magnification (EVM) and testing the inference speed and accuracy against the formerly used algorithms for emotion classification.

2. (a) Applying the emotion classification on real-time interviews by taking the audio-visual cues from the interviewee. This approach uses a combination of convolutional neural networks related to FER and SER and a multimodal aggregator network to achieve a stable emotion classifier.

   (b) Implementing an image pre-processing algorithm to denoise the images present in the training datasets on which the CNN is used to train the emotion classifier.

   (c) Designing an algorithm to detect and classify facial micro-expressions based on real time image animation.

1.4 Thesis Outline

The outline of this thesis is as follows:

Chapter 1 is the introduction part, the problem statement accompanied by the research pursued in this thesis and a brief introduction to the different methodologies and solutions implemented over the years to overcome the problem of detecting emotions.

Chapter 2 consists of literature review, encompassing the theories and recent work done in the field of deep learning. Different concepts concerning neural networks, convolutional neural networks, transfer learning backbone networks, an introduction to generative adversarial networks and its components, the datasets required to train the models have been explained in detail in this chapter.

Chapter 3 depicts the architecture flow of the various emotion classifiers implemented in the thesis which are revisited in detail in the following chapters.

Chapter 4 details an implementation of a motion magnification model for emotion detection. The different stages composed in the structure of the experiment are explained along with the algorithm used.

Chapter 5 comprises the experiment involving a multimodal based approach for detecting emotion using audio-visual cues. The architecture and the algorithm of the implemented model is discussed in this chapter.

Chapter 6 details an experiment on a denoised emotion classifier and a comparison of the performance metrics of this new algorithm to train the emotion classifier models to the models discussed in the previous sections. Eventually, a real time image animation based emotion classifier was designed which surpassed the accuracy of all the previously developed models discussed in Chapter 4 and Chapter 5 .

Chapter 7 provides the results obtained after executing the simulations for the designed experiments from Chapter 4, 5 and 6. The different performance metrics associated with the algorithms are discussed in this chapter.

Finally **Chapter 8** is the conclusion, restating the main conclusions and contributions of this work. Future directions for the architecture and novel uses of a real-time emotion classifier are also described.

Chapter 2

Background and Related Work

In this chapter, the state of the art in regards to concepts needed to properly understand, automatically detect, and then identify the human emotions are described, In this context, concepts related to neural networks (NN) with a strong focus towards Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN)s are introduced and discussed. There is a very sustained research activity in this area and valuable surveys were recently published in regards to CNNs in general, like [27] [28] [29] [30] [31]. The motivation stems from the fact that among many other aspects covered by the CNN and Machine Learning (ML) domains the most commonly used transfer learning network architectures are used as a backbone for training purposes. Thus the focus of this Chapter aims at the analysis of the components of CNNs with emphasis of networks used for Transfer Learning and on Generative Adversarial Networks (GAN)s.

2.1 Neural Network

Neural Networks (NN) or Artificial Neural Networks (ANN) consists of ordered sets of node layers, each containing an input layer, one or more hidden layers, and an output layer. Every node, or the neuron is connected to another neuron and has a specific weight and threshold associated with it. To activate the neuron, it must reach the threshold value which further sends the data to the next layer of the network. If the neuron is not activated, no data is being passed along to the next layer of the network. One of the first neural networks can be traced back to

the year 1943 which was developed by Pitts and McCulloch [32] [33]. This Neural Network was also known as the Multi Layer Perceptron (MLP).

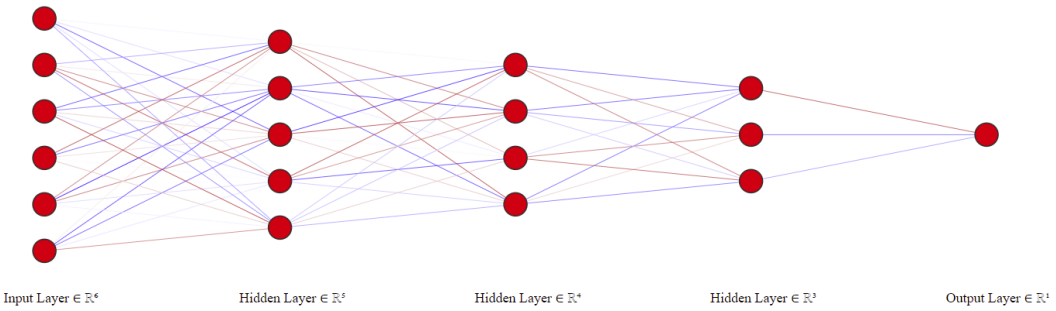

FIGURE 2.1: Example of a Neural Network Architecture

Neural networks are heavily relying on the training data to learn and improve their validation accuracy over time and, once these networks are finely tuned for deployment they become very powerful tools to be used in the fields of deep-learning and, artificial intelligence.

The following equation 2.1 shows how every individual neuron can be classified as a self-regression model which has different parameters such as X a vector with m components x_i symbolizing the input data which can be pixels in an image or phonemes in a recorded or real-time speech or any real data being numbers or elements from a list, etc,. The weight of the variable of the linear regression relationship, b being an additional node, called the bias, $f(x)$ being the activation function or the threshold function through which the weighted sum is passed resulting in output y.

$$y(x, w) = \sum_{i=1}^{m} w_i x_i + b = w_1 x_1 + w_2 x_2 + ... + b \tag{2.1}$$

$$f(x) = \begin{cases} 1, & \text{if } \sum w_1 x_1 + b \geq 0 \\ 0, & \text{if } \sum w_1 x_1 + b < 0 \end{cases}$$

The weights are assigned once the input layer is determined. This process of passing data from one layer to the next layer is called a feed-forward network. The evaluation of the model trained for a specific purpose like image classification, object detection etc. is done by using a cost function.

2.2 Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of deep learning model which utilizes a grid pattern to solve a variety of problems, mainly comprised of images. These CNN's are designed to learn from a set of spatial hierarchies of features [34] [35]. The building blocks of a CNN can be classified into 3 layers namely a i) convolutional layer, a ii) pooling layer which performs tasks such as feature extraction and a iii) fully connected layer which is mostly used for classfication purposes [36].

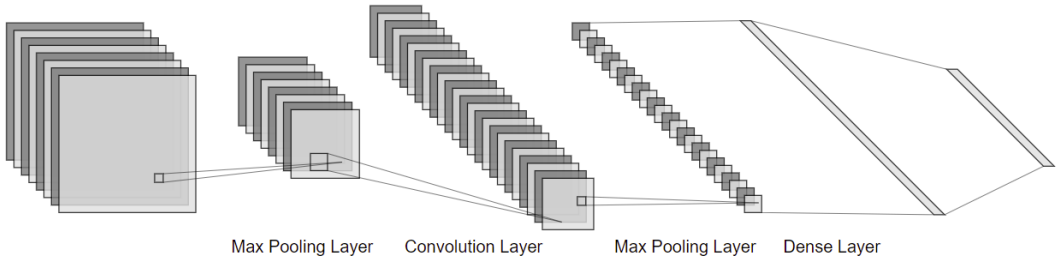

FIGURE 2.2: Example of a Simple Convolutional Neural Network

2.2.1 Convolutional Layer

A 2D dimensional brightness array has its intensity values in range from 0-255. Hence, for a multicolor image three separate intensity matrices are required which are the three RGB channels. Mathematically, a discrete 2D convolution can be described as follows where $K \in R^{(2h_1+1)*(2h_2+1)}$, the convolution of image I with a filter of K

$$conv(I * K)_{u,v} = \sum_{u=-h_1}^{h_1} \sum_{v=-h_2}^{h_2} K_{u,v} I_{r-u,s-v} \tag{2.2}$$

$$K = \begin{bmatrix} K_{-h_1,-h_2} & \dots & K_{-h_1,h_2} \\ . & K_{0,0} & . \\ K_{h_1,-h_2} & \dots & K_{h_1,h_2} \end{bmatrix} \tag{2.3}$$

The main principle in CNN operations is to match a filter of a given size with a specific patch of the image whose resultant feature map provides the information on how good the local filter fits the patch. The task of the convolution layer is to detect the local concurrences of the features from the previous layer and further mapping their information to a feature map. The image is split into perceptrons, which in turn create local receptive fields by compressing these images in the maps of dimensions $m_2 * m_3$. In each layer, there are m_1 filters and the number of filters applied in a stage can be deduced by the total depth of the number of feature maps. The output Y_i^l with a layer l has a total of m_1^l feature maps of sizes $m_2^l * m_3^l$ where the i^{th} feature map can be computed as [37]

$$Y_i^l = B_i^l + \sum_{j=1}^{m_i^{(l-1)}} K_{ij}^l * Y_j^{(l-1)} \tag{2.4}$$

where B_i^l is defined as the bias matrix.

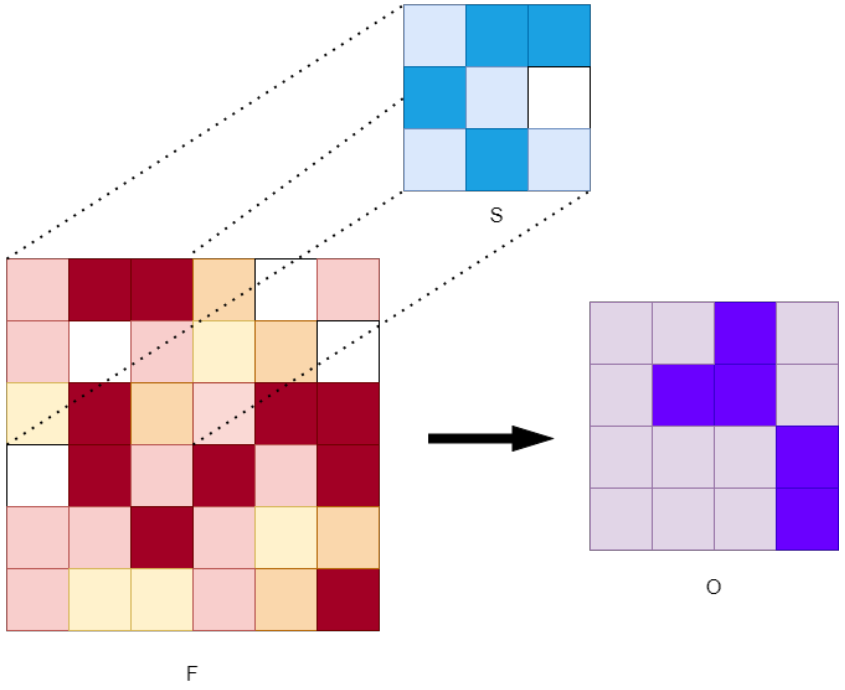

FIGURE 2.3: Convolution done on a Filter (F) using a 3x3 Stride (S) resulting in a feature map output O

2.2.2 Backpropagation

Backpropagation is a technique which is used to adjust the weights of the neural networks. In case of a model having convolutional neural networks, the major

optimization parameters are the kernel parameters, pooling layer weights, dense layer or fully connected layer weights and the bias parameters. In backpropagation [38], as the name suggests the calculation of the gradient is performed backwards through the network. i.e. the gradient of the last layer's weights calculated first and the gradient of the weights of the first layer being computed at last. The partial derivatives of the weights for each layer are learnt while adjusting the final weights of the network output to achieve a better convergence to the expected value which are stored in a gradient vector. [39].

$$W_i = W_i - \eta \frac{\partial E(W, b)}{\partial W_i} \tag{2.5}$$

$$b_i = b_i - \eta \frac{\partial E(W, b)}{\partial b_i} \tag{2.6}$$

where η is the learning rate, W_i being the weight updated, and b_i the bias parameter and E the error function.

2.2.3 Activation Functions

In the process of training a CNN each of the neural network Weights receives an update proportional to $\frac{\partial E(W,b)}{\partial b_i}$. In some cases the gradient is so small that it prevents the weight from changing its value. This is known as the vanishing gradient problem. Also, neurons can sometimes be pushed into states in which they become inactive for essentially all inputs. This is called the dead neuron problem. This happens when the learning rate is set too high. The vanishing gradient problem and the dead neuron problem are the main consequences from the mostly adopted activation functions. In a typical vanishing gradient problem a deep layer feed forward network or any neural network is unable to pass on the required gradient information during the backpropogation process which basically is the result of the inability of the models to learn on a specific dataset or to prematurely converge to a poor solution [40].

2.2.3.1 Sigmoid Function

The sigmoid functions are one of the most commonly used activation functions which are defined as

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.7}$$

where x is the input to the activation function. This function is a continuous function and is bounded in the range of (0,1) and is also diffrentiable. Fig 2.4 shows the graph and the gradient of the sigmoid function [41].

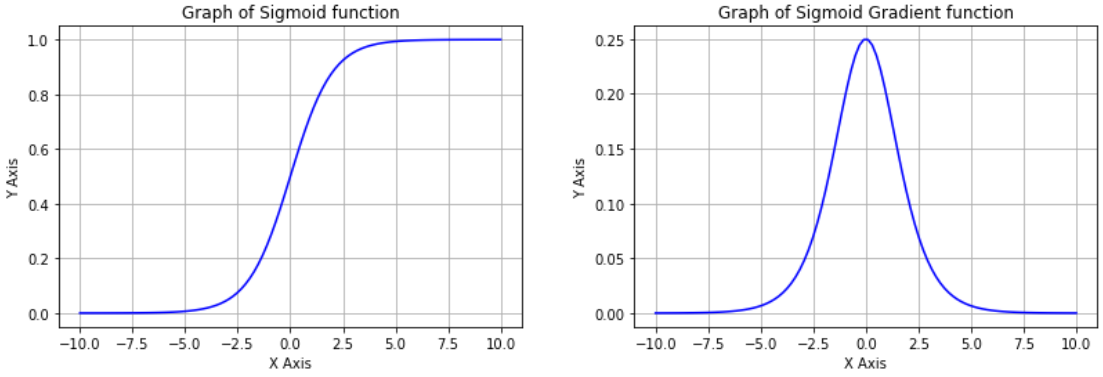

FIGURE 2.4: Graph of sigmoid function (a), Graph of the gradient of the sigmoid function (b).

Being bounded in the range (0,1), this function always has a non-negative output, i.e. a large change to the input value leads to only a slight change in the output resulting in small gradients which further results in the vanishing gradient problem.

2.2.3.2 Tanh Function

The tanh function or the hyperbolic tangent function gives a better performance in training multi-layer neural nets which can be defined as

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{2.8}$$

where x is the input to the activation function. The hyperbolic tangent function can also be expressed by the following equation

$$tanh(x) = 2sigmoid(2x) - 1 \tag{2.9}$$

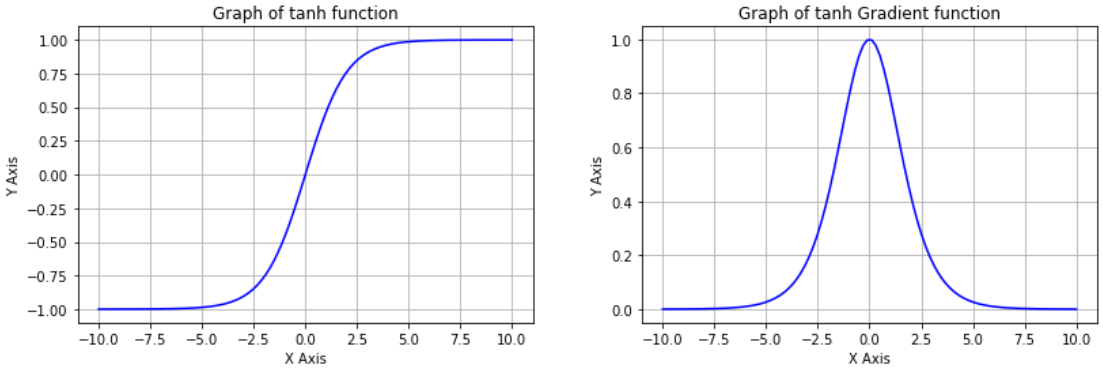

FIGURE 2.5: Graph of tanh function (a), Graph of the gradient of the tanh function (b).

It is also a differentiable function but has its boundary limits as (-1,1) which makes it versatile as it can produce an output which can be positive, negative or zero [42]. Fig 2.5 shows the graph and the gradient of the tanh function. As it can provide a zero-centered output unlike the sigmoid function it helps during the backpropogation process but resembles sigmoid i.e, the vanishing gradient is also encountered by using tanh as an activation function because of which ReLU was introduced.

2.2.3.3 ReLU Functions

The ReLU or the Rectified Linear Unit activation function was first introduced by Nair et al. [43]. This function is defined as

$$f(x) = max(0, x) \tag{2.10}$$

where x is the input to the activation function. The ReLU function is continuous, not bounded function and is also not zero-centered. This function is only not diffrentiable at $x = 0$. ReLU is computationally cheap unlike the hyperbolic tangent as it forces the negative values to zero because of which there is no vanishing gradient problem occurrence in this activation function.

Fig 2.6 shows the graph and the gradient of the ReLU function. Any input to this function which is a negative number gives the output as 0. The model inputs which have negative weights fail to contribute to the training process leading to a dead neurons. To overcome the dead neurons a new variant of ReLU was devised known as the Leaky ReLU.

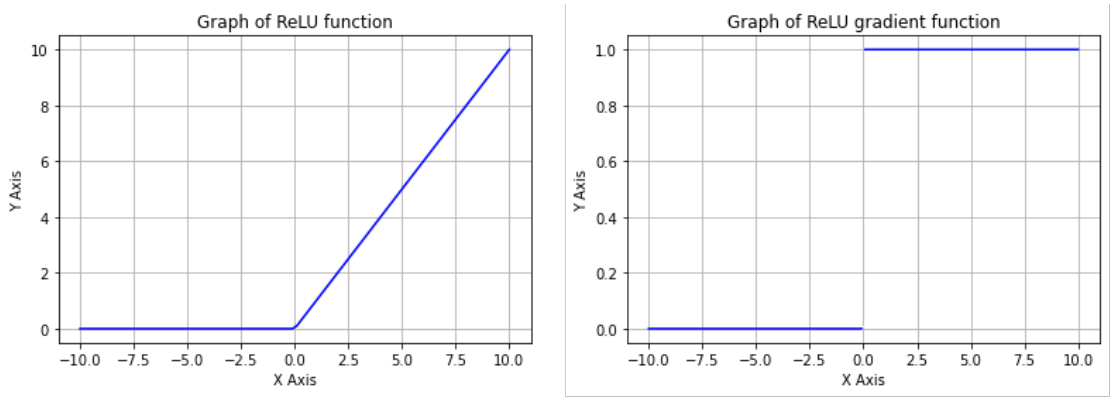

FIGURE 2.6: Graph of ReLU function (a), Graph of the gradient of the ReLU function (b)

2.2.3.4 Leaky ReLU Function

The leaky ReLU or the LReLU function [44] was introduced to overcome the dead neuron problem from its predecessor. The LReLU activation function is defined as

$$f(x) = \begin{cases} 0.01x, & \text{for x} \leq 0 \\ x, & \text{elsewhere} \end{cases}$$

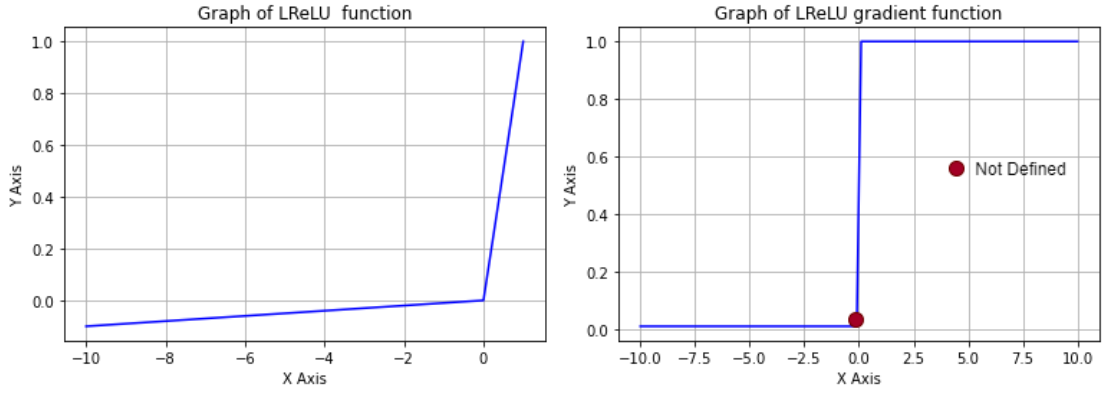

FIGURE 2.7: Graph of LReLU function (a), Graph of the gradient of the LReLU function (b)

where x is the input to the activation function. Fig 2.7 shows the graph and the gradient of the LReLU function. The LReLU like ReLU is continuous and not a bounded function. It has also low computational cost and is a zero centered activation function. Instead of subjugating all the negative values to zero the LReLU functions allows a small part of the negative units. In the positive part of the function, there is no scope of the vanishing gradient problem as the derivative

is always 1 but at the negative side the gradient is always 0.01 which is close to zero and can cause a vanishing gradient.

2.2.4 Pooling Layer

To reduce the spacial size on activation maps there is a requirement of a downsampling layer i.e, the pooling layer. Pooling layers are often used after performing multiple stages of convolutions in order to reduce the computational costs of the entire neural network also simultaneously minimizing the risk of overfitting the model [37].

$$m_1^l = m_1^{l-1} \tag{2.11}$$

$$m_2^l = \frac{(m_2^{l-1} - F^l)}{S^l + 1} \tag{2.12}$$

$$m_3^l = \frac{(m_3^{l-1} - F^l)}{S^l + 1} \tag{2.13}$$

where F are the Filter and S is the Stride hyperparameters and the output is of size $m_1^l * m_2^l * m_3^l$ by taking an input of $m_1^{l-1} * m_2^{l-1} * m_3^{l-1}$. It is not necessarily required that the window of the pooling filter must be a square, although its highly unlikely to use a rectangular window while doing the pooling operation. The pooling operation is performed by defining a window of dimension $F^l * F^l$ which reduces the data to a single value. This window is then advanced by a Stride S^l after every operation until the entire volume is reduced spatially. No learning is being taken place when the pooling operation is being performed. Max pooling and Average pooling are the two most common reduction methods used for this operation. Max pooling finds the highest value within the window and discards the rest values whereas average pooling uses the mean of the values within the window. Fig. 2.8 and Fig. 2.9 show the operation of average pooling and max pooling respectively.

2.2.5 Batch Normalization

Batch Normalization is the process where the mean and variance generated from the output activations from a CNN layer follows a unit gaussian distribution [45]. Batch Normalization reduces the internal covariance shift of the layer activations.

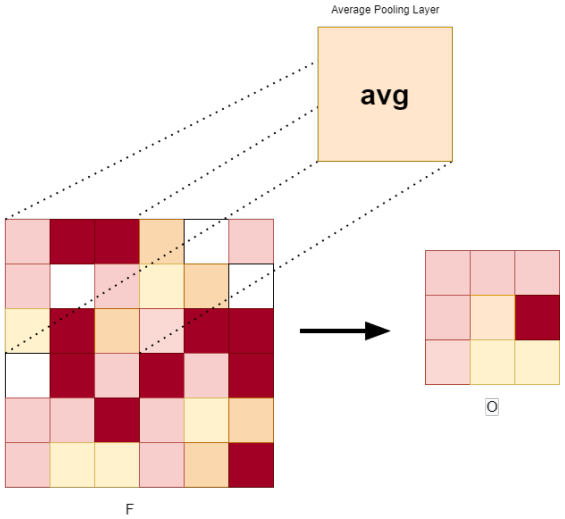

FIGURE 2.8: Average Pooling done on a Filter (F) resulting in a feature map output O

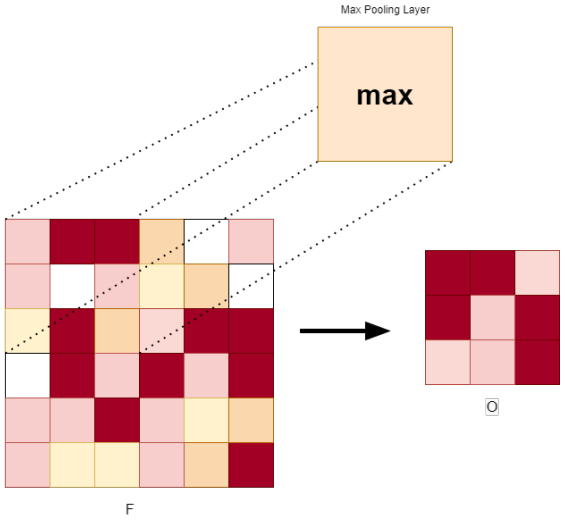

FIGURE 2.9: Max Pooling done on a Filter (F) resulting in a feature map output O

Internal covariance shift depicts the change in the activation distribution of that particular layer whose parameters are being updated steadily while in training. Higher internal covariance shift will increase the training time as it takes a long time to converge. Using normalization of this distribution gives a consistent activation distribution which can enhance network performance, improve convergence and can also help to avoid some major issues such as activation saturation and vanishing gradients. Assuming a total set of activations [45]

$$x^i : i \in [1, m] \tag{2.14}$$

$$x^i = \{x_j^i : j \in [1, n]\} \tag{2.15}$$

where the convolutional neural network has the input batch with m images, the first and the second order mean and variance of the batch for each dimension of the activation functions can be computed as follows

$$\mu_{x_j} = \frac{1}{m} \sum_{i=1}^{m} x_j^i \tag{2.16}$$

$$\sigma_{x_j}^2 = \frac{1}{m} \sum_{i=1}^{m} (x_j^i - \mu_{x_j})^2 \tag{2.17}$$

where μ_{x_j} and $\sigma^2 x_j$ are the mean and variance respectively. The normalized activation operation is shows as

$$\hat{x_j^i} = \frac{x_j^i - \mu_{x_j}}{\sqrt{\sigma_{x_j}^2 + \epsilon}} \tag{2.18}$$

Despite of performing the normalization operation, it can sometimes unlearn the patterns which were previously learned by the network. To overcome this, the normalized activations are further rescaled and shifted so that they are able to learn useful discriminative representations.

$$y_j^i = \gamma_j \hat{x_j^i} + \beta_j \tag{2.19}$$

The batch normalization algorithm is applied after the convolutional neural network weight layers and before applying the activation function.

2.3 Loss Functions

2.3.1 Cost Functions used for Regression

The ultimate goal of training any neural network is to minimize the cost function to ensure the correctness of fit for a given specific observation. After every epoch, the algorithm adjusts its weights through gradient descent which allows the model

to determine the direction to take i.e, minimizing the cost function to gradually converge at the local minimum.

2.3.1.1 Mean Absolute Error

This cost function i.e, MAE is one of the widely adapted cost functions in regression which measures the average magnitude of errors in a set of predictions, irrespective of their direction. All the individual deviations encountered have equal importance [46].

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i| \tag{2.20}$$

where i is the index of the sample $\hat{y}$ is the predicted value, y is the expected value , m is the total number of samples in the dataset.

2.3.1.2 Mean Squared Error and Root Mean Squared Error

In this cost function the average squared difference between the predictions and expected results are taken into account. It is a modification from the MAE cost function where in the former the absolute value of difference was considered and here they are squared. In MSE the partial error is equivalent to the area of the square created out of the geometrical distance between the measured points. All region areas are summed up and averaged. Root mean square error can be treated as an extension of the MSE where the average of the square root of sum of the squared differences between predictions and actual observations are considered [47].

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{2.21}$$

$$RMSE = \frac{1}{\sqrt{m}}\sqrt{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{2.22}$$

where i is the index of the sample $\hat{y}$ is the predicted value, y is the expected value ,m is the total number of samples in the dataset.

2.3.2 Classification Loss Functions

The final classification is the output from the last layer of the CNN model architecture. The loss functions which are implemented at the output layer are defined to calculate the predicted error for the training samples in the neural network model. This error is the difference between the actual output and the predicted output which is optimized during learning. There are majorly two classes of loss functions in classification problems i.e, binary classification and multi-class classification [48].

2.3.2.1 Binary Classification Loss

The Support Vector Machine (SVM) hinge loss and the squared hinge loss are some commonly used binary classification loss functions. The SVM hinge loss is an error function used while training an SVM classifier. This loss maximizes the margin between the true and negative class samples which can be computed using the following expression

$$L(p, y) = \sum_{n} max(0, m - (2y_n - 1)p_n) \tag{2.23}$$

where m is defined as the margin which usually is set to 1 and p being the predicted output and y being the desired output [49]. The squared hinge loss has the square of the max function as shown [48]

$$L(p, y) = \sum_{n} max^2(0, m - (2y_n - 1)p_n) \tag{2.24}$$

This loss shown in eq. 2.24 is more sensitive to margin violations than its predecessor eq. 2.23.

2.3.2.2 Multi-Class Classification Loss

In multi-class classification problems, two of the losses which are most popular are the expectation loss and the cross-entropy loss. The cross entropy loss most commonly known as the "softmax loss" is defined as [48]

$$L(p, y) = \sum_{n} y_n \log(p_n) \tag{2.25}$$

where y is the output and p being the probability for each output class. The probability of each class present can be computed by

$$p_n = \frac{\exp \hat{p_n}}{\sum_{k} \exp \hat{p_k}} \tag{2.26}$$

The expectation loss is defined as follows

$$L(p, y) = \sum_{n} |y_n - \frac{\exp p_n}{\sum_{k} \exp p_k}| \tag{2.27}$$

where $n \in [1, N]$. This loss is said to minimize the expected misclassification probability hence the nomenclature. Similar to the cross-entropy loss expectation loss also uses the softmax function however it directly maximizes the probability of fully correct predictions [50]. Even though the expectation loss provides more robustness, this function is rarely used in the CNN due it not being a convex or concave function which can lead to optimization issues during the training process.

2.4 Recent Studies in Emotion Detection and Identification

Guo et al. [51] proposed Local Binary Pattern Three Orthogonal Planes (LBP-TOP) features using a nearest neighbour classifier to compare the euclidean distance between the known and the unknown samples for the micro-expression detection experiment. This method achieved a best accuracy of 63% on the Spontaneous Micro-expression Database (SMIC). Increase in the facial expression activity further increased the error in this method which invoked a larger computation time to process the emotion classification making it unsuitable for real time or near real time applications.

Pelachaud et al. [52] had devised an experiment which generated animated facial expressions for different synthetic speeches but this work only focused on synthetic speeches and not on the emotions. De Silva et al. [53] proposed a rule-based method for the singular classification of audio-visual input data into the six emotion categories. Each of their subjects were asked purposely to portray different emotions by displaying the related facial muscular movement and simultaneously were asked to speak a single English work of choice which were processed separately and the rules for emotion classification were defined.

Kim et al. [54] proposed a new feature representation for the micro-expressions in which all temporal states were encoded by a CNN which is passed to the long short term memory (LSTM). A recurrent neural network (RNN) where the temporal characteristics are being analyzed [55] was inroduced. The authors evaluated their model on Chinese Academy of Sciences Micro-Expression (CASME) II dataset and achieved an accuracy rate of 60.98%. Talukdar et al. [56] used LBP-TOP to extract the salient features and then magnified motions of various micro-expression have been passed through a Support Vector Machines (SVM) classifier. The accuracy of their proposed model was 62% on the set of faces considered (SMIC-NIR) [57]. Chen and Huang [58] proposed a set of different methods for singular classification of input audio visual data into one of the six emotion categories [27]: happiness, sadness, disgust, fear, anger, and surprise. They collected different data from a total of 5 subjects who displayed the six basic emotions by producing the facial expression right before or after an appropriate vocal emotion applying a single-modal classification method in a sequential manner.

During the identification of facial features, an image noise generated by the face feature instability caused by the presence of the emotion (face muscle movements) makes the face detection imprecise. Thus, there is a need to use different data such as speech or text to complement the face pictures, by connecting the CNN cells with feedback connections [59]. Sometimes, humans while interacting, show different kind of gestures associated with speech to express a certain emotion [28]. As a consequence, traditional methods become obsolete when predicting emotions, thus there is a need for the introduction of all the data, such as the face image, the text, the voice, and other data in the neuron topology of connections.

A Delaunay-Based Temporal Coding Model (DTCM) for ME recognition was proposed by Lu et al. [60] which used the Active Appearance Model (AAM) to define the 68 facial points on which the delaunay triangulation was implemented. The

triangulation divides the facial region into segments with a triangular shape. The normalisation was performed on the neutral face removing the personal appearance difference and local temporal variations were used to code the feature space. The use of delaunay triangulation generates very vast number of sub-regions further leading to a large number of local features which was avoided by segregating the subregions in which ME's were present based on a standard deviation analysis. They achieved better result than state of the art, with 82.86% and 64.19% SMIC and CASME II respectively. Kamarol et al. [61] proposed Spatio-Temporal Texture Map (STTM) to extract the ME features. The method was evaluated on CASME II and was further compared with Volume Local Binary Pattern (VLBP). The average accuracy achieved by STTM was 91.71% which was a great improvement but the computational time soared to 2.57 seconds making it difficult to perform in real-time.

Liong et al. [62] [63] instead of using all frames of ME's for analysis used only the apex and the onset frame which were extracted using a Bi-Weighted Oriented Optical Flow (Bi-WOOF). Their theory was evaluated on CASME II and achieved an F1-score (harmonic mean of the precision and recall) of 0.61 which depicts a low precision. An enhancement of their proposed theory was implemented, in which their team proposed features known as optical strain weights which were computed using the facial optical strain magnitudes allowing them to reach an accuracy rate of 63.16% on CASME II. A similar kind of approach was pursued by Oh et al. [64] in which their theory states that there are changes on the facial contours in the process of ME recognition. A feature extraction method was developed called Intrinsic Two-Dimensional local structures (I2D). The proposed feature extraction method was evaluated on the CASME II and SMIC with an F1 score of 0.44 and 0.41 respectively.

A hot wheel patterns from three orthogonal planes (HWP-TOP) was devised by Ben et al. [65] where they use smooth SVM as the classifier which achieved a decent accuracy rate on CASME II dataset omitting 2 classes (fear and sadness). Distance estimation between points predicted using Random Walk-based (RW) [66] for learning different features was performed after the feature extraction stage. The usage of RW in the model reduces the dimensionality of the features and further minimizes the computation complexity. The authors chose Area Under Curve (AUC) as their performance metric in which the model gave a score of 0.8812 on SMIC and 0.9456 on CASME dataset.

Another deep learning technique that has recently achieved success within the SER systems is the attention mechanism [67] [68] [69] [70]. Incorporating traditional methods for SER the method assigns all locations of a given utterence to get equal attention; however, as emotion is not evenly distributed over the utterence of every sample the classification results have poor accuracy. In the attention mechanism, the classifier treats the sample's specific location based on a set of attention weights which are already pre-assigned to the data containing an emotionally salient portion. Mirsamadi et al. [71] used attention mechanism instead of low-level descriptors (LLD) and LSTMs to focus on emotionally salient parts of a sentence and ignore the silent frames. This method was tested on the Interactive Emotional Dyadic Motion Capture (IEMOCAP) dataset. Li et al. [69] utilizing a combination of multitask learning and attention mechanism. The authors propsed a new architecture named CNN-BLSTM whose inputs were a speech spectogram. This new architecture was 7.7% more efficient than the multi-channel CNN on the IEMOCAP dataset. Section 2.5 shows some of the most frequently used CNNs for emotion detection and classification. The first GANs introduced by Goodfellow and team [72] in 2014 had motivation from noise-contrastive estimation [73]. The first GAN to be used for an image processing application such as image enhancement to produce a higher image quality was developed in 2017 along with the first deep-fakes of the faces. Adoption of these frameworks in the ME's and emotion classification to increase the accuracy of the classifier proceeded but had a little success. Nowadays, GANs are prominently used for applications in data augmentation, data representation, and denoising while performing FER or SER. Latif et al. [74] used GANs for the robustness of the SER model showing how the samples generated by adding indistinguishable noise to the original samples can affect the SER systems.

2.5 Commonly Adapted Backbone Networks for Transfer Learning

Transfer learning is the process in which the knowledge of a previously trained model is applied to a new but related problem. Transfer learning can not only save huge time while training but having a better initial model can improve the learning rate, accuracy and also the convergence time. Some of the backbone

networks used in this thesis to train the convolutional neural network models are discussed in this section.

2.5.1 VGG-16

VGG-16 has the input image of the dimensions (224,224,3). The first two layers consist of 64 channels having a filter size F of (3x3). The next layer is a max pooling layer having a stride S of (2x2), followed by a set of convolution layers and pooling layers. A total of 3 fully connected layers are used with outputs of 4096, 4096 and 1000 respectively. The hidden layers in this architecture use ReLU as their activation function because of its lower computational cost which also results in faster learning. Similar to the AlexNet network architecture, VGG16 also uses activation dropouts in the first two dense layers which reduce the overfitting of the model.Table 2.1 shows the VGG-16 architecture [75].

TABLE 2.1: Architecture of VGG-16

Layer	**Dimensions**	**Kernel Size**	**Stride**	**Activation**
Convolution	224x224x64	3x3	1	relu
Convolution	224x224x64	3x3	1	relu
Maxpool	112x112x64	2x2	2	relu
Convolution	112x112x128	3x3	1	relu
Convolution	112x112x128	3x3	1	relu
Maxpool	56x56x128	2x2	2	relu
Convolution	56x56x256	3x3	1	relu
Convolution	56x56x256	3x3	1	relu
Convolution	56x56x256	3x3	1	relu
Maxpool	28x28x256	2x2	2	relu
Convolution	28x28x512	3x3	1	relu
Convolution	28x28x512	3x3	1	relu
Convolution	28x28x512	3x3	1	relu
Maxpool	14x14x512	2x2	2	relu
Convolution	14x14x512	3x3	1	relu
Convolution	14x14x512	3x3	1	relu
Convolution	14x14x512	3x3	1	relu
Maxpool	7x7x512	2x2	2	relu
FC	4096	-	-	relu
FC	4096	-	-	relu
FC	1000	-	-	Softmax

2.5.2 ResNet-50

The starting layers of the ResNet were similar to that of the Inception network, i.e, a convolutional layer of size 7x7 with 64 filters and stride S of size 2 which is followed by a pooling layer of dimensions 3x3 and stride 2. One of the key differences in ResNet is there exists a batch normalization layer after every convolutional layer [76]. The architecture of ResNet50 is depicted in Table. 2.2. The four modules which ResNet uses comprise of residual blocks. In this first residual block , the total number of channels is being doubled compared with the previous residual module and simultaneously the height and width are halved. Fig. 2.10 shows a regular block vs a residual block where the input is denoted by **x**. Assuming, the mapping reached by learning is $f(x)$.

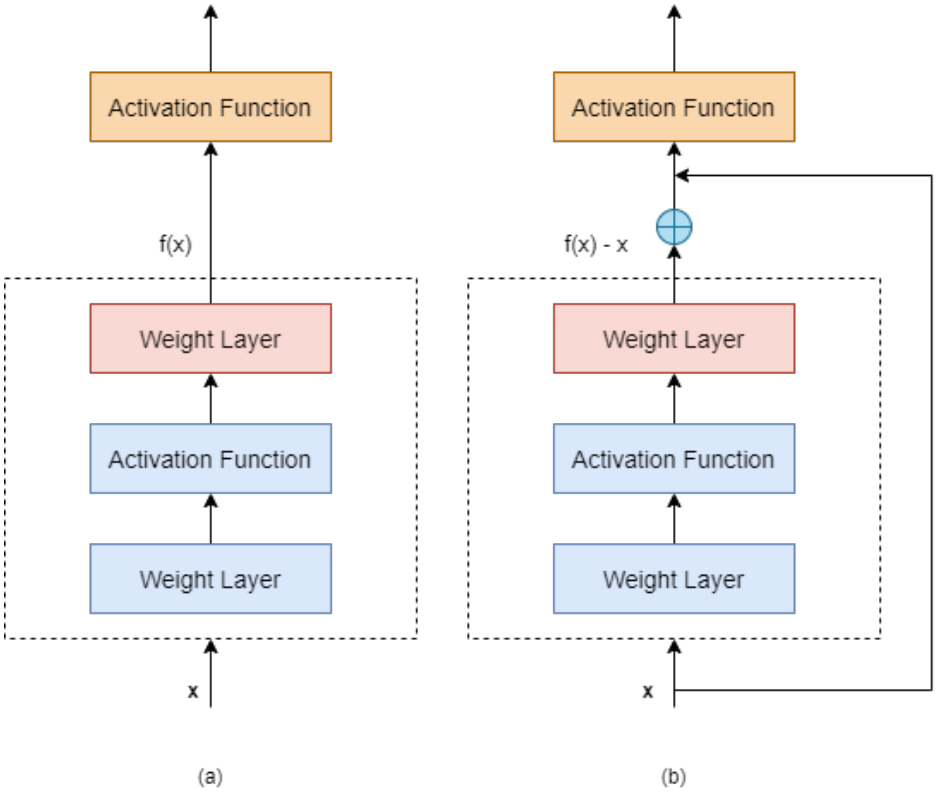

FIGURE 2.10: (a) Regular Block (b) Residual Block

2.5.3 Inception-V3

Inception networks [77] developed by google have proven to be much more computationally efficient than the former VGG networks both in terms of the number of parameters as well as the memory cost incurred. The use of these following

TABLE 2.2: Architecture of ResNet50

Layer	Output Size	18-layer	34-layer	50-layer
conv1	112x112	7x7, 64, stride 2	7x7, 64, stride 2	7x7, 64, stride 2
conv2	56x56	$\begin{bmatrix}(3,3), & 64\\(3,3), & 64\end{bmatrix}$ x 2	$\begin{bmatrix}(3,3), & 64\\(3,3), & 64\end{bmatrix}$ x 3	$\begin{bmatrix}(1,1), & 64\\(3,3), & 64\\(1,1), & 256\end{bmatrix}$ x 3
conv3	28x28	$\begin{bmatrix}(3,3), & 128\\(3,3), & 128\end{bmatrix}$ x 2	$\begin{bmatrix}(3,3), & 128\\(3,3), & 128\end{bmatrix}$ x 4	$\begin{bmatrix}(1,1), & 128\\(3,3), & 128\\(1,1), & 512\end{bmatrix}$ x 4
conv4	14x14	$\begin{bmatrix}(3,3), & 256\\(3,3), & 256\end{bmatrix}$ x 2	$\begin{bmatrix}(3,3), & 256\\(3,3), & 256\end{bmatrix}$ x 6	$\begin{bmatrix}(1,1), & 256\\(3,3), & 256\\(1,1), & 1024\end{bmatrix}$ x 6
conv5	7x7	$\begin{bmatrix}(3,3), & 512\\(3,3), & 512\end{bmatrix}$ x 2	$\begin{bmatrix}(3,3), & 512\\(3,3), & 512\end{bmatrix}$ x 3	$\begin{bmatrix}(1,1), & 512\\(3,3), & 512\\(1,1), & 2048\end{bmatrix}$ x 3

techniques like factorized convolutions, regularizations, dimension reduction, and parallelized computations have led this network to an optimal efficiency rate. The factorized convolutions help greatly in reduction of the computational efficiency by reducing the number of parameters involved, using smaller convolutions instead of 5x5 as shown in Fig. 2.11, can help the model converge faster. Table 2.3 shows the VGG-16 architecture.

TABLE 2.3: Architecture of InceptionV3

Layer	Patch Size / Stride	Input Size
Convolution	3x3/2	299x299x3
Convolution	3x3/1	149x149x32
Convolution Padded	3x3/1	147x147x32
Pooling	3x3/2	147x147x64
Convolution	3x3/1	73x73x64
Convolution	3x3/2	71x71x80
Convolution	3x3/1	35x35x192
3xInception	As shown in 2.11 (a)	35x35x288
5xInception	As shown in 2.11 (b)	17x17x768
2xInception	As shown in 2.11 (c)	8x8x1280
Pooling	8x8	8x8x2048
Linear	logits	1x1x2048
Softmax	classifier	1x1x1000

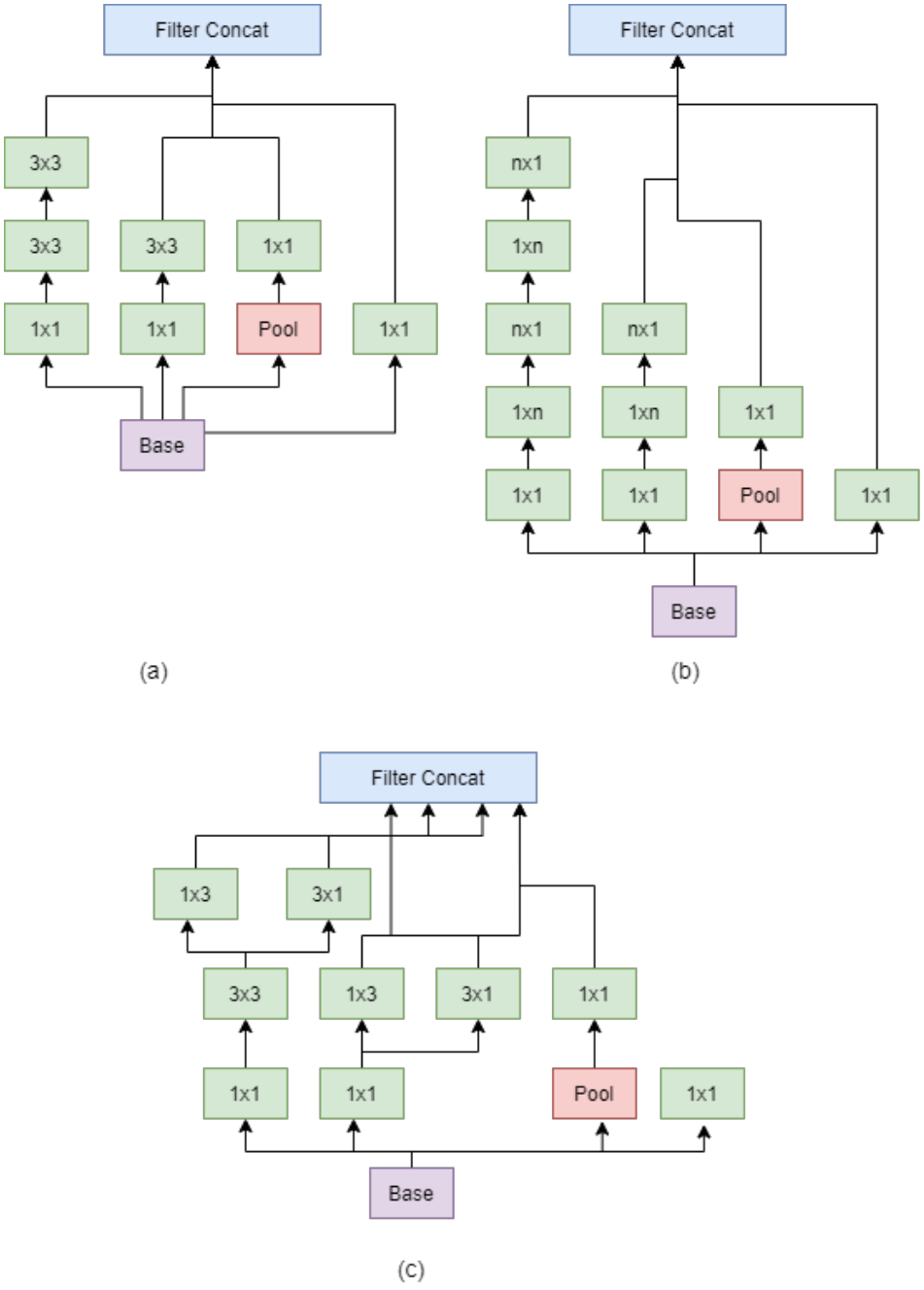

FIGURE 2.11: (a) Inception modules where each 5 × 5 convolution is replaced by two 3 × 3 convolution. (b) Inception modules after the factorization of the n × n convolutions. (c) Inception modules with expanded the filter bank outputs.

2.6 Generative Adversarial Network

Generative Adversarial Networks are a class of Machine Learning method based on using two CNNs which play a zero-sum game. Their usage is for semi-supervised learning, i.e. the role of man in the loop is less intensive as for the supervised CNNs which need examples on any particularity and peculiarity of the learned object. Given a training set, this technique learns to generate new data with the same statistics as the training set. Generative Adversarial Networks (GAN) were developed based on the principle of game theory. GAN consists of two separate networks i.e, the generator (G) and the discriminator (D). The Generator's G role is to generate as much fake data as possible by filling in the potential distribution of the real data. The discriminator's D role is to accurately classify real data from the fake data generated by the generator. The input to the generator is usually a uniform random noise. This noise is mapped to a completely new data space and the fake value is obtained $G(z)$. This vector $G(z)$ is a multi dimensional vector.

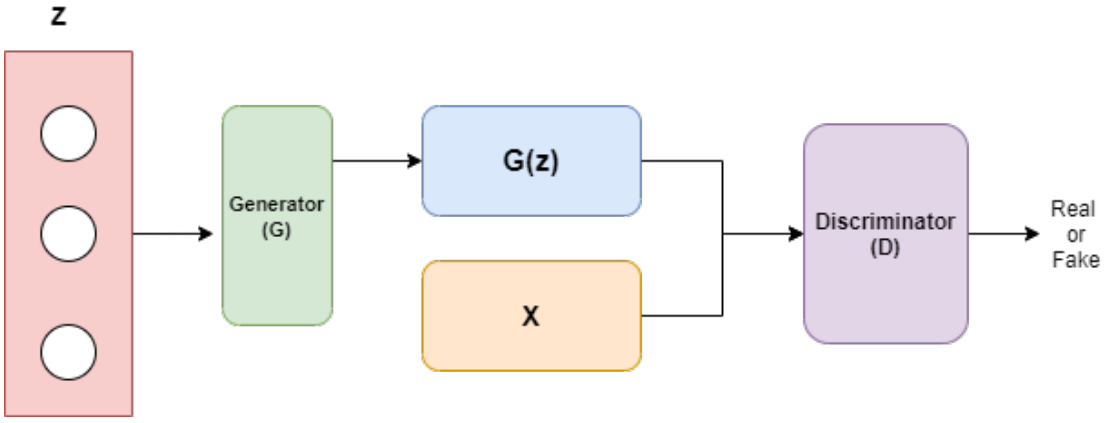

FIGURE 2.12: Architecture of the Generative Adversarial Network

The discriminator can be considered as a binary classifier as it takes the feedback from the real data and input from the generator which then gives the output which is a probability whether the sample is a real or fake. The optimal state is achieved when the discriminator is no longer able to identify the real data from the fake data. The loss function of the discriminator D can be computed using the cross-entropy as [30]

$$J^{(D)} = \frac{-1}{2} E_{x \sim pdata} \log D(x) - \frac{1}{2} E_z \log(1 - D(G(z))) \tag{2.28}$$

where $J^{(D)}$,$J^{(G)}$ are the discriminator and generator loss functions respectively, x represents the real sample data, z being the noise vector, $G(z)$ the data generated by the generator and E represents the expectation value. To reach the optimal point the value of $D(G(z))$ must approach zero and $G \rightarrow 1$. Thus, the loss of the generator is [30]

$$J^{(D)} = -J^{(G)} \tag{2.29}$$

$$min_G max_D V(D,G) = E_{x \sim pdata} \log D(x) + E_{z \sim p(z)} \log (1 - D(G(z))) \tag{2.30}$$

2.6.1 Architecture Optimization Based GANs

2.6.1.1 Convolution based GANs

For supervised learning, CNN is very widely adopted. The original GANs adopted the Multi Layer Perceptron (MLP) for the architecture of the generator and the discriminator. CNN being more efficient in feature extraction from images than MLP made GAN to be coupled with CNN leading to a Deep Convolutional Adversarial Networks (DCGAN) [78]. Fig. 2.13 shows the architecture of this DCGAN which replaces the dense layers or the fully connected layers with a deconvolution layer in the generator.

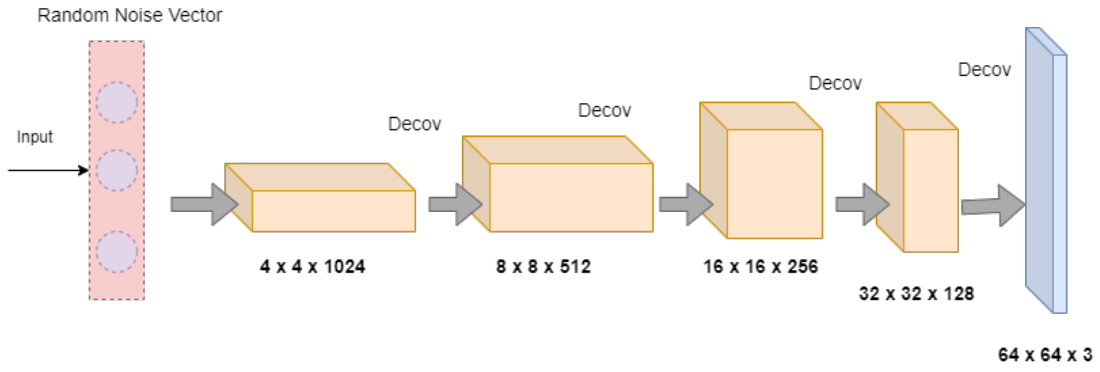

FIGURE 2.13: Architecture of the Generator in Deep Convolution GANs

2.6.1.2 Condition based GANs

The input of the generator being a random noise vector, there occurs a probability for the training operation to collapse because of the inputs being unrestricted in nature. To overcome this, a Conditional Generative Adversarial Network (CGAN) [79] was implemented in which a conditional variable c of the form

label, data etc. was induced in both the generator as well as the discriminator which enabled the variation in data generation process by adding the conditions to the model. Figure 2.14 shows the architecture of the CGAN in which the objective function could be defined as [79]

$$min_G\ max_D V(D,G) = E_{x\sim pdata(x)} \log D(x|c) + E_{z\sim p(z)} \log\left(1 - D(G(z|c))\right) \quad (2.31)$$

There were relatively a few models which incorporated this methodology. InfoGAN [80], a type of condition based GAN corrects the latent code c and the generated data x using mutual information. The key difference of this architecture Fig. 2.15 from CGAN is that the latent code c is an unknown and needs to be discovered while the training phase. The objective function can be computed as [80]

$$min_G\ max_D V_{info}(D,G,Q) = V(D,G) - \lambda L_I(G,Q) \quad (2.32)$$

here λ is a hyperparameter of the constraint function $I(c, G(z,c))$. A further new development was done with CGANs when and Auxiliary Classifier GAN (ACGAN) was implmented [81]. The condition variable c will not be added to the discriminator like the former approaches instead a new classifier will be used to compute the probability of the different class labels.

2.6.1.3 Autoencoder based GANs

Autoencoders are used in neural networks when a reconstruction phase from the input to output is required. The architecture of a typical autoencoder consists of two parts, an encoder $z = f(x)$ and a decoder $x = g(\hat{z})$. Here the input gets converted to the hidden layer (also known as the latent code z) by reducing the dimension, the decoder is further used to receive the code from the hidden layer h as the input which tried to reconstructs its input x after the training phase. Being an unsupervised learning model, as no labels are needed to train it this approach is prominently used in conjunction with latent variable model theory further applying to generative adversarial models. Autoencoder is also imperfect that the hidden layer obtained by the encoder is not evenly distributed in the specified space, resulting in a large number of gaps in the distribution. [30]. Combining the idea

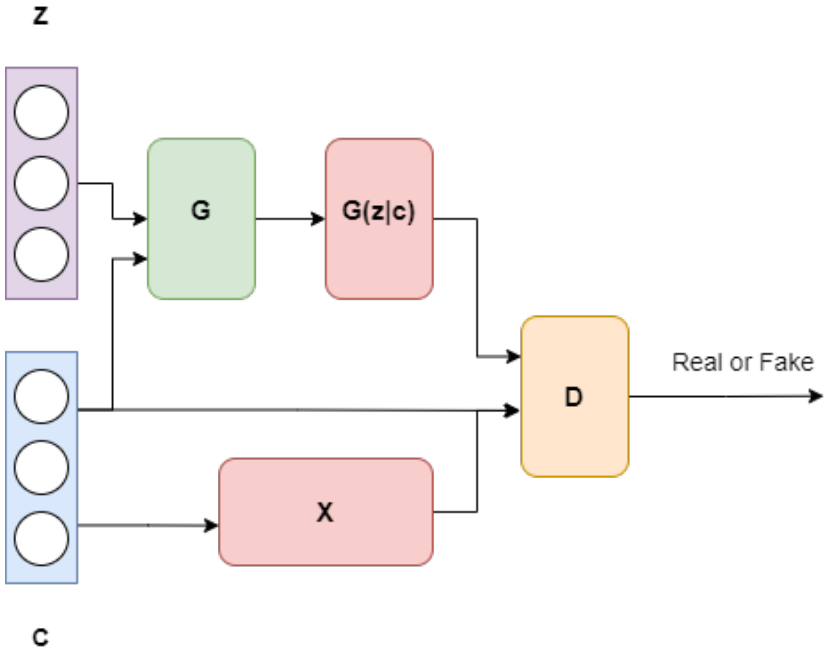

FIGURE 2.14: Derived GAN Model CGAN

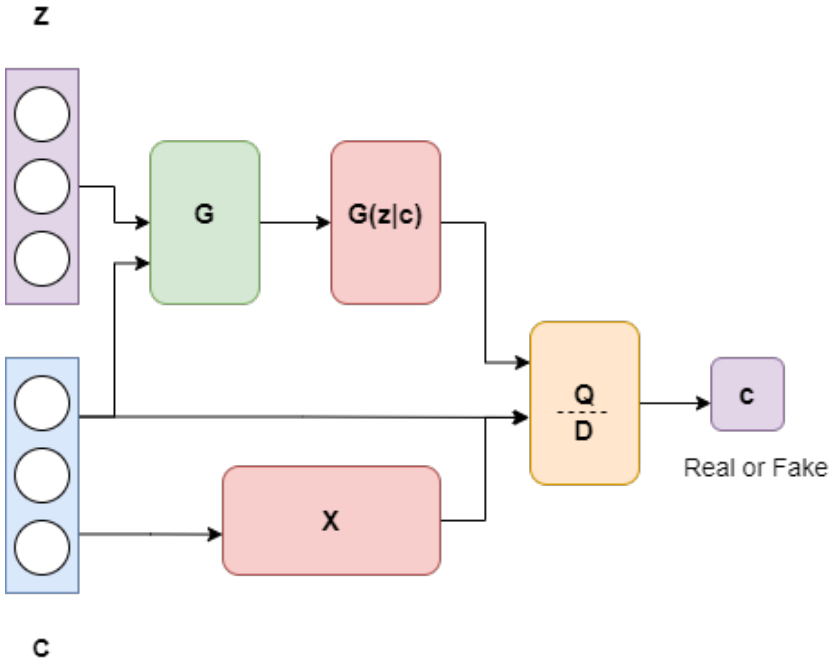

FIGURE 2.15: Derived GAN Model InfoGAN

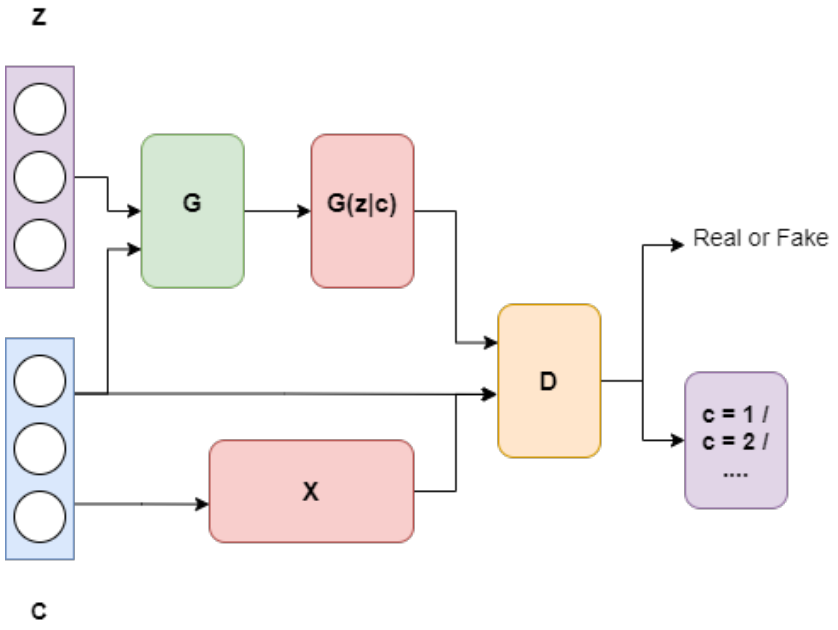

FIGURE 2.16: Derived GAN Model ACGAN

of the adversarial network into autoencoders Adversarial Autoencoder (AAE) [82] was developed where the random prior distribution is being forcefully fed on to the distribution of the latent code z obtained by the encoder to ensure that there are no gaps in the prior distribution, so that the decoder can reconstruct useful samples from any given part of it. The architecture of this AAE model can be seen in Fig 2.17 where the latent code represents the fake data and z' represents the specified distribution $p(z)$ which are the discriminator inputs.

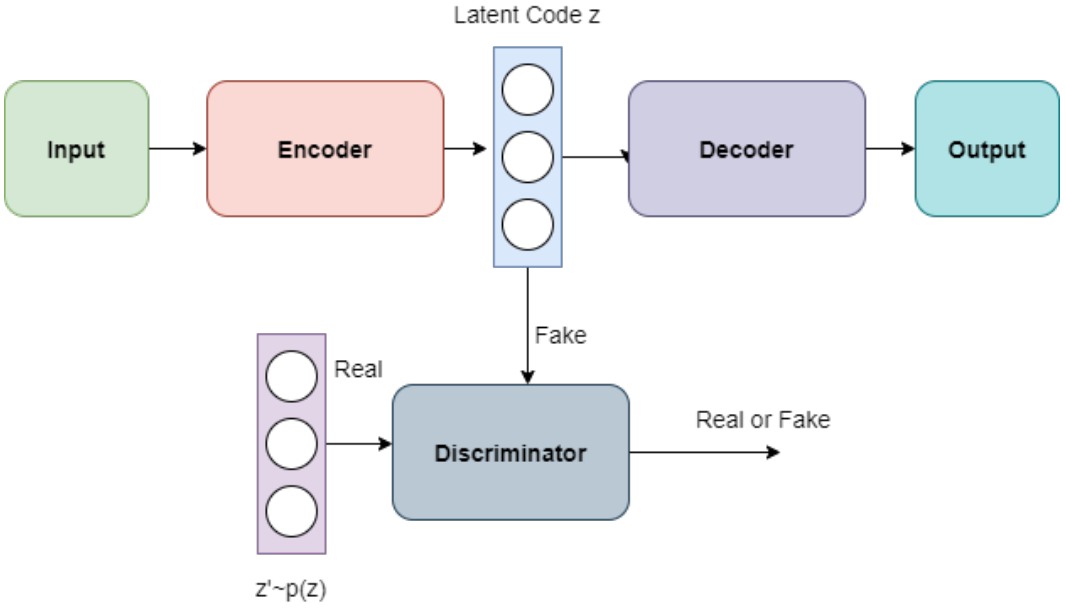

FIGURE 2.17: Architecture of the Adversarial Autoencoder

2.6.2 Objective Function Optimization Based on GANs

Optimization of the objective function is a high concern for achieving an optimal solution. Different methods for optimizing the objective function are discussed in this section. Metz et al. [83] proposed the unrolled GANs which use a gradient based loss function to enhance the generator, Jenson Shanon divergence method was used to minimize the loss function. Some other methods included using different regularizations. Che et al. [84] proposed two regularizers, trying to make the learning stable further stating if the overlap between the distribution of the generated data and the real data is minimum the divergence will be set to a constant value but this will make the gradient to reach zero creating a vanishing gradient problem. To overcome this problem a Wasserstein Generative Adversarial Network (WGAN) [85] was proposed showing the EM distance to produce better gradient behaviours in distribution learning compared to several other distance metrics.

The first original GANs tend to follow the approach where the discriminator has the ability to model infinitely without being held by any restrictions on the true sample (real sample) distribution which leads to overfitting and very poor generalization factor.

2.7 Dataset Requirements

These following datasets were collected for systematically and comprehensively evaluation various models for emotion classification tasks

1. **RAVDESS** [86] : The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) contains a total of 7356 files. The database comprises of a set of 24 professional actors including equal male and female actors who vocalizes two different lexically-matched statements in a neutral North American accent. Speech samples includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and songs consists of calm, happy, sad, angry, and fearful emotions. Every expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.

2. **WIDERFACE** [87] : This is one of the face-detection benchmarks dataset in which the images are selected from the publicly available WIDER dataset

which consists of a total of 32,203 images and label 393,703 faces with a high degree of variability in scale, pose and occlusion which are further organized based on 61 event classes. The dataset uses similar evaluation metrics employed in the PASCAL VOC dataset.

3. **RAF-DB** [88] [89] : Real-world Affective Faces Database (RAF-DB) is a large-scale facial expression database with around 30K great-diverse facial images. The dataset images in RAF-DB are of great variability in subjects, age, gender and ethnicity, head poses, lighting conditions, occlusions like glasses, facial hair and self-occlusion, post-processing operations etc. RAF-DB has ample diversities, large quantities, and rich annotations which further includes a 7-dimensional expression distribution vector for each image, 5 accurate landmark locations, 37 automatic landmark locations, bounding box, race, age range and gender attributes annotations per image and different set of baseline classifier outputs for basic emotions and compound emotions

4. **CEW** [90] : This dataset named Closed Eyes in the Wild (CEW) cotains a total of 2423 subjects, among which 1192 subjects with both eyes closed are collected directly from Internet, and 1231 subjects with eyes open are selected from the Labeled Face in the Wild.

5. **MRL** [91] : MRL eye dataset comprises of infrared images in low and high resolution, all captured in various lightning conditions and by different devices. A total of 37 individuals were used as subjects in the formation of the dataset including 33 males and 4 females comprising the total images in the dataset to be 84,898 images. Images were taken in different lightning condition, occlusions like glasses, gender, age and ethnicity.

6. **JAFFE** [92] [93] : The Japanese Female Facial Expression (JAFFE) dataset consists of 213 images of different facial expressions from 10 different Japanese female subjects. Each subject was asked to do 7 facial expressions (i.e, neutral, happy, sad, angry, fearful, surprise, and disgust) and the images are annotated with average semantic ratings on each facial expression by a total of 60 annotators.

7. **SMIC** [57] : SMIC database includes spontaneous micro-expressions elicited by emotional movie clips. Different emotional motion pictures clips were shown to the subjects to induce strong emotions and the subjects were asked to suppress their true feelings while watching the clips to create a lie-based

scenario where micro-expressions can be detected. This dataset contains 164 video clips from a total of 16 subjects.

8. **SAMM** [22] : The Spontaneous Micro-Facial Movement Dataset (SAMM) comprises of a total of 159 spontaneous micro-facial movements obtained through an emotional inducement experiment. 32 participants were chosen as the subjects from a wide diverse demographic who further were classified 13 different ethnicites with a mean age of 33.24 years and an even gender split with 17 male and 16 female participants.

9. **300-W** [94] [95] [96] : The 300-W is a face dataset that consists of 300 indoor and 300 outdoor in-the-wild images which further covers a wide variation of identity, expression, illumination conditions, pose, occlusion and face size. This dataset has a larger percentage of partially-occluded images than its counterparts. All the images are annotated using the 68-point mark-up using a semi-automatic methodology and there contain many images in the dataset with more than one face and has a diverse variety of face sizes.

10. **HELEN** [97] : The HELEN dataset is comprised of 2330 different face images having dimensions of 400×400 pixels with labeled facial components. These dataset images were generated through manually-annotated contours along eyes, eyebrows, nose, lips and jawline.

11. **LFPW** [98] : The Labeled Face Parts in-the-Wild (LFPW) consists of 1,432 faces from images downloaded from the web from various websites like google, flickr, yahoo etc. All images were further labeled using 3 MTurk workers and 29 points

12. **FERG-2D** [99] : The Facial Expression Research Group 2D Database (FERG-DB) is a database of 2D images of stylized characters with annotated facial expressions. The dataset comrpises of 55767 annotated facial images using 6 different stylized characters with gender disparity (3 males, 3 females) which were modeled using MAYA software and were rendered out in 2D. These images for each character are classified into seven types of expressions - anger, disgust, fear, joy, neutral, sadness and surprise.

13. **VoxCeleb** [100] : This dataset contains over 100,000 utterances for 1251 different celebrities which are extracted from videos uploaded to the YouTube. This dataset is gender balanced containing 55% males which spreads over

to different ethnicities etc. The videos included in the dataset are shot included in the dataset are shot in a large number of challenging multi-speaker acoustic environments [100] like outdoor stadiums, quiet studio interviews etc all of which contain the real world noise data.

TABLE 2.4: Summary of the datasets used

Name	**Application**	**Brief Information**
RAVDESS [86]	SER	Contains 7356 samples of speech and song with different expressions.
WIDERFACE [87]	Face Detection	Contains 30k images with 7 basic and 12 compound emotions.
RAF-DB [88] [89]	FER	Contains 32,203 images and label 393,703 faces.
CEW [90]	Eye Blink Detection	Contains a total of 2423 subjects in which 1197 images are closed eye images and the rest are of open eye.
MRL [91]	Gaze Tracking	Contains 84,898 images using a set of 37 individuals
JAFFE [92] [93]	FER	Consists of 213 images of different facial expressions from 10 different Japanese female subjects further classified into 7 emotions.
SMIC [57]	FER	Contains 164 video clips from a total of 16 subjects.
SAMM [22]	FER	Comprises of a total of 159 spontaneous micro-facial movements.
300-W [94] [95] [96]	FLD	Consists of 300 indoor and 300 outdoor in-the-wild images.
Helen [97]	FER	Comprises of 2440 faces images.
FERG-2D [99]	FER	Contains 55767 annotated facial images, classified into 7 emotions.
VoxCeleb [100]	GAN	Comprises of 100,000 utterances for 1251 celebrities.
LFPW [98]	FER	Comprises of 1,432 faces from website images.

2.8 Summary

In this chapter, an introduction to neural networks, the different layers in a convolutional neural network, generative adversarial networks to generate deepfakes, and a few transfer learning networks implemented to train the different CNN models in the next chapters, were discussed. The dataset requirements to train the convolutional neural networks models is listed in Table 2.4.

Chapter 3

A Novel Architecture for Emotion Detection

This chapter presents the high level design of a novel architecture for the analysis and identification of different emotions using CNN based classifier algorithms. The image-animation based emotion classifier introduced in this thesis, is a novel algorithm to detect emotions irrespective of gender and racial biases. The model can also be deployed to embedded systems with GPUs thus building less overhead costs than previously implemented algorithms. The architecture has been tested in both near-real time and real-time experiments. The architecture consists of 3 different blocks each of which is discussed in detail in the next chapters.

3.1 Motivations for a New Architecture

Recent advancements in the development of emotion classifiers have endeavored to achieve decent accuracy results as discussed in Chapter 2 but a stable emotion classification technique is still not extensively researched, devised and implemented. There were several hindering blocks encountered when implementing an emotion classifier which can be related to insufficient data leading to perform data augmentation further biasing the classifier. Similarly the face occlusions are creating a hindrance for the face detector networks, so are racial differences leading to a lack of capturing the context of the emotion etc. Human expressive behaviours in different real-time scenarios involve encodings from different perspectives and the facial expressions being treated as single modality. Also, as mentioned in

Section 1.2, the visualization of the CNNs used for emotion classification have demonstrated that there is a congruity between the features learned by the network and the ground truths. These are the facial areas defined by the Action Units (AUs) resulting in designing neural network filters that distribute the weights as per their significance in regards to their association with the facial muscle action parts. Due to the fore-mentioned issues, in this thesis different architectural components are devised to develop emotion classifier based on their use cases. A use-case based emotion classifier can be effectively deployed with good accuracy, thus justifying the purpose. The Eulerian Magnification algorithm for emotions using facial-microexpression cues can be deployed for lie detection systems, the multimodal network approach for emotion classification can be used in proctoring exams, interviews etc. The real time image to avatar based emotion classifier can be deployed in facial recognition systems, security etc.

3.2 Architecture Requirements

A series of functional and nonfunctional requirements capturing the functionality and the performance of the emotion detection and identification platform, listed below, were used for devising, developing and implementing a novel architecture for emotion detection and identification as introduced in the thesis.

1. **Functional Requirements** :

    1.1. The emotion detection and identification platform called EDIN shall be built in Python;

    1.2. The operating system in which EDIN runs shall be Windows 10; a Widows 8 system can be used as well;

    1.3. EDIN shall capture face pictures, video streams from a given RGB camera;

    1.4. EDIN shall be endowed with a microphone to capture speech sentences;

    1.5. The CNN developed for the face detection shall have the backbone architecture of SSDlite with MobileNet-V2;

    1.6. The CNN for face detection is common throughout the three core algorithms of EDIN;

1.7. EDIN core is based on three principle algorithms :

   i. Eulerian Video Magnification Model

   ii. Multimodal Network Model

   iii. Image-to-Avatar Animation Model

1.8. EDIN will detect ME's using the EVM by magnifying the user's expression;

1.9. EDIN further classifies the emotion into one of the seven universal emotion classes using a CNN for the emotion classifier;

1.10. A 50 Layer CNN shall be used as a backbone network for training the emotion classifier for the EVM algorithm;

1.11. A multimodal algorithm for emotion detection and identification shall be devised for being the one of the core algorithms of EDIN;

1.12. A total of 6 CNN's (face detection, facial landmarks detection, eye blink detection, gaze tracking, emotion classifier, speech emotion recognition) shall be used in the development of this thesis's multimodal neural network;

1.13. The backbone network used to train the facial landmarks CNN shall be based on XceptionNet architecture;

1.14. Eye points localization shall follow the detection of 68 point facial landmarks;

1.15. The CNN's for Eye blink detection, gaze tracking shall take the localized eye landmarks as inputs;

1.16. The multimodal EDIN system will allow the interviewer to accurately read the interviewee's emotions for a better assessment of his performance;

1.17. The image animation model used in the thesis shall run on a pre-trained GAN network;

1.18. The image-avatar animation of the EDIN system shall map the real time face movements on the avatar;

1.19. The emotion classifier used in the image-avatar EDIN system shall be applied on the avatar images;

1.20. A 50 Layer CNN shall be used as a backbone network for training the emotion classifier for the image-avatar animation algorithm;

2. **Non-Functional Requirements** :

2.1. **Usability**

i. The system can be used by interviewers, security officials for the detection and classification of the emotions;

ii. The software is segregated into three main blocks which should be deployed as per the use case. EVM model is used to detect ME's, mostly used for security purposes. Multimodal Network is used while administering interviews. Image-Animation is a real time emotion classifier and can be deployed in congruence with a facial recognition model for better recognition accuracy;

iii. The image-avatar animation EDIN system shall provide accurate classification of ME's of an average accuracy of 92.3%. This shall allow the model to be deployed in several use cases;

2.2. **Software Requirements**

i. A Python IDE similar to PyCharm is required for the software development as the entire code to train and deploy the models is based on Python Programming Language;

ii. Anaconda is used for creating virtual environments and channels;

iii. A 3D creation suite similar to Blender is used for 3D pipeline modelling, expressions rigging and for rendering;

2.3. **Hardware Requirements**

The specifications of the computer on which the models were trained and deployed.

i. Processor : Intel Core i7 9800X

ii. Hard Drive (HDD) : 4 TB

iii. Memory (RAM) : 32 GB

iv. GPU Card : Dual Nvidia GeForce RTX 2080 Ti

v. Camera and Microphone is needed for the face detection and speech synthesis.

3.3 Architecture and Component Design

Fig. 3.1 shows the architecture of the emotion detection and identification network (EDIN). The core of EDIN is based on the work taken by the following neural networks such as SSDLite MobileNet, ResNet50, Inception V3 which analyze people's audio or visual movements in order to identify one or more of the emotions from the list of emotions described in Chapter 1. The data on which the three Neural Networks perform their analysis is collected from a video camera or from any other video streams generator of either live or recorded audio video streams, and from a microphone either live or recorded.

1. **Stage-1**: The **Pre-Processing Block** is composed of a face detection module. A real time video or a stream is passed as an input to this stage. The face detection module is trained on SSDLite and MobileNetV2 joint networks for a high inference speed. This module is described in greater detail in Section 4.1 which discusses the architecture of the face detection module. The output of this stage is a stream of cropped faces. The face detection and pre-processing algorithm is used with all 3 components devised in this thesis.

   The **Pre-Processing Block** contains also a Speech Generator module which is used for generating designed sentences

2. **Stage-2**: The **Algorithm Block** is the heart of the EDIN architecture. 3 different algorithms are used at the core of the novel architecture proposed. These are:

   2.1. **Motion Magnification Algorithm**: This algorithm is used to increase the magnification of the facial micro-expressions from the stream of cropped face generated from Stage 1. The video stream is reconstructed constituting a delay factor thus, making this algorithm to perform in near real time pipeline.

   2.2. **Multimodal network**: this algorithm takes two inputs one being the cropped faces stream from Stage-1 and the other being an audio input. Using a combination of CNNs used for speech synthesis, facial micro-expressions detection, eye blink detection, and gaze tracking, EDIN produces weights for the aggregator network which produces the emotion ID.

2.3. **Image to Avatar based Animation Module**: This module like the previous algorithms takes the input as the stream of cropped faces and generates a deep fake motion for the avatar based on the facial key-points detected. The three blocks are further described in detail in Chapter 4, 5, 6 respectively.

3. **Stage-3**: Three CNNs are used for the emotion classification for each of the algorithms from Stage-2. The simple CNN and the Enhanced CNN for emotion classification in Stage 2 are trained on the RAF-DB dataset whereas the CNN fused with the image animation module is trained on the FERG dataset. The enhanced CNN used for the multimodal network is a fusion network which takes weights from the three CNNs.

4. **Stage-4**: The final output from Stage-4 predicts the emotion into one of the seven classes namely: happy, fear, sad, angry, disgust, surprise and neutral-the last emotion being still disputed by tne researchers in the psychology domain, as humans cannot be neutral in their emotion vis à vis a certain position they have to take.

   Motion Magnification Algorithm and the Multimodal Network, being computationally heavy, perform the inference in near-real time if the computer used is endowed with powerful GPU units, whereas the Image to Avatar Animation Module infers the results in real-time.

3.4 Summary

This chapter shows the high level design of the novel architecture implemented for the emotion classification and identification using different CNNs and algorithms. The various hardware and software requirements, functional requirements are listed in this chapter to devise the emotion classifier. The next Chapter 4 details in the implementation of a Motion Magnification based Emotion Classifier.

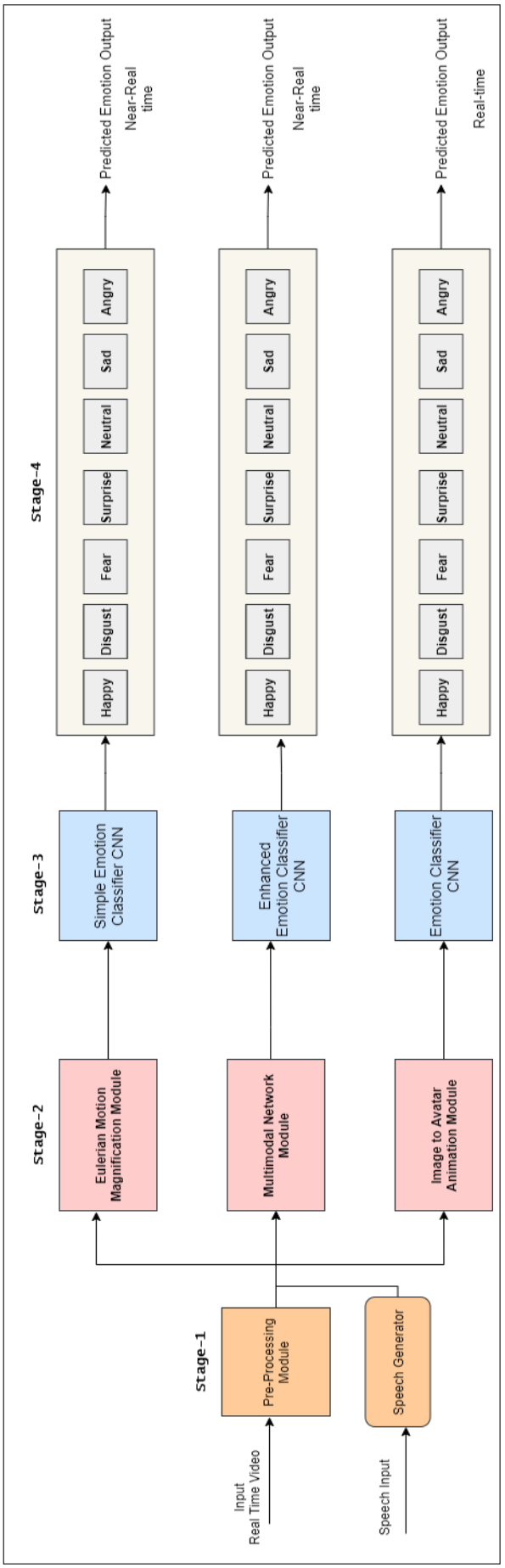

FIGURE 3.1: Architecture of EDIN

Chapter 4

Motion Magnification for Emotion Detection

In this chapter, an Eulerian Video Magnification (EVM) algorithm for the classification of the ME's was implemented. A set of developed parameter optimization methods and techniques that are part of the architecture, led to a stable feature detection algorithm. This algorithm helps in enhancing the subtle micro-expressions to a desired magnification factor which further increases the accuracy of the emotion classifier. A constrained set of values can be used for the amplification of the micro-expressions leading to diminishing the tendency for an error.

4.1 Single Shot MultiBox Detector for Face Detection

The Single Shot MultiBox Detector (SSD) algorithm comprises of two parts, the first being the extraction of the feature maps for which it uses Convolutional Neural Network for classification called VGG16 [75] which is followed by the usage of a set of convolution layers for the detection phase. Single Shot MultiBox Detector (SSD) [101] was designed for real time applications. Instead of using the traditional sliding window approach, SSD divides the image frame using a grid and every grid cell is responsible for the detection of the objects present in that region. The prediction is composed of a boundary box and scores for each class. The highest score for that specific class is selected. Many of these predictions can even contain

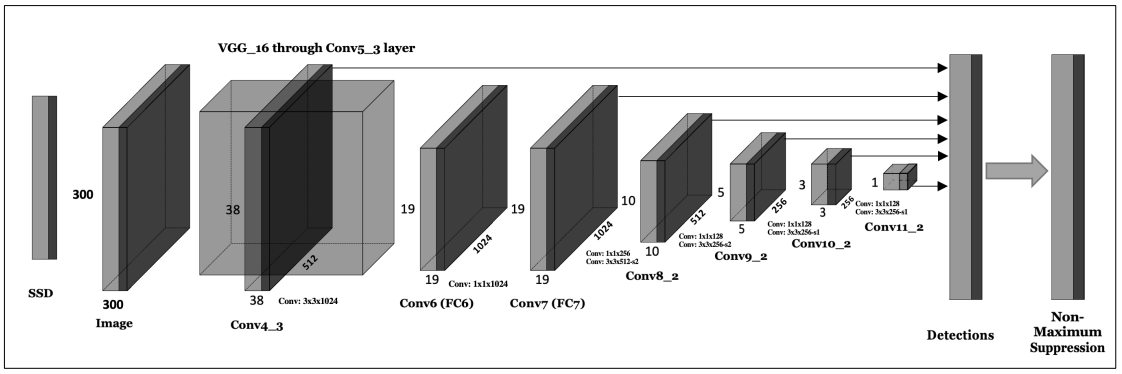

FIGURE 4.1: SSD Architecture

null object. SSD reserves a class "0" to depict these kind of objects. To deal with the problem of detecting multiple objects in a single frame SSD uses an anchor box. As not all the objects are necessarily in a square shape nor they have the same size as the grid cell, it is important for the anchor box to use a set of aspect ratios and zoom level to determine how much an anchor box's shape must be scaled. Fig. 4.1 depicts the architecture of the Single Shot Multibox Detector. The input image profile for SSD has the dimensions 300x300 [102] [103]. A linear increasing rule is adapted by the boxes i.e, the scale of the box is directly proportional to the size of the feature map which can be defined as

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m - 1}, k \in [1, m] \tag{4.1}$$

where m is referred to as the total number of the feature maps, s_k is the ratio of the size of the box which is relative to the picture. s_{min} and s_{max} represent the minimum and maximum values of the ratio which were 0.2 and 0.9 [101].

While training the SSD model, the default box is set a specific aspect ratio which is usually from $a_r \in \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$ The width and height of the box can be calculated as follows:

$$w_k^a = s_k\sqrt{a_r}, h_k^a = s_k/\sqrt{a_r} \tag{4.2}$$

The central coordinate of the default box with aspect ratio of 1 is set to

$$\left(\frac{i + 0.5}{|f_k|}, \frac{j + 0.5}{|f_k|}\right), i, j \in [0, |f_k|) \tag{4.3}$$

where f_k is the size of the feature map. Every grid cell can be associated with one or more anchor boxes which are pre-defined. The anchor boxes with the most

overlap with an object are matched with a bounding box using the ground truth for every image as a reference which further helps in predicting the object location and its associated class.

In this chapter, the devised model uses the backbone network of MobileNetV2 [104] paired with a framework of SSDLite for the face detection. SSDLite is a variant of the regular SSD where all of the regular convolutions are replaced by the separable convolutions in the prediction layers. Fig. 4.2 shows the block architecture of the MobileNetV2 [104]. MobileNetV2 is an improved version of its predecessor MobileNetV1 which uses depth-wise separable convolution as the building blocks [105] but has a couple of new features like linear bottlenecks between the layers and shortcut connections between the bottlenecks. Depth separable convolution can be treated as an accumulation of two operations with a (3x3) Depthwise Convolution and (1x1) size Pointwise Convolution.

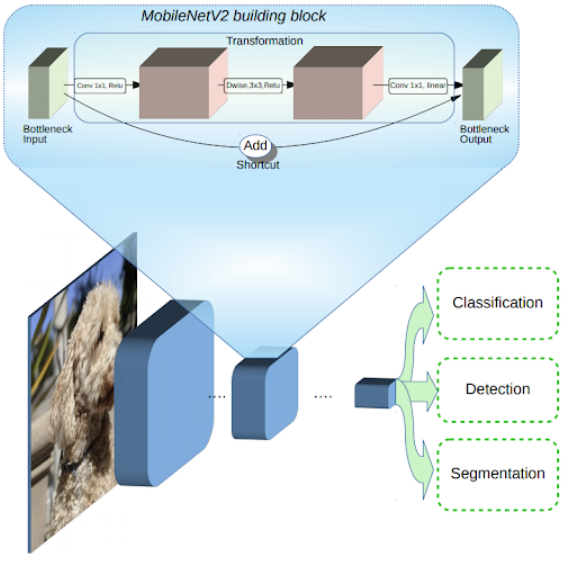

FIGURE 4.2: Overview of MobileNetV2 architecture

This inverted residual module with the linear bottleneck takes an input of a low dimensional compressed representation which is primarily expanded to high dimension and which is then filtered by a depth-wise convolution.

Table 4.1 shows the architecture of the MobileNetV2 network where t represents the expansion factor, n represents the number of repeating the operations, c depicts the total number of output channels and s denotes the strides. A combination of MobileNetv2 with SSDLite achieves a significant increase in the accuracy of the model prediction with a low computational complexity. The inference speed is also

TABLE 4.1: Architecture of MobileNetV2

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

comparably faster than its predecessor. Previously used face detection models included Haar-Cascades [106], Dlib Face Detector [107] etc. showed a poor accuracy during changes in the yaw and pitch angles because of which a need for a faster yet accurate model for face detection was needed which was achieved using the above mentioned approach.

4.2 Functional Requirements

A series of functional requirements capturing the functionality listed below, were used for devising, developing and implementing an emotion classifier based on EVM.

1. The EVM shall be based on two CNNs such as;
   1.1. SSDLite coupled with MobileNetV2
   1.2. ResNet50
2. SSDLite shall be used for the face detection;
3. ResNet50 shall be used as the backbone network for emotion classification;
4. The core of this specific component of the general architecture will be a motion magnification algorithm;

5. The EVM shall be responsible for cropping the video stream for faces;
6. The EVM shall amplify the face micro-movements;
7. The reconstructed video stream with the amplified signals on the face shall be stored in a memory at controlled locations;
8. The output of this algorithm shall be used by to detect and classify ME's;
9. The CNN for emotion classification shall read the video data stored by the EVM from memory for predicting an output label;

4.3 Eulerian Motion Magnification

The hypothesis made in this Section 4.3 is that the emotions are always accompanied by face, hands, and/or body movements. Some of these movements are difficult to see. Subtle changes which aren't visible to the naked eyes can be magnified using the Eulerian motion magnification (EMM) [102]. Unlike the Langrangian approach for motion magnification, EMM doesn't use the motion vectors which are needed to be estimated at a given time and location explicitly instead the amplitude or the phase on the whole image grid is magnified making it less prone to errors [108]. Traditional EMM approach used motion magnification for both large and small motions therefore increasing the computation speed as they were applied to the entire frame grid. Fig. 4.3 [86] shows motion magnification for different values of α which is the magnification factor. Amplitude Eulerian motion magnification (A-EMM) was applied on the output of the face detected from the SSD MobileNetV2 network in order to detect the subtle ME's.

Let $I(x,\ t)$ denote the image intensity at a spatial location x and time t. As this image has undergone a translational motion with the displacement function $\delta(t)$, the new image intensity profile is [102]:

$$I(x,\ t) = f(x + \delta(t)) \tag{4.4}$$

Given the motion is characterized with different intensities and $B(x, t)$ being the result of applying a temporal bandpass filer to the image intensity signal, the

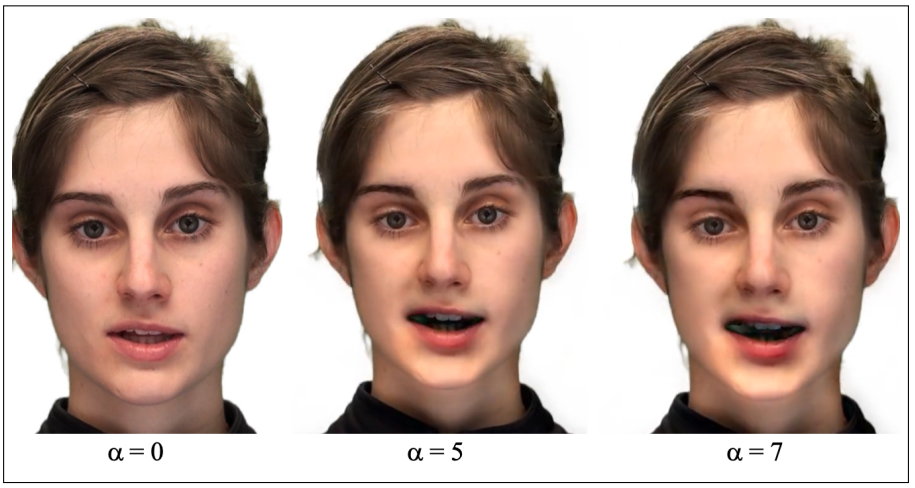

FIGURE 4.3: Motion Magnification with different values of α

computation of the pixel intensity I is done as:

$$\hat{I}(x,\ t) = I(x) + \alpha * B(x,\ t) \tag{4.5}$$

where α is known as the magnification factor. If $\delta(t) << 0$ implying that a small translational motion has occurred, $\hat{I}(x,\ t)$ can be computed using the first order Taylor Series as:

$$\hat{I}(x,\ t) \approx f(x) + \sum_{k} \alpha B(x,\ t) \tag{4.6}$$

where k defines the passband of the temporal filter with γ_k being the corresponding attenuation factor and $B(x,\ t)$ the output of the temporal bandpass filter which can be determined as:

$$B(x,\ t) = \sum \gamma_k \delta(t) \frac{\delta f(x)}{\delta x} \tag{4.7}$$

A-EMM when applied to a each pixel in its respective RGB channel of the video magnifies both amplitude as well as the color. Using a 2D isotropic Gaussian filter achieves spatial pooling i.e.,increasing the temporal signal to noise ratio:

$$G(x, y) = \frac{1}{\sum_{|x| \leq M, |y| \leq M} e^{-\frac{x'^2 + y'^2}{2\sigma^2}}} e^{-\frac{x^2 + y^2}{2\sigma^2}} \tag{4.8}$$

where $x, y \in I$ depict the centered pixel locations, M being the size of the filter and the normalization factor N is:

$$N = \frac{1}{\sum_{|x| \leq M, |y| \leq M} e^{-\frac{x'^2 + y'^2}{2\sigma^2}}} \tag{4.9}$$

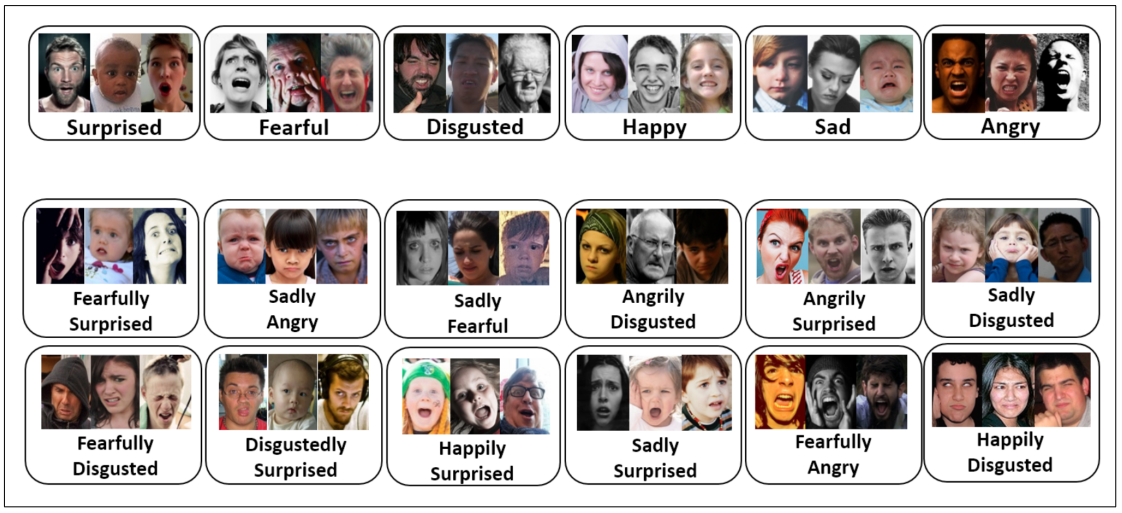

FIGURE 4.4: RAF-DB dataset

In order to reconstruct the image at the original resolution and get a decent approximation of the magnified signals the A-EMM magnification factor α [108] is bounded as $(1+\alpha) * \delta(t) < \frac{\lambda_c}{8}$. The optimal value of alpha is chosen to enhance only the microexpressions disregarding other signals amplification.

4.4 Emotion Classifier

A convolutional neural network was used to train the emotion classifier based on VGG16 [75]. A Deep convolutional neural network (DCNN) such as VGG16 can lead to prominent results if provided with a sufficient training data. Transfer learning can greatly help in the reduction of over-fitting problems when there is a lack of data. The VGG16 network used in this model leaves out the top layers of the fully connected layers which are connected to the output of the convolution base which further makes the network architecture flexible as the input size can be varied. Here the input image given to the network are of dimensions 64 x 64. The number of convolutions kernels of size (3x3) in the network begin with 64 in the first layer, 128 in the second and 512 in the third. Pooling is achieved using a step size 2 and (2 x 2) kernel. The batch normalization layers are removed in this implemented network model as these layers worsen model generalisation to out-of-domain data. The final dense layer is used with 7 class labels for the basic emotions and 11 for the compound emotions from the RAF-DB dataset.

4.5 Architecture Design EVM

To detect the micro-expressions and the emotions associated with them in real time the following architecture was devised as shown in Fig. 4.5.

In Stage 1, the real time video was passed through the face detector which comprised of MobileNetV2 and was paired with the framework of SSDLite used to detect the faces. In Stage 2, the detected faces were cropped for being passed on as an input to the band-pass filter eq. 4.7 to extract the subtle temporal changes which are then amplified using A-EMM with a magnification factor $\alpha \in [3, 6]$. These amplified signals are then added back to the original signal finishing the reconstruction with the same resolution i.e.,having the same amount of pixels. The output of this stage is a motion magnified video of the cropped face.

The motion magnified video is passed through Stage-3 block of 4.5, where a trained CNN performs the feature extraction. Using a Softmax loss function, an emotion is produced as an output belonging to one of the 7 classes of emotions. The algorithm used to detect and classify ME's using EVM is shown in 4.1. The simulation results for the experiment are presented in Chapter 7.

Algorithm 4.1 Algorithm for Emotion Detection using Motion Magnification

while $face = True$ **do** ssd face detection
    crop face
    while $frequency \in (0.5 - 1.5Hz)$ **do**
        perform spatial pooling using Gaussian filter $\sigma = 1$
        perform upsampling and downsampling,
        Choose the magnification factor $\alpha \in [3, 6]$ so that $(1 + \alpha) * \delta(t) < \frac{\lambda_c}{8}$
        perform ampltitude magnification using EVM
        reconstruction of the frame
        detect micro-expressions using CNN **return** emotion label
    end while
end while

The face detector is trained on WIDER FACE Dataset [87] containing ≈32000 images and the emotion classifier is trained on Real-world Affective Faces Database(RAF-DB) [88] [89] consisting of 29672 real world images with 7 basic emotions and 11 compound emotions. These compound emotions are basically a combination of two different basic emotions.

A confusion matrix is the ideal choice for a performance metric while solving a classification problem. This provides the classification results in a matrix format

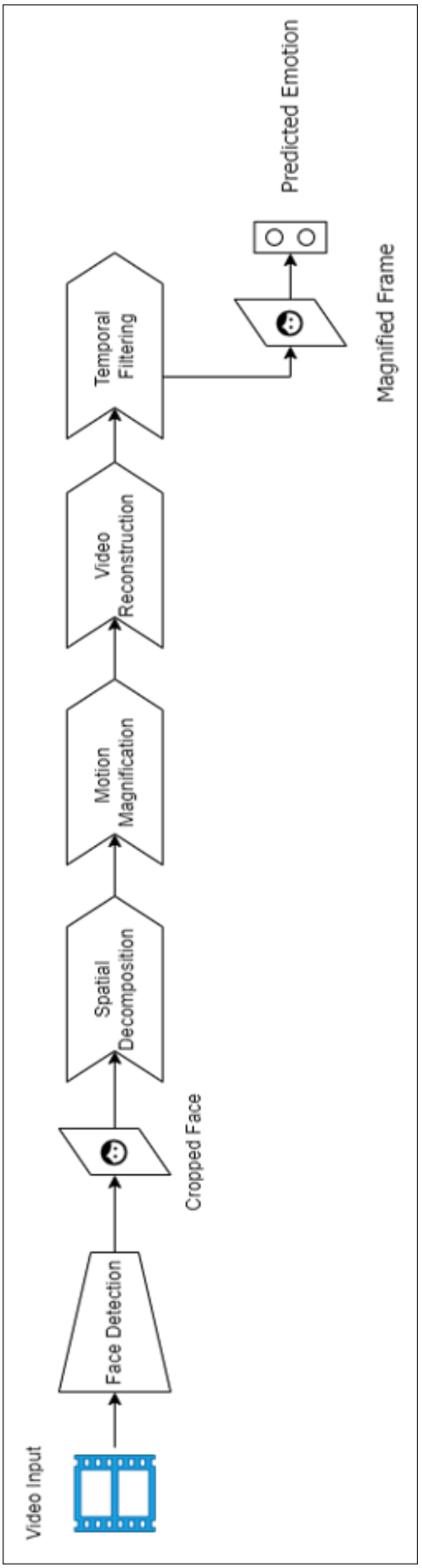

FIGURE 4.5: Architecture of the EVM Emotion Classifier

which contains all the predicted and the wrongly predicted classes. The average of the primary diagonal gives the validation accuracy of the model. Diagonal elements represent the total examples of t_n and t_p where there label is the same of their true label whereas other matrix elements represent the examples which our emotion classifier has wrongly predicted.

4.6 Summary

In this chapter, the implemented algorithm using Eulerian motion magnification accentuated the intensity of the subtle expressions further boosting the emotion classification prowess. The algorithm consistently performs better than the baseline and current algorithms for emotion classification by achieving an accuracy of 88.47%. By using two different strategies concurrently and a state of the art face detector a near real time model was achieved. With this framework in place, the future work includes usage of multimodal approach for emotion classification shown in Chapter 5 .

Chapter 5

Multimodal Convolutional Neural Networks for Emotion Classification

Combination of different audio-visual cues have been quite prominent in recent years [109]. Studies have shown that when provided a noisy environment, the audio wave-forms can distort a lot making it difficult to predict the emotion using the facial muscular movements alone. In this chapter a novel algorithm for multimodal aggregator network for emotion identification was devised comprising of a facial micro-expressions and speech emotion recognition synthesizer.

5.1 Functional Requirements

A series of functional requirements capturing the functionality listed below, were used for devising, developing and implementing an emotion classifier based on a multimodal neural network.

1. The multimodal network shall be comprised of 3 different core CNNs such as;
    1.1. SSDLite coupled with MobileNetV2
    1.2. ResNet50
    1.3. XceptionNet

2. Audio streams shall be stored in a dedicated memory space for predictions;
3. Video streams shall also be stored in a dedicated memory space for predictions;
4. The eye-blink detection shall be performed using a custom made CNNs as discussed in Section 5.4 on the saved video stream;
5. The gaze tracking shall be performed using another custom made CNNs as discussed Section 5.3 on the saved video stream;
6. The Enhanced Emotion Classifier shall be based on an aggregator network with its input parameters being the weights from;
   (a) Eye-Blink CNN
   (b) Gaze Tracking CNN
   (c) ResNet50 based Emotion Classifier
7. The speech synthesis module shall attempt to predict the emotion from the stored audio stream of the user speeches in order to increase the emotion identification probability;
8. The core of this newly devised architecture will be the final Multimodal Aggregator Network (MAN);
9. The MAN shall comprise of the input weights calculated from the FER and SER data;
10. The output of the MAN shall be the predicted emotion;

5.2 Facial Landmarks Detection

As CNN technologies evolved on both the hardware and software components more and more researchers dedicated research time and efforts to deciphering the human emotions. Associated with the above, the need for the development of an accurate facial landmark detection model was, and still is, crucial. In this thesis the approach is set for the detection and identification of micro-expressions which humans cannot control and which are at the basis of one of the six (6) or seven (7) human emotions [110]. Traditional methods were based on holistic approaches

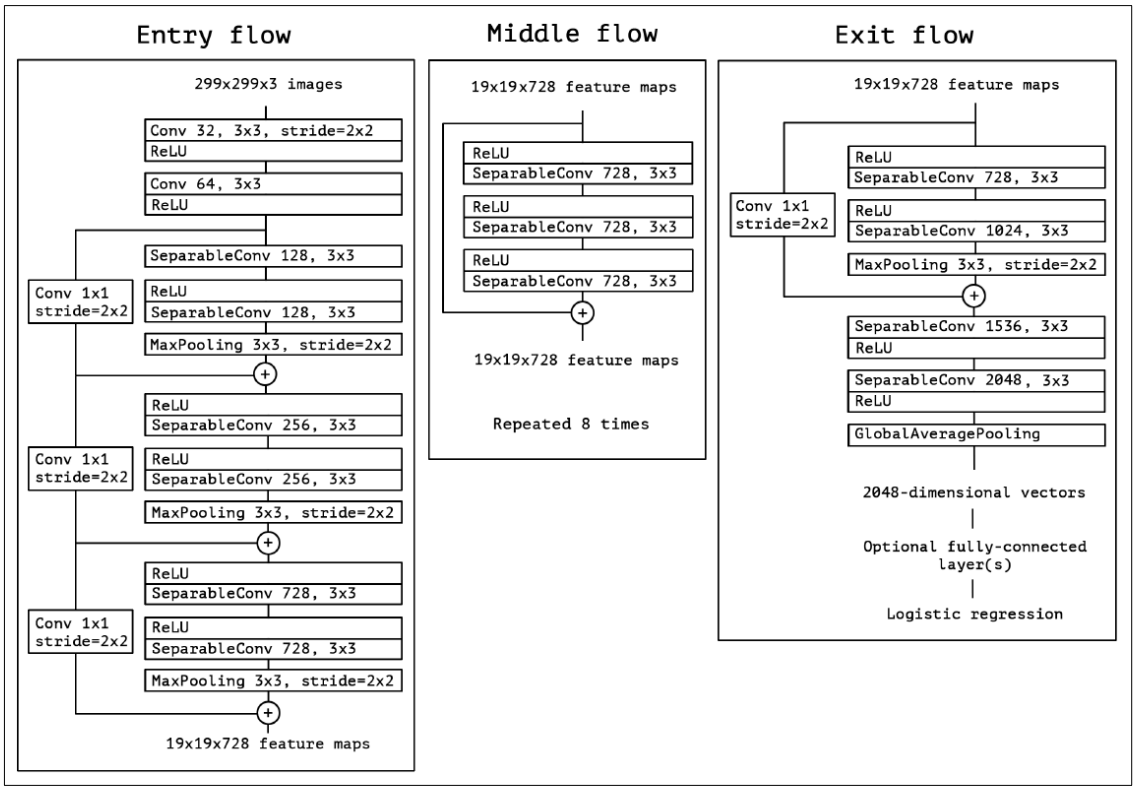

FIGURE 5.1: Architecture of XceptionNet

i.e, the landmarks positions were considered with a reference to the face as a whole leading to high inaccuracy. This thesis proposes that the micro-emotions detection and identifications shall be solved on using multiple CNNs. Typically, a CNN learns filters in the 3 dimensional space using 2 spatial dimensions which are the height and the width, along with a channel. This accounts for the CNN filter to perform simultaneously the mapping across the different channel and spatial correlations. In the proposed model, the training network derived from XceptionNet [111] was used. This makes the decoupling of the detected emotion stronger. The name was derived from its predecessor "Extreme Inception" [112].

Figure 5.1 [111] shows the architecture of the XceptionNet comprising of three major blocks the entry flow, the middle flow and the exit flow. The entry flow has a total of 8 convolutional layers. The middle flow or the middle block is comprised of 8 blocks, each block having 3 convolutional layers which makes a total of 24 CNN's in the middle flow. The final block or the exit flow consists of 4 convolutional layers which makes the entire network consisting of 36 different convolutional layers.

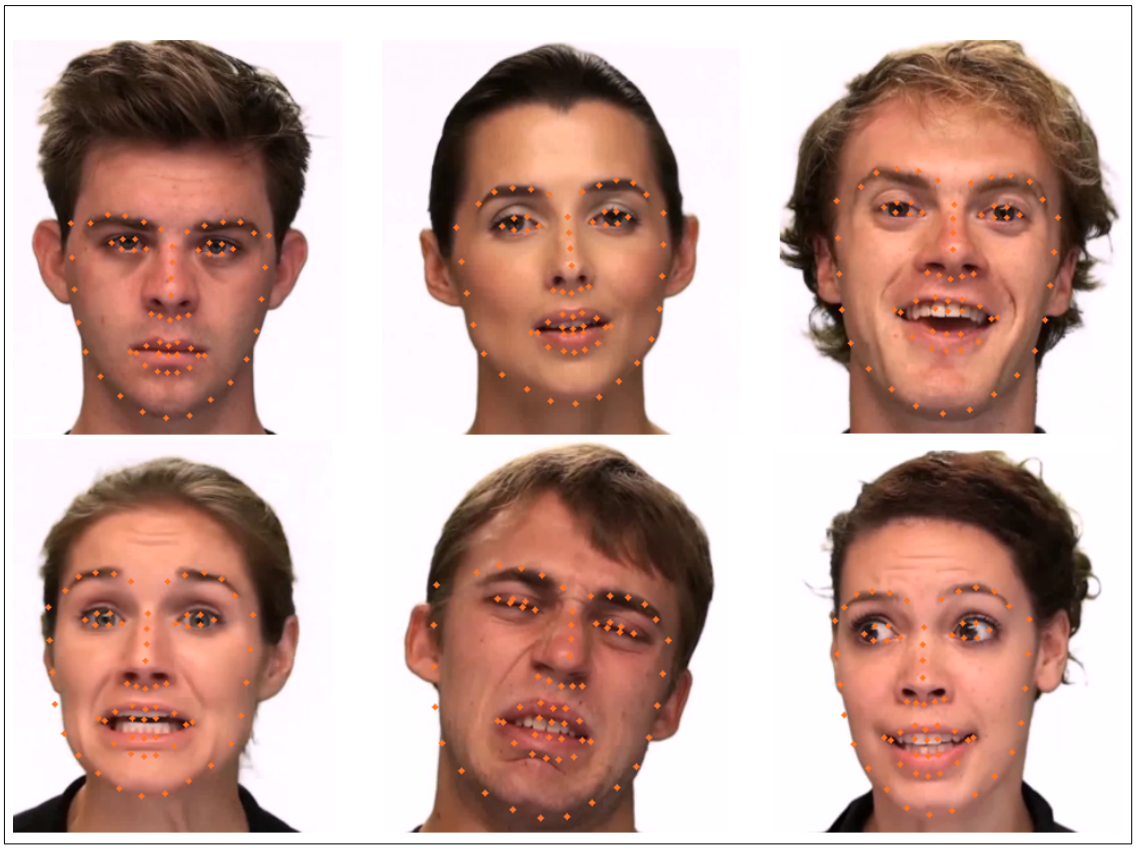

FIGURE 5.2: 68 Facial Landmarks on Faces

The first channel "wise" spatial convolutions are being performed by the depth-wise separable convolutions without any non-linearities followed by the point-wise convolutions. The training of this network was carried on the ibug300W Dataset [94] [95] [96] which detects the 68 different landmarks on the face as shown in Fig. 5.2 and the final loss function used here is of the type L2 (mean-squared error) given by the eq. 5.1 where i is the index of the sample $\hat{y}$ is the predicted value, y is the expected value and n is the total number of samples in the dataset.

$$loss = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 / n \tag{5.1}$$

5.3 Convolutional Neural Network for Gaze Tracking

Appearance based gaze tracking is achieved using the eye localised landmarks which were identified from the facial landmarks CNN and the eye image is taken as the input of the network. A novel structure is designed to estimate the gaze point and the gaze direction. MRL Eye Dataset [91] has been used to train the convolutional neural network to estimate the gaze direction which showed an accuracy of 89.97%. This dataset contains infrared images in different resolutions and intensities making it suitable for the purpose of the thesis. The loss function used to assess the model was the Mean Absolute Error (MAE).

MAE is a defined metric (measure) which is used to assess the average absolute distance between the actual values and the predicted values.

Let as assume a vector of n different true values, y and a vector of n predicted values $\hat{y}$. Then, the MAE would be defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5.2}$$

As there are two output predictions for every image x,y MAE will be associated with both of them MAE_x, MAE_y. An average metric of the former can be used to represent the model performance. Also the euclidean distance between the actual and the predicted coordinate, E_D is calculated by

$$E_D = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \tag{5.3}$$

5.4 Convolutional Neural Network for Eye-Blink

Traditional approaches used the concept of Eye Aspect Ratio (EAR) which used a scalar value that responds predominantly to the opening and closing of the eye. The output aspect ratio is computed using the following equation [113].

TABLE 5.1: Architecture of Convolutional Neural Network for Gaze Tracking

Layer	**Kernel Size**	**Stride**	**Number of Filters**	**Output**
Conv1	(3,3)	(1,1)	32	(64,64,32)
Pool1	(2,2)	(2,2)	-	(32,32,32)
Conv2	(3,3)	(1,1)	64	(32,32,64)
Conv3	(3,3)	(1,1)	64	(32,32,64)
BatchNorm1	-	-	-	(32,32,64)
Pool2	(2,2)	(2,2)	-	(16,16,64)
Conv4	(3,3)	(1,1)	128	(16,16,128)
Conv5	(3,3)	(1,1)	128	(16,16,128)
Conv6	(3,3)	(1,1)	128	(16,16,128)
BatchNorm2	-	-	-	(16,16,128)
Pool3	(2,2)	(2,2)	-	(8,8,128)
Conv7	(3,3)	(1,1)	256	(8,8,256)
Conv8	(3,3)	(1,1)	256	(8,8,256)
Conv9	(3,3)	(1,1)	256	(8,8,256)
BatchNorm3	-	-	-	(8,8,256)
Pool4	(2,2)	(2,2)	-	(4,4,256)
FC1	-	-	1024	1024
FC2	-	-	512	512
FC3	-	-	2	2

Algorithm 5.1 Algorithm for Gaze Tracking

while $face = True$ **do** ssd face detection
    crop face
    landmark detection using XceptionNet
    localize eye landmarks,
    Estimating the gaze vector using CNN
    Calculating $E_D = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}$
    Computing error in prediction using MAE
    return gaze vector
end while

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\,\|p_1 - p_4\|} \tag{5.4}$$

where the parameter points p_1, p_2, p_3, p_4, p_5, p_6 could be inferred from the Figure 5.3. EAR method being extremely fast in inference was yielding a poor validation accuracy because of which a deep learning approach was needed. A convolutional neural network was implemented which was trained on Closed Eyes in the Wild

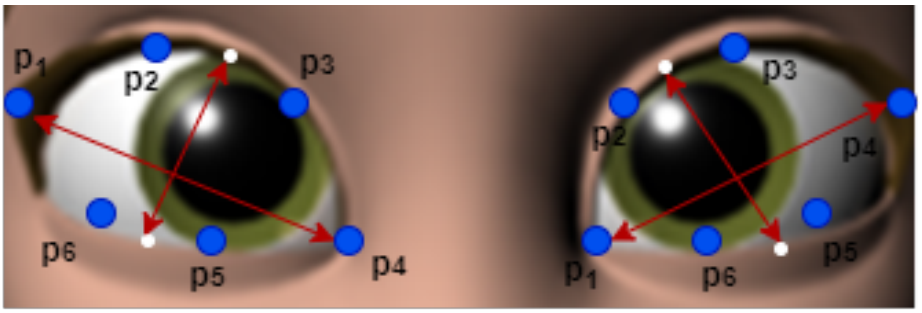

FIGURE 5.3: Eye Aspect Ratio

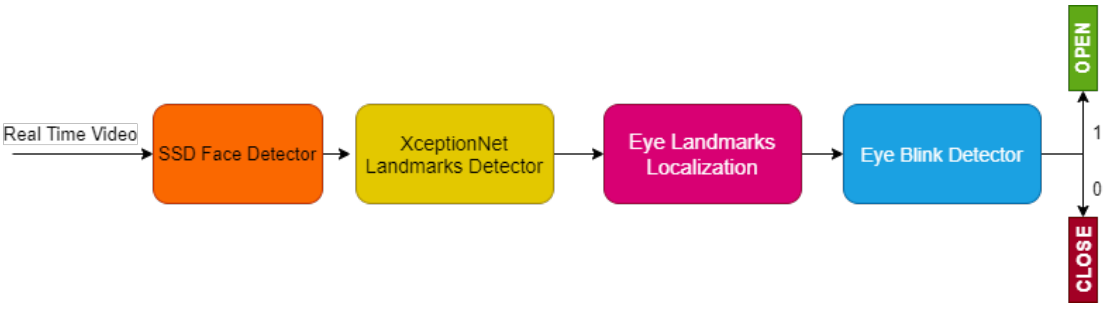

FIGURE 5.4: Architecture of the Eye State Classifier

(CEW) dataset [90] for the eye states classification. Fig 5.4 shows the states process to detect the eye state. The classifier predicts the final state as "Open" or "Close". The final validation accuracy of the model after being trained on 19 epochs was 99.3%.

5.5 Emotion Classifier using Speech Synthesis

Extracting and analyzing the emotion from the speech of an individual is called speech emotion recognition (SER). This section introduces the detection of the emotional state of the individual using speech [114]. The speech analyzer extracts an ample amount of the acoustic characteristics from the speech input signal and analyzes it without causing a hindrance to the acoustic properties. The following section shows the necessary steps involved to produce a state of the art audio-based emotion classifier.

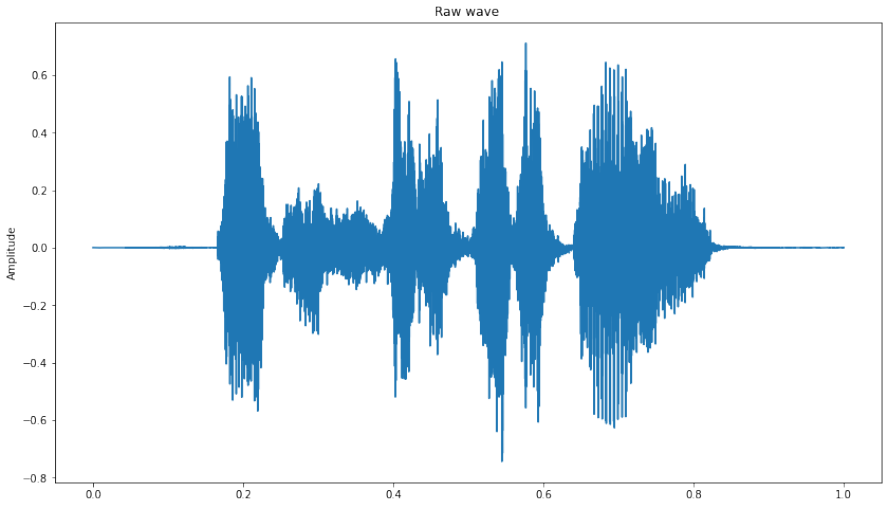

FIGURE 5.5: Original Raw Audio Wave

5.5.1 Data Augmentation

The dataset used to train the convolutional neural network was RAVDESS [86]. This database is gender balanced comprising of 24 participants. 2451 samples from the dataset were chosen in which speech included calm, happy, sad, angry, fearful, surprise and the disgust expressions whereas the song contained calm, happy, sad, angry and fearful emotion. To convert the analog speech signals into a discrete time domain a sampling operation was performed as its easier for the computer to process [115]. The relationship between the sampling frequency f_s and the time period T is given below.

$$f_s = \frac{1}{T} \tag{5.5}$$

All the signals are timed for exactly 3 seconds so that they contain an equal number of features.

Data Augmentation can be used by changing the pitch and speed of the samples. Similarly, a noise DA technique i.e, a noise signal having a zero mean value with a Gaussian distribution function as base, can be added to the audio samples to further increase the size of the dataset. Researchers [116] [117] use AWGN as a noise model to test system robustness and performance improvement of SR and SER.

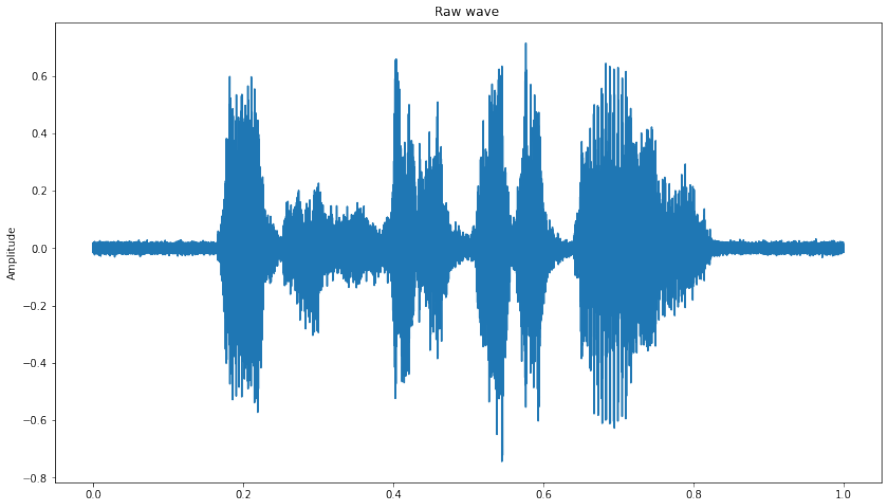

FIGURE 5.6: Noise Augmentation on Original Wave

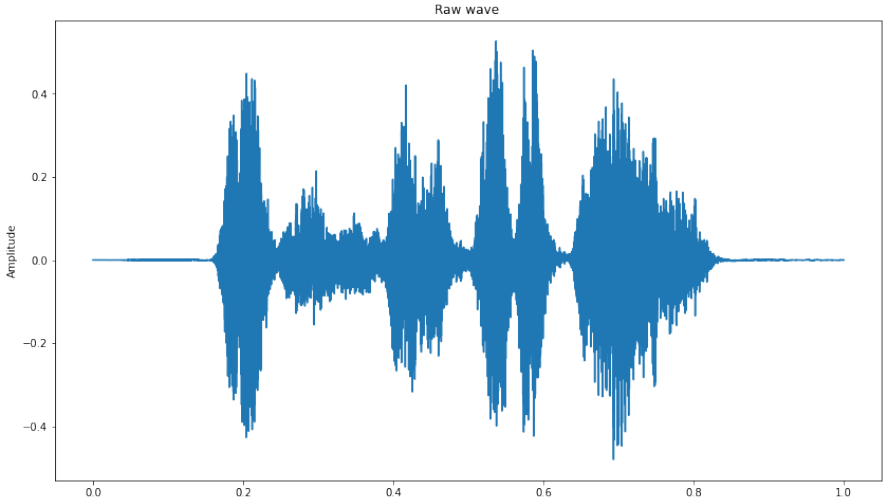

FIGURE 5.7: Pitch Augmentation on Original Wave

5.5.2 Mel Frequency Cepstral Coefficients Feature Extraction

Mel Frequency Cepstral Coefficients (MFCC) feature extraction is composed of 3 stages. The coefficient's computation is quite similar to how humans perceive the speech signals. Speech signals have tones which varies with frequency, each tone with an actual frequency f (Hz) [118] and the pitch which is being computed on the Mel Scale. The Mel-frequency scale has a linear frequency for the range below 1000 Hz and a logarithmic spacing above 1000 Hz, Pitch of 1 KHz tone. 40 dB above the perceivable threshold is defined as 1000 mels, which is used as a reference point. MFCC can efficiently characterize low frequencies compared to high frequency regions and because of this it is being widely used in the Speech

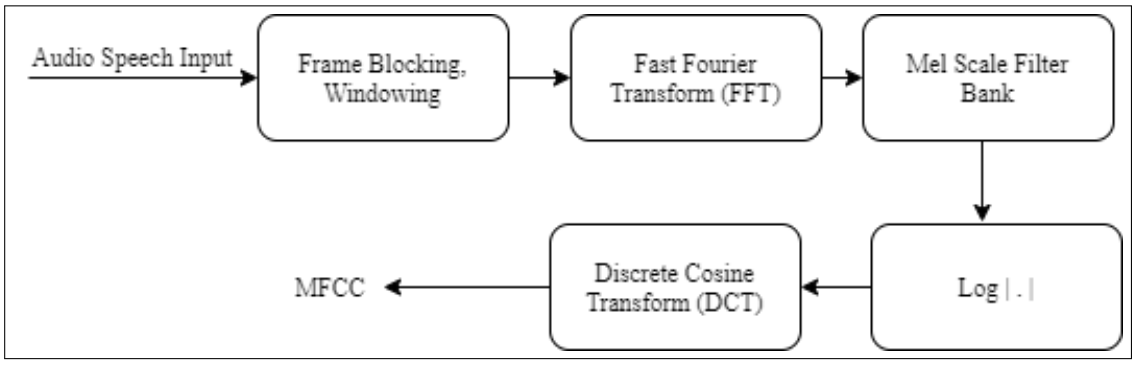

FIGURE 5.8: MFCC Block Diagram

Synthesis applications. In the first stage of the computation of MFCC, windowing is performed to split the audio signal into small frames on which they are further processed. The resultant signal from this stage is of the form :

$$Y(n) = x(n).W(n), \quad 0 \leq n \leq N-1 \tag{5.6}$$

$$W(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1. \tag{5.7}$$

where the window function $W(n)$ is of a finite duration sequence of length N. In this thesis, a Hamming window has been implemented which can be defined as shown in eq. 5.7. The hamming window being an extension of hann window approximates the value of $\alpha \approx 0.54$ leading to eq. 5.7. The second stage deals with the Fast Fourier Transform (FFT) which is applied to calculate the power spectrum of each individual frame which is followed by the filter bank processing using mel-scale. The formula which is used to compute mels for any frequency [119] is shown in eq. 5.8. Fig. 5.9 shows the Mel-spectogram of the 7 universal emotions.

MFCC coefficients are calculated after the discrete cosine transform (DCT) which is applied to the speech signal after the translation of the power spectrum to the log domain. Fig. 5.8 shows the block diagram of MFCC processor. The final MFCC coefficients equation can be calculated using eq. 5.9.

$$f_{mel} = 2595\log_{10}(1 + \frac{f}{700}) \tag{5.8}$$

$$C_{\mathrm{m}} = \sqrt{\frac{2}{Q}}\sum_{l=0}^{(Q-1)}\log[e(l+1)]\cdot\cos\left[m\cdot\left(\frac{2l-1}{2}\right)\cdot\frac{\pi}{Q}\right] \tag{5.9}$$

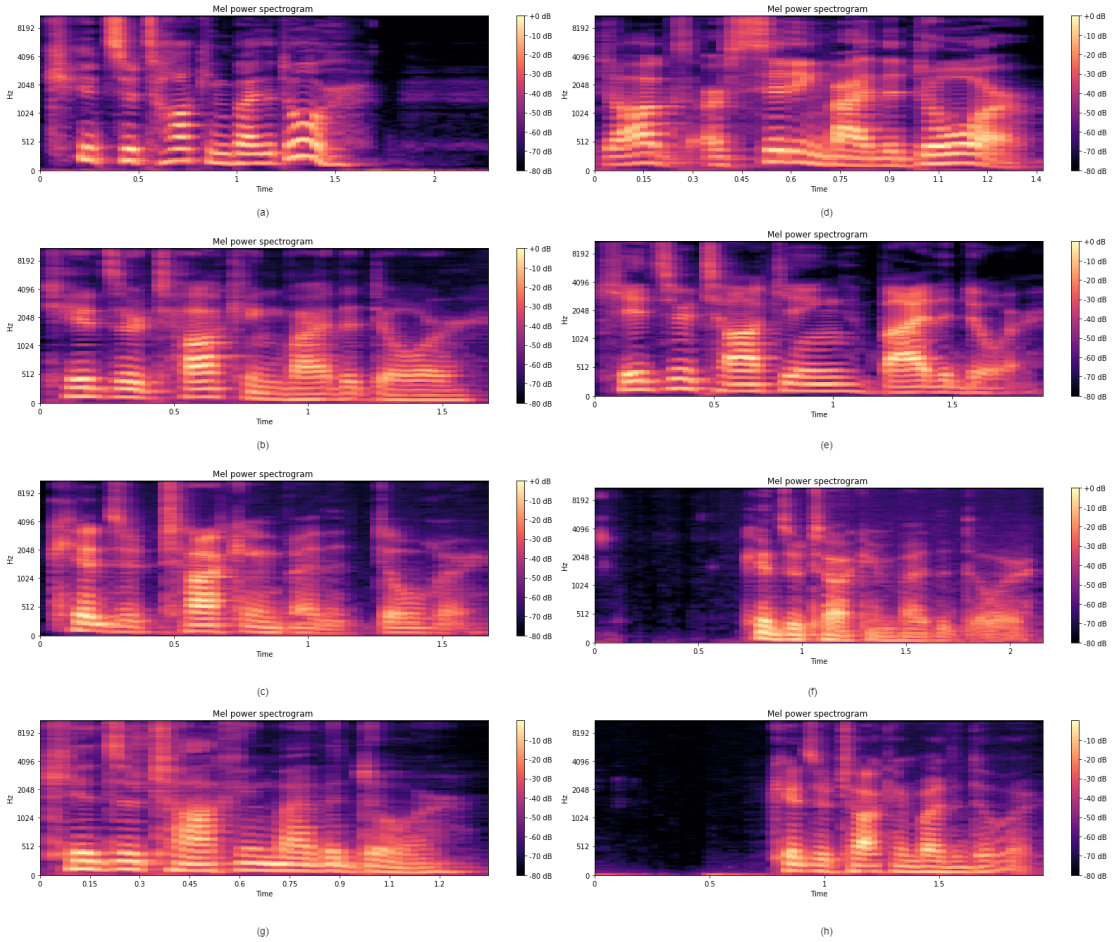

FIGURE 5.9: Mel Power Spectogram for different emotions.(a) Anger, (b) Calm, (c) Disgust, (d) Fearful, (e) Happy, (f) Sad, (g) Neutral, (h) Surprise

where f is the actual frequency (in Hz), $0 \leq m \leq R$-1, R is the desired number of cepstral features.

5.6 Enhanced Emotion Classifier

In Chapter 4, the emotion classifier was trained on RAF-DB dataset with a validation accuracy of 78% 4.4. To improve the accuracy of this classifier the following modification to the network are implemented. The weights generated from the 5.3 and 5.4 can be coupled and associated with the seven universal emotions by using a simple aggregator network which increases the validation accuracy of the model to 82.2%. Fig. 5.11 shows the devised implementation of the enhanced emotion classifier.

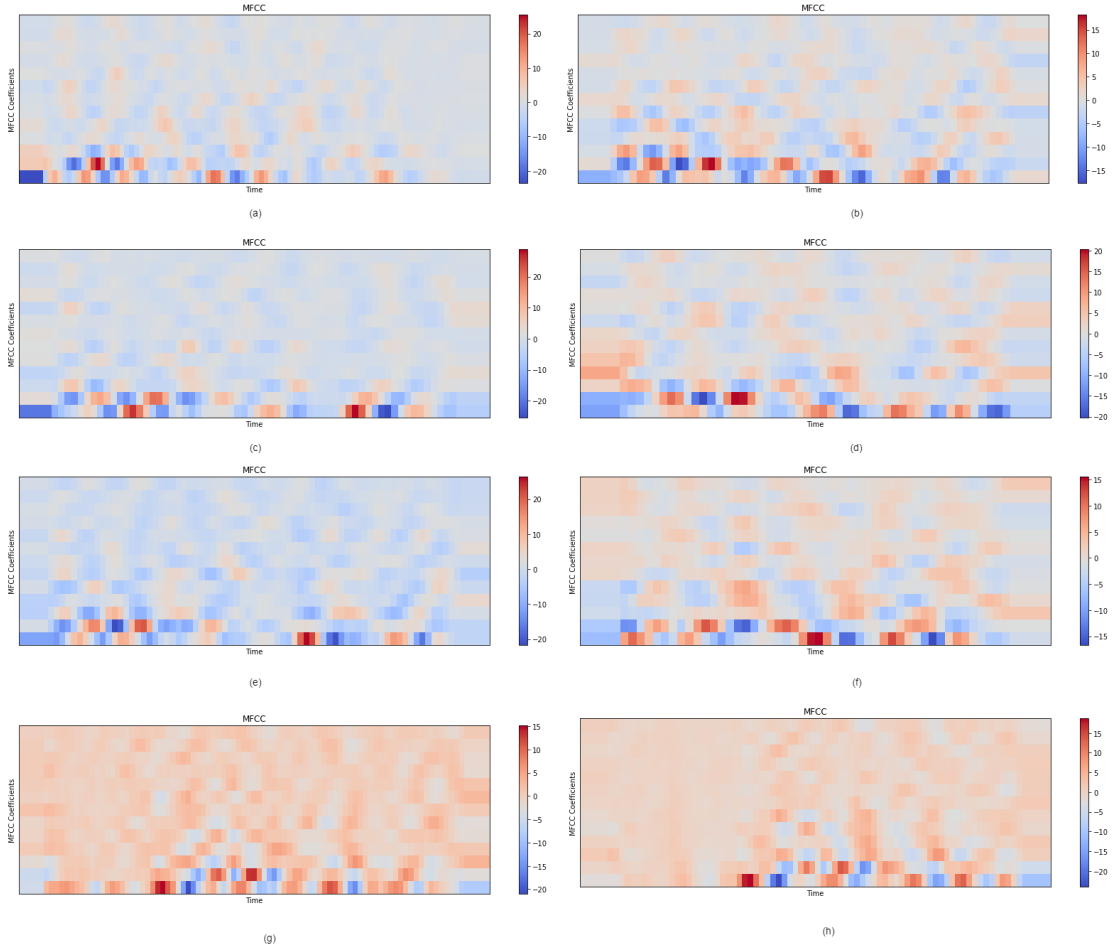

FIGURE 5.10: MFCC coefficients for different emotions.(a) Anger, (b) Calm, (c) Disgust, (d) Fearful, (e) Happy, (f) Sad, (g) Neutral, (h) Surprise

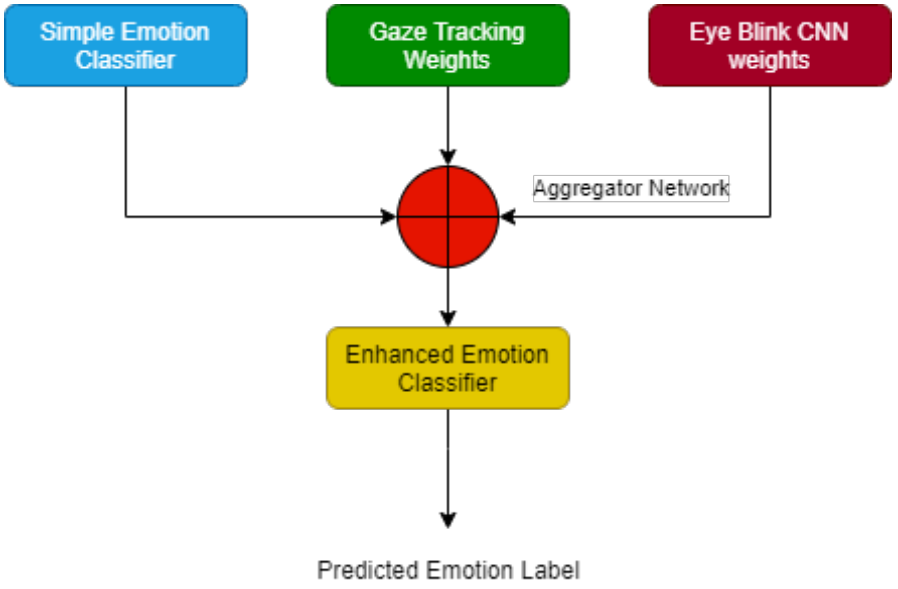

FIGURE 5.11: Enhanced Emotion Classifier

Table 5.2: Architecture of Convolutional Neural Network for SER

Layer	Kernel Size	Stride	Activation
conv1 (1D)	256	8	relu
conv2 (1D)	256	8	relu
BatchNorm1	-	-	-
conv3 (1D)	128	8	relu
conv4 (1D)	128	8	relu
conv5 (1D)	128	8	relu
conv6 (1D)	128	8	relu
BatchNorm2	-	-	-
conv7 (1D)	64	8	relu
conv8 (1D)	64	8	relu
FC1	4096	-	-
FC2	4096	-	-
FC2	1024	-	-
FC3	16	-	softmax

5.7 Architecture Design of Multimodal Network for Emotion Classification

The requirements and resulting architectures described in this thesis were set up to analyze the emotions of an interviewee. Further details of the platform are explained in what follows.

MobileNetV2 and SSDlite were used to detect the face exposing the analyzed emotion its output being submitted to a classifier in order to predict the emotion type.

The Facial landmarks CNN was used to define the face mesh comprising of 68 landmarks.

The Eye regions were localized which were then fed as an input to the Eye blink detector and the Iris Tracking module. The Facial emotion aggregator network took the the output weights from the emotion classifier, eye blink detector and the gaze tracking module to give the final output for the first stage. The accuracy obtained from first stage was 82.1 %.

In the second stage, the audio from the interviewee was taken as an input and fed to the CNN of the speech emotion classifier which classifies the audio on the basis

of gender and emotion. The accuracy rate obtained from second stage was 89.42 %.

Algorithm 5.2 Algorithm for Automated Emotion Analyzer of Interviews using Multimodal CNNs

while $face = True$ **do**
    crop face
    landmark detection using XceptionNet
    localize eye landmarks,
    estimating the gaze vector using CNN,
    predicting the total eye bpm using the blink detector
    if *gaze vector weights & bpm weight* $= True$ **then**
        predicting using the enhanced emotion classifier
        return face emotion label
    end if
    if *speech input* $= True$ **then**
        change *sampling rate* to 44100 and $offset = 0.5$
        trim the signal to $t = 2.5s$
        compute $MFCC$ coefficients by limiting $n = 13$
        visualize the mel spectogram
        return speech emotion label
    end if
        return predicted emotion
end while

The algorithm described above is the final stage of the implemented architecture. A special classifier is used here to classify the emotions.

Final output weights from first and second stages are fed into the multimodal network such that to it will then predict the final output which is one of the seven universal emotions. The accuracy of the final predicted emotion ranged from 87.47 % to 91.47 %. The simulation results for the experiment are presented in Chapter 7.

Algorithm 5.3 Algorithm for SER

while *speechinput* $= True$ **do**
    change *sampling rate* to 44100 and $offset = 0.5$
    trim the signal to $t = 2.5s$
    compute $MFCC$ coefficients by limiting $n = 13$
    visualize the mel spectogram
    predict using the trained model
        return speech emotion label
end while

The following evaluation metrics were used to measure to calculate the overall performance of our devised architecture. The training and testing dataset were split in a ratio of 80:20. The performance metric parameters t_p, t_n, f_p, f_n represent true positive, true negative, false positive and false negative results respectively.

Accuracy is given by the the ratio of the sum of t_p and t_n amongst entire data in the test set.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{5.10}$$

Precision is defined as the fraction of t_p among the real positive elements which can be defined as.

$$Precision = \frac{tp}{tp + fp} \tag{5.11}$$

Recall is the total true positives divided by the relevant elements defined as

$$Recall = \frac{tp}{tp + fn} \tag{5.12}$$

F1 score is a measure which is dependent on the precision and recall of the model

$$F_1 score = 2 * \frac{prec * recall}{prec + recall} \tag{5.13}$$

5.8 Validation

The validation of the multimodal emotion classifier was done using the Berkeley Expressivity Questionnaire [120] [121] [122] [123] which assesses the emotion based on 3 scales positive, negative, impulse. The interviewee is to be asked a series of questions using the former expressivity questionnaire. The scale is divided in three facets known as: Negative Expressivity, Positive Expressivity, and Impulse Strength.The model predicted emotion vs the questionnaire predicted emotion were calculated with an accuracy of 88 $\pm$3 %.

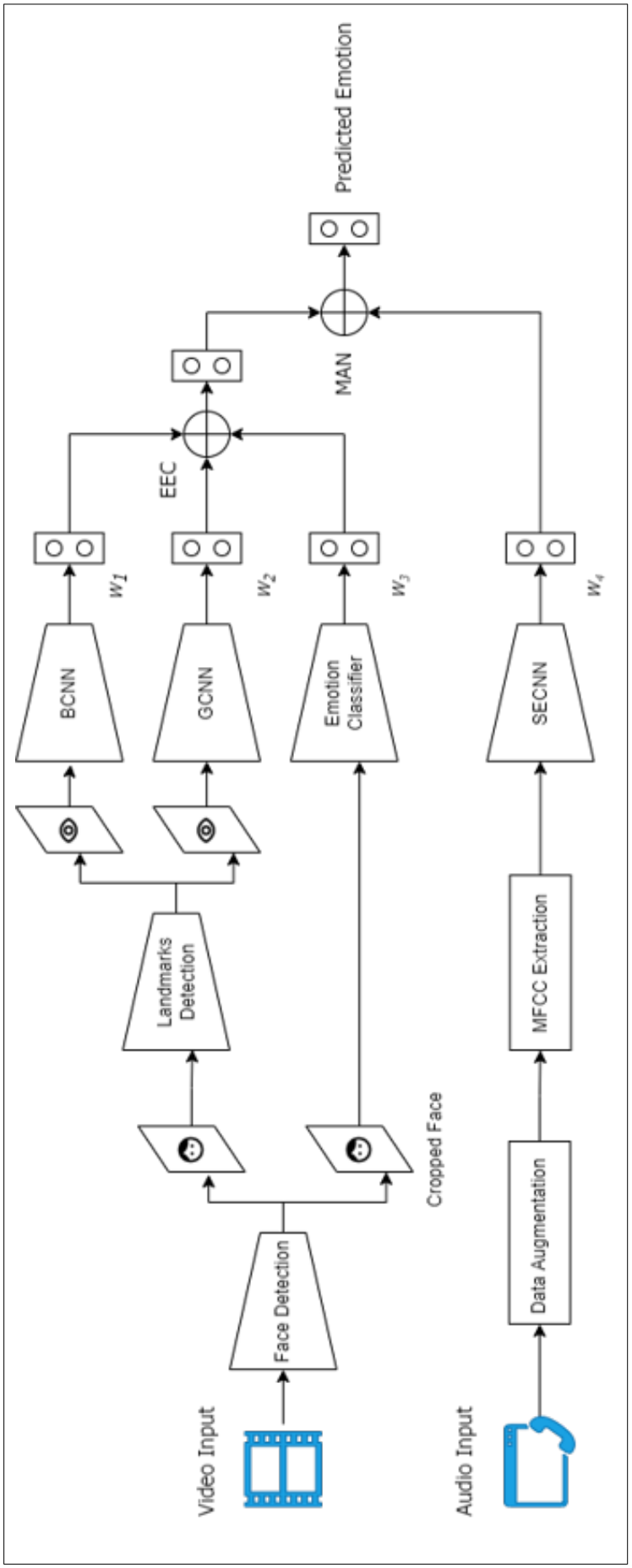

FIGURE 5.12: Architecture of the Multimodal Emotion Analyzer

5.8.1 Berkeley Expressivity Questionnaire

Each question has a 7 point Likert-type ranging from 1 (strongly disagree) to 7 (strongly agree) [123]. Table 5.3 shows the 7 point Likert scale followed by the expressivity questionnaire.

1. Whenever I feel positive emotions, people can easily see exactly what I am feeling.
2. I sometimes cry during sad movies.
3. People often do not know what I am feeling.
4. I laugh out loud when someone tells me a joke that I think is funny.
5. It is difficult for me to hide my fear.
6. When I'm happy, my feelings show.
7. My body reacts very strongly to emotional situations.
8. I've learned it is better to suppress my anger than to show it
9. No matter how nervous or upset I am, I tend to keep a calm exterior.
10. I am an emotionally expressive person.
11. I have strong emotions.
12. I am sometimes unable to hide my feelings, even though I would like to.
13. Whenever I feel negative emotions, people can easily see exactly what I am feeling.
14. There have been times when I have not been able to stop crying even though I tried to stop.
15. I experience my emotions very strongly.
16. What I'm feeling is written all over my face.

5.8.1.1 Scoring

Scoring is done based on the responses in which items 3, 8, 9 are reverse scored. Items 3, 5, 8, 9, 13, 16 make up the Negative Emotionality scale and Items 1, 4, 6, 10 constitute the Positive Emotionality scale and Items 2, 7, 11, 12, 14, 15 comprise up the Impulse Strength scale. An overall emotional response was calculated based on the scores from these 3 scales.

TABLE 5.3: 7 Point Rating Scale for BEQ

1	**2**	**3**	**4**	**5**	**6**	**7**
Strongly Disagree			**Neutral**			**Strongly Agree**

5.9 Summary

In this chapter, a deep learning based multimodal fusion network comprising of facial micro-expressions and speech synthesis has been developed. A new architecture comprising MAN was developed. This architecture is capable to deploy automatic emotion recognition methods to different use cases such as analysis of a person during interview, proctoring students during online examinations etc. The multimodal aggregator network uses all the single modal features from multiple CNN's and maps them to a unified space before the emotion classification. The use of synthetic micro-expressions developed using Generative Adversarial Networks (GANs) [72] [124] for detection and classification and using image to avatars animation algorithm for emotion classification is discussed in the next Chapter 6.

Chapter 6

Image Animation based Emotion Classification

This chapter extends the research pursued for a stable Emotion Classifier in Chapter 4 and Enhanced Emotion Classifier in Chapter 5. A real time system for emotion classifier was devised in this Chapter which is an improvement in both accuracy and speed from the formerly devised methods in the domain literature. The architecture of the new model implemented and algorithms used to denoise the ME databases are discussed in this chapter.

6.1 Functional Requirements

A series of functional requirements capturing the functionality listed below, were used for devising, developing and implementing an image-avatar animation based emotion classifier network.

1. The image-avatar animation model shall be comprised of different CNNs and GAN as a backbone network such as;
    1.1. SSDLite coupled with MobileNetV2 for face detection
    1.2. ResNet50 for emotion classification
    1.3. Pre-Trained GAN network for first order model motion
2. The architecture shall comprise an image-avatar generation network;

3. The pre trained GAN network shall create the animation model from the video stream;
4. The denoised emotion classifer shall use the Non-Local Means algorithm on the training dataset;
5. The denoised emotion classifier shall compose of a single CNN network while prediction;
6. No dedicated memory shall be needed for the video stream as the avatar generation is done in real-time;
7. The image-avatar animation model shall use a sample image from the test dataset for the generation of the avatar motion;
8. ResNet50 shall be used as a backbone network for training the emotion classifier;
9. The emotion classifier shall be trained on an expression rigged database;
10. The output of the image-animation based emotion classifier shall be the predicted emotion;

6.2 Denoised Emotion Classifier

The CNN based classifier introduced in this approach uses a state of the art dataset known as RAF-DB, achieving a high accuracy rate for the emotion detection and identification. Nonetheless due to the noise concentration in the images of the dataset the classifier can't reach the desired accuracy which yields to poor inference results.

To improve the accuracy of the emotion identification an image denoising - a technique widely used in the field of digital image processing is adopted for cleaning the dataset. Instead of applying the image denoising as a postprocessing technique, a pre-processing technique was used. In this thesis, a Non-Local Means (NLM) algorithm for denoising the dataset is proposed. NLM is widely used in medical imagery by the Magnetic Resonance Imaging (MRI) which assumes that the images contain excessive redundancy which can be further exploited to eradicate the

noise in the image i.e, pixels with similar neighbourhood regions can be used to determine the final denoised value of the pixel [125] [126].

$$U(i) = X(i) + N(i) \tag{6.1}$$

Where $X(i)$ is the original image, $N(i)$ is the Gaussian white noise with a mean μ and a variance of σ^2, and $U(i)$ is the image on which NLM is used.

$$U = \{U(i) \mid i \in I\} \tag{6.2}$$

Here i represents the field for all the pixels in the specified image domain on which NLM is being applied by weighting out the averages of all the pixels in the noise image. It can be represented by the equation [127]

$$NL[U](i) = \sum_{j \in I} w(i,j)u(i) \tag{6.3}$$

The value of $w(i,j)$ depends on i and j as

$$0 \leq w(i,j) \leq 1 \tag{6.4}$$

$$\sum_{j} w(i_2,j) = 1 \tag{6.5}$$

The below equation represents the L^2 norm of $d(i,j)$ calculated between the neighborhood matrices N_i and N_j, α is the standard deviation for the weighting W, where N_i denotes a square neighborhood of fixed size N×N centered at pixel i, $u(N_i)$ denotes the intensity gray values of u at N_i [128].

$$d(i,j) = \|u(N_i) - u(N_j)\|^2_{2,\alpha} \tag{6.6}$$

The Gaussian weighting coefficient w_i is then reformed into a discretized format known as the Gaussian weighting coefficient template. This shows that while

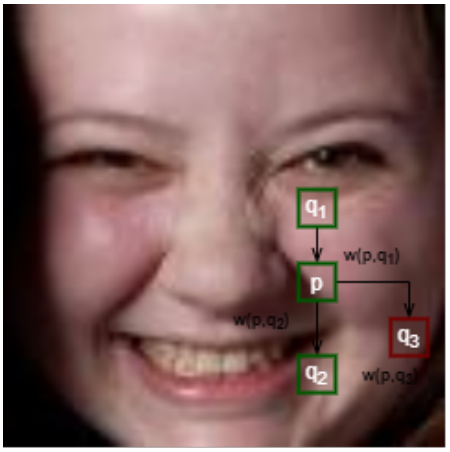

FIGURE 6.1: NL-Means to Denoise an Image

measuring the neighborhood distance, the center of the matrix is associated with a greater weight value than the pixels away from the center [128].

$$w(i,j) = \frac{1}{c(i)} f_k(d(i,j)) \tag{6.7}$$

$$f_k = \exp(-\frac{d(i,j)}{h^2}) \tag{6.8}$$

$$c(i) = \sum_{j \in I} \exp(-\frac{d(i,j)}{h^2}) \tag{6.9}$$

where $c(i)$ is the normalization factor, h being the attenuation factor. Fig 6.1 depicts the Non Local Means algorithm to denoise the image. Similar pixel neighbourhoods give a large weight $w(p, q_1)$, $w(p, q_2)$ while different neighborhoods give smaller weight $w(p, q_3)$.

The base network to train the denoised emotion classifier was VGG-16. Table 6.1 shows the performance metrics of the emotion classifier after NL-Mean was applied on the training dataset. Section 6.3 shows futher improvement in detecting emotions

Algorithm 6.1 Algorithm for Denoising the Emotion Classifier using NL-Mean

while $face = True$ **do**
    crop face
    align the training set only.
    use an odd value for the template patch (t), $\in [3, 5, 7, 11, 13]$
    choose an optimal value of $h < 20$, Higher values of h can reduce noise but won't preserve image details. Here we used $h = 10$.
    set the value of h_{color}. This parameter removes the colored noise. choose an optimal value of $h_{color} < 20$. Here we used $h_{color} = 10$.
    set the $searchWindowSize = 21$
    if $searchWindowSize > 30$ **then**
        Stop as computation time will increase.
    end if
    return denoised image training dataset
end while

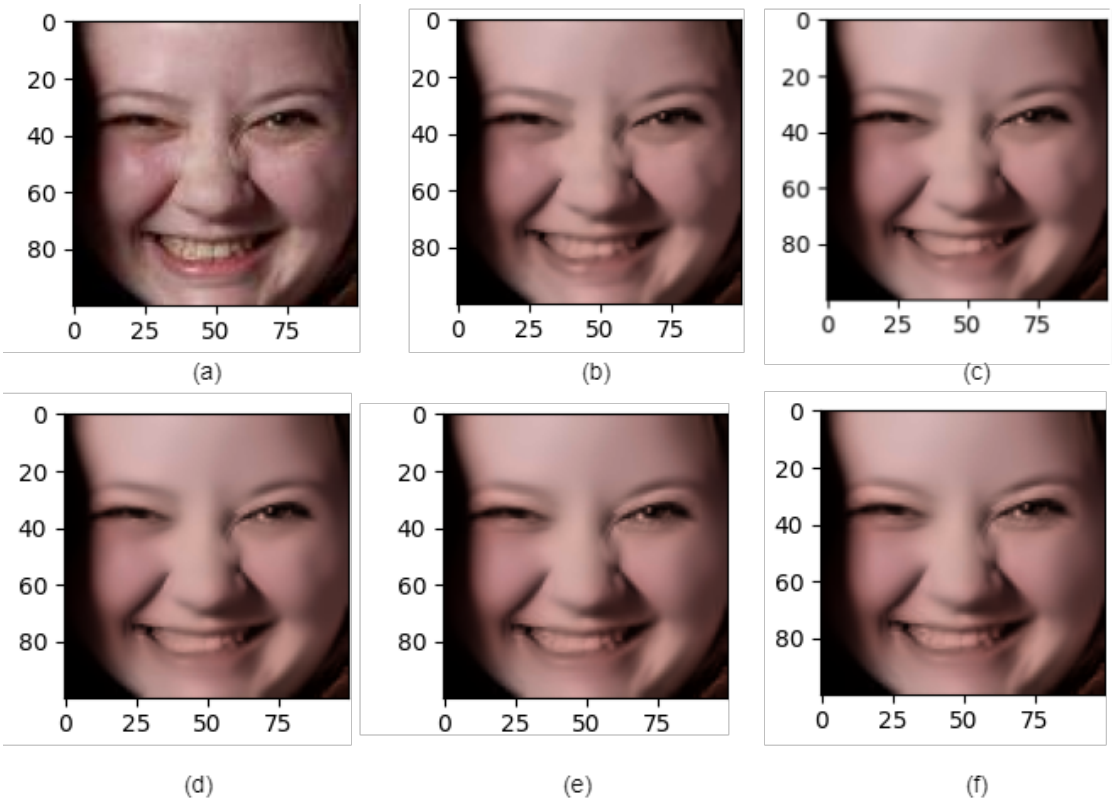

FIGURE 6.2: Different smoothed images based on varying template patch values in pixels which are used to compute weights.(a) shows the original image of the training dataset, (b) shows the applied NL-Means on the image with a value of template patch as 3, (c) shows the applied NL-Means on the image with a value of template patch as 5, (d) shows the applied NL-Means on the image with a value of template patch as 7, (e) shows the applied NL-Means on the image with a value of template patch as 11, (f) shows the applied NL-Means on the image with a value of template patch as 13

TABLE 6.1: Performance Accuracy for the Denoised Emotion Classifier

Template Patch Value	Accuracy
0	78%
3	**85.2%**
5	83.6%
7	81%
11	80.08%
13	80%

6.3 Image Animation based Emotion Classifier

As seen in the above Section 6.2, emotion classifier can be biased due to the noise generated by the training samples of the emotion images. This applies to the testing of the Emotion Classifier built using ResNet50 as well. To overcome this noise from the testing side, the following architecture Figure 6.8 was devised to attain the desired accuracy based on image animation [129].

An optical flow can be defined as the motion of the image produced by an object which moves between two consecutive frame. There are numerous methods to compute optical flows. One of them is the brightness constancy constraint which assumes that the pixel intensities occurred in between the frames, is constant.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{6.10}$$

where $I(x, y, t)$ is the image, Δx, Δy, Δt represent the rate of change of the pixels of that image, and t is the time on which when applying a Taylor expansion it can be inferred that eq. 6.11 holds with V_x and V_y being the velocities of the optical flow obtained, as given in eq. 6.12.

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \tag{6.11}$$

$$\nabla \boldsymbol{I} \cdot \overrightarrow{\mathbf{V}} = -I_t \tag{6.12}$$

where ∇I is the image gradient, which relates to the point where there is a maximum change in the intensity or the color of the image, while $\overrightarrow{V}$ is the vector of the velocity of the optical flow defined on the flow of images. Image animation

is a process in which a video sequence output is generated with specific motions extracted from a video stream (dynamic stream) and objects from still images (static). Fig. 6.4 [129] shows the process of image animation using the first order model algorithm. The Source image $\mathbf{S} \in \mathbb{R}$ is the image which is being animated. The driving frame $\mathbf{D} \in \mathbb{R}$ is usually taken from a similar video sequence containing the objects and motions of interests.

The motion module consists of two major blocks: a keypoint detector block which is an unsupervised detector for predicting the keypoints using the weights from source frame ($\mathbf{S}$) and the driving frame ($\mathbf{D}$) and a Local Affine Transformations block where it estimates the backward optical flow [129]. The equations which depict these affine transformations are shown in eq. 6.17 to eq. 6.29.

The homogeneous transformation matrix from the frame of reference of $\mathbf{D}$ to $\mathbf{S}$ i.e., $T_{\mathbf{S}\leftarrow\mathbf{D}}$ can be defined using eq. 6.13 for the frame $\mathbf{D}$ w.r.t frame $\mathbf{S}$ with a translation vector $d \in [x_t, y_t, 1]$ w.r.t the origin of frame $\mathbf{D}$ from frame $\mathbf{S}$ in the co-ordinates of frame $\mathbf{S}$. The height and width have the normalized coordinates such that $-1 \leq x, y \leq 1$. The homogeneous transformation matrix is a 3x3 matrix whose elements are the translation and rotation parameters.

$$T = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6.13}$$

The dense motion field in these images is computed using the optical flow which is modeled by the function $T_{\mathbf{S}\leftarrow\mathbf{D}}$. This function maps each and every pixel located in $\mathbf{D}$ to its corresponding location in $\mathbf{S}$. In forward optical flow, the iteration is done over the original source image using the default values but is been mapped to a new location in the new image, whereas in the backward optical flow the behaviour is opposite i.e, iteration is done over the new image instead of the source image and a bilinear interpolation is used to compute the value of these pixels. Interpolation can be defined as the way to estimate new data points within a given range usually applied to transform images by reducing its dimensions.

Assuming that there exists an abstract reference frame $\mathbf{R}$ using which the transformations from $\mathbf{R}$ to $\mathbf{S}$, $\mathbf{R}$ to $\mathbf{D}$ ($T_{\mathbf{S}\leftarrow\mathbf{R}}$, $T_{\mathbf{D}\leftarrow\mathbf{R}}$) can be estimated. The output of the keypoint detector network consists of the parameters of affine transformations which are being modeled around each keypoint.

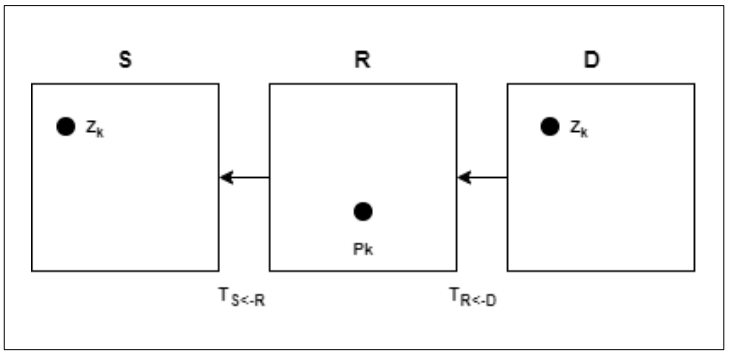

FIGURE 6.3: Position of a Keypoint using the reference frame in Transformation Matrix

To compute the Local Affine Transformations, the first order Taylor expansion is used to represent $T_{\mathbf{S}\leftarrow\mathbf{D}}$ in k points. The frame $\mathbf{X}$ in eq. 6.14 is a random frame which is used for interpreting the equations and is replaced by either $\mathbf{S}$, $\mathbf{D}$.

$$T_{\mathbf{X}\leftarrow\mathbf{R}}(p) = T_{\mathbf{X}\leftarrow\mathbf{R}}(p_k) + \left(\frac{d}{dp}T_{\mathbf{X}\leftarrow\mathbf{R}}(p)\middle|p = p_k\right)(p - p_k) + o(||p - p_k||) \quad (6.14)$$

where, p_1,p_2,.. p_k represent the coordinates of the keypoints given in the reference frame $\mathbf{R}$. The last term little-o (o) shows the approximated Taylor expansion around $(p - p_k)$. Since, the transformation function is a multi-variable function depending on 2 variables the derivative of this is composed of a matrix of gradients, also known as the Jacobian matrix.

The keypoint detector network is based on the U-Net architecture [130] which adds the convolutional layer consisting of 10 filters to the output of the U-Net Model. The 10 filters each represent a keypoint, i.e., the output of this convolutional layer is a feature map of size (BxHxWx10) where B represents the batch size, HxW represent the dimensions of the frame. The location of the keypoints included in the feature map can be obtained by computing the maximum value of each of the matrices.

To compute the Jacobians from the image using the keypoint detector, a convolutional layer comprising of 40 filters (for 4 variables of the Jacobian) is fed to the output of the U-Net model which is initialized by the weights and biases to follow the transformation whose output is the original matrix (identity transformation). The final jacobian variable matrix has a shape of (10x2x2). The final output for the affine transformations is the spatial weighted average is computed using the keypoints weight's confidence map. The motion function $T_{\mathbf{X}\leftarrow\mathbf{R}}$ is denoted by the

value generated from each keypoint. The Jacobians computed at each of those coordinates of the keypoints ranging from p_1,p_2,.. p_k are given by [129]

$$T_{\mathbf{X}\leftarrow\mathbf{R}}(p) \approx \left\{\left\{T_{\mathbf{X}\leftarrow\mathbf{R}}(p_1), \frac{d}{dp}T_{\mathbf{X}\leftarrow\mathbf{R}}(p)\middle|p=p_1\right\}, ...\left\{T_{\mathbf{X}\leftarrow\mathbf{R}}(p_k), \frac{d}{dp}T_{\mathbf{X}\leftarrow\mathbf{R}}(p)\middle|p=p_k\right\}\right\} \tag{6.15}$$

Now, $T_{\mathbf{S}\leftarrow\mathbf{D}}$ can be obtained by using the following equation

$$T_{\mathbf{S}\leftarrow\mathbf{R}}(p_k) = \begin{bmatrix} a_1 & b_1 & p_x \\ c_1 & d_1 & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = x.\mathbf{u}_1 + y.\mathbf{v}_1 + \mathbf{p} \tag{6.16}$$

$$T_{\mathbf{D}\leftarrow\mathbf{R}}(p_k) = \begin{bmatrix} a_2 & b_2 & p_x \\ c_2 & d_2 & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = x.\mathbf{u}_2 + y.\mathbf{v}_2 + \mathbf{p} \tag{6.17}$$

$$T_{\mathbf{D}\leftarrow\mathbf{R}}(p_k)^{-1} = \frac{1}{a_2d_2 - b_2c_2} \begin{bmatrix} d_2 & -b_2 & -p_xd_2 + p_yb_2 \\ -c_2 & a_2 & p_xc_2 + p_ya_2 \\ 0 & 0 & a_2d_2 - b_2c_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = x.\mathbf{u}_2' + y.\mathbf{v}_2' + \mathbf{p}' \tag{6.18}$$

where $\begin{pmatrix} a_1 & c_1 & 0 \end{pmatrix}^\top$, $\begin{pmatrix} b_1 & d_1 & 0 \end{pmatrix}^\top$, $\begin{pmatrix} a_2 & c_2 & 0 \end{pmatrix}^\top$, $\begin{pmatrix} b_2 & d_2 & 0 \end{pmatrix}^\top$, $\begin{pmatrix} -d_2 & -c_2 & 0 \end{pmatrix}^\top$, $\begin{pmatrix} -b_2 & a_2 & 0 \end{pmatrix}^\top$ are vectors and $\begin{pmatrix} p_x & p_y & 1 \end{pmatrix}^\top$, $\begin{pmatrix} x & y & 1 \end{pmatrix}^\top$, $\begin{pmatrix} p_x' & p_y' & c \end{pmatrix}^\top$ represent points in the domain. Using the properties of homogeneous transformation matrix $T_{\mathbf{A}\leftarrow\mathbf{B}} = T_{\mathbf{B}\leftarrow\mathbf{A}}^{-1}$

$$T_{\mathbf{S}\leftarrow\mathbf{D}} = T_{\mathbf{S}\leftarrow\mathbf{R}} \circ T_{\mathbf{R}\leftarrow\mathbf{D}} = T_{\mathbf{S}\leftarrow\mathbf{R}} \circ T_{\mathbf{D}\leftarrow\mathbf{R}}^{-1} \tag{6.19}$$

$$T_{\mathbf{S}\leftarrow\mathbf{D}} \approx T_{\mathbf{S}\leftarrow\mathbf{R}}(p_k) + J_k(z - T_{\mathbf{D}\leftarrow\mathbf{R}}(p_k)) \tag{6.20}$$

where J_k is the Jacobian defined as

$$J_k = \left(\frac{d}{dp}T_{\mathbf{S}\leftarrow\mathbf{R}}(p)\middle|p=p_k\right)\left(\frac{d}{dp}T_{\mathbf{D}\leftarrow\mathbf{R}}(p)\middle|p=p_k\right)^{-1} \tag{6.21}$$

Substituting eq. 6.16 and eq. 6.17 in eq. 6.21 we get J_k as

$$J_k = \frac{d}{dp}(x.\mathbf{u}_1 + y.\mathbf{v}_1 + \mathbf{p})\frac{d}{dp}(x.\mathbf{u}_2' + y.\mathbf{v}_2' + \mathbf{p}') \tag{6.22}$$

$$z - T_{\mathbf{D}\leftarrow\mathbf{R}}(p_k) = x\mathbf{u}_2 + y\mathbf{v}_2 + (\mathbf{z} - \mathbf{p}) \tag{6.23}$$

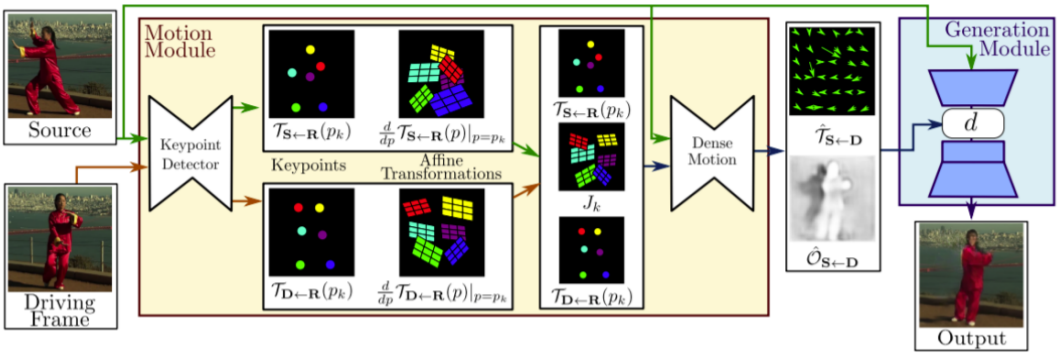

FIGURE 6.4: Architecture of Image Animation using First order Model

Assuming the value of eq. 6.22 as c which is the absolute value of the jacobian

$$J_k(z - T_{\mathbf{D}\leftarrow\mathbf{R}}(p_k)) = c(x\mathbf{u}_2 + y\mathbf{v}_2 + (\mathbf{z} - \mathbf{p})) \tag{6.24}$$

$$J_k(z - T_{\mathbf{D}\leftarrow\mathbf{R}}(p_k)) = x\mathbf{u}_2'' + y\mathbf{v}_2'' + (\mathbf{z}' - \mathbf{p}') \tag{6.25}$$

Substituting the value of eq. 6.25 in eq. 6.20

$$T_{\mathbf{S}\leftarrow\mathbf{D}}(z) \approx x.\mathbf{u}_1 + y.\mathbf{v}_1 + \mathbf{p} + x\mathbf{u}_2'' + y\mathbf{v}_2'' + (\mathbf{z}' - \mathbf{p}') \tag{6.26}$$

$$T_{\mathbf{S}\leftarrow\mathbf{D}}(z) \approx x.(\mathbf{u}_1 + \mathbf{u}_2'') + y.(\mathbf{v}_1 + \mathbf{v}_2'') + (\mathbf{z}' - \mathbf{p}' + \mathbf{p}) \tag{6.27}$$

Using eq. 6.27 and substituting $(\mathbf{u}_1 + \mathbf{u}_2'')$=$\mathbf{u}_3$, $(\mathbf{v}_1 + \mathbf{v}_2'')$=$\mathbf{v}_3$ and $(\mathbf{z}'$- $\mathbf{p}'$+$\mathbf{p}) = \mathbf{z}''$

$$T_{\mathbf{S}\leftarrow\mathbf{D}}(z) \approx x.\mathbf{u}_3 + y.\mathbf{v}_3 + \mathbf{z}'' = \begin{bmatrix} a_3 & b_3 & z_x'' \\ c_3 & d_3 & z_y'' \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{6.28}$$

where similar to eq. 6.16, eq. 6.17

$$\mathbf{u}_3 = \begin{bmatrix} a_3 & c_3 & 0 \end{bmatrix}^\top, \mathbf{v}_3 = \begin{bmatrix} b_3 & d_3 & 0 \end{bmatrix}^\top, \mathbf{z}'' = \begin{bmatrix} z_x'' & z_y'' & 0 \end{bmatrix}^\top \tag{6.29}$$

eq. 6.16 represents the transformation from **D** to **S** in z.

The partial derivatives of the Jacobian matrix describe how the neighborhood of the x and y is being transformed using the transformation matrix which further allows to get the local affine transformations. From eq. 6.19, it can be inferred that the transformation $T_{\mathbf{S}\leftarrow\mathbf{R}}$ is locally bijective in the neighborhood.

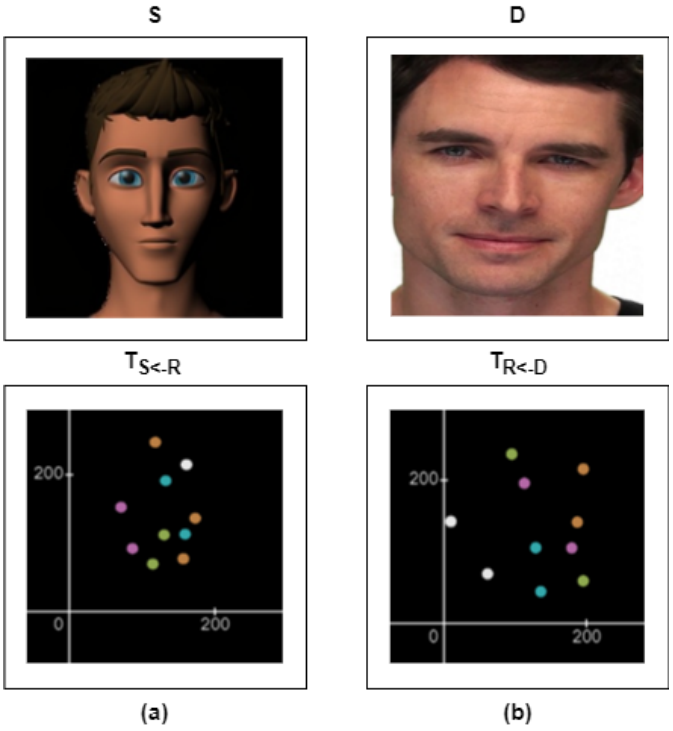

FIGURE 6.5: (a) shows the transformation matrix $T_{\mathbf{S}\leftarrow\mathbf{R}}$,(b) shows the transformation matrix $T_{\mathbf{R}\leftarrow\mathbf{D}}$

The generator network as described in the Fig. 6.4 is based on the Johnson architecture for the fast style transfer [131] as shown in Fig 6.7. The output from the DCNN block is fed to the Johnson network as an input. The output of the Johnson network is the static image being transformed into a deep-fake.

6.3.1 Training

The dataset used here to train the emotion classifier was Facial Expression Research Group 2D Database (FERG-DB) [99], which consists of 55767 annotated face images of stylish cartoon characters, classified into seven universal emotion. Fig. 6.6 shows the sample image of the 2D styled expression rigged characters.

6.4 Architecture Design

Figure 6.8 shows the proposed architecture for the image animation based emotion classifier. Stage-1 comprises of the real time video being passed through the face detection module which detects the faces and gives the cropped face as an

Algorithm 6.2 Algorithm for Image Animation based Emotion Classifier

while $face = True$ **do**
    crop face
    passing the source and driving frame into keypoint detector.
    source frame transformation $T_{\mathbf{S}\leftarrow\mathbf{R}}(p_k)$.
    driving frame transformation $T_{\mathbf{D}\leftarrow\mathbf{R}}(p_k)$.
    computing affine transformations using $\frac{d}{dp}T_{\mathbf{S}\leftarrow\mathbf{R}}(p)$, $\frac{d}{dp}T_{\mathbf{D}\leftarrow\mathbf{R}}(p)\Big|p = p_k$.
    computing jacobian J_k.
    motion generation output using the optical flow $\hat{T}_{\mathbf{S}\leftarrow\mathbf{D}}(p_k)$ and occlusion map $\hat{O}_{\mathbf{S}\leftarrow\mathbf{D}}(p_k)$.
    feeding motion generation output into the emotion classifier. **return** predicted emotion
end while

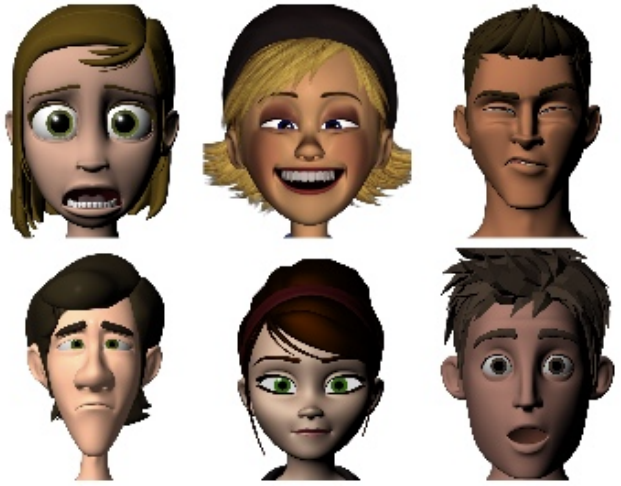

FIGURE 6.6: FERG Database Sample (Anger, Happy, Disgust, Fear, Neutral, Surprise)

output. The cropped face is fed as one of the inputs to the Image Animation module for Stage 2. Another input comprises of a driving frame from the FERG database i.e. an image with a neutral face. The keypoint detector detects the facial points, passing the source **S** and driving frame **D** to compute the local affine transformations as shown in the Algorithm 6.2. The final output is extracted via the generation module.

The final output from Stage 2 from the generation module is fed to the next stage which does the feature extraction and a final emotion is predicted. The source video is from a movie dataset called RAVDSS for its song and speech [86] content. The image animation is being performed in real time on the cropped face regions

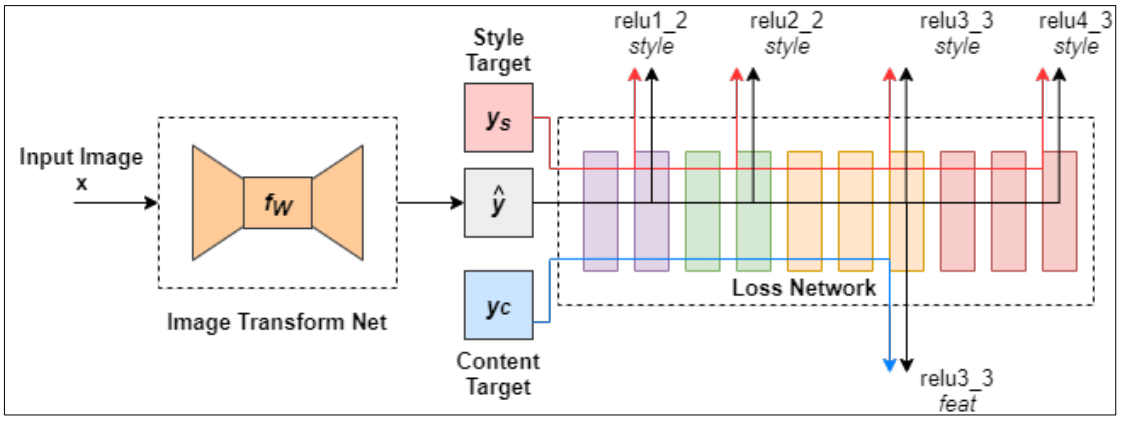

FIGURE 6.7: Architecture of the Johnson Network showing the perceptual loss function implementation

only using a neutral image from the training dataset as shown in 7 Fig. 7.11. The emotion classifier detects the emotion from this animated frame.

6.5 Summary

Previous architectures and algorithms devised in this thesis, discussed in Chapter 4 and Chapter 5 were computationally too complex to be deployed in real-time. In this chapter, a new architecture and a new algorithm involving a real-time image to avatar animation was developed. An emotion classifier network was applied to it and was trained on different well-adapted networks like ResNet50, InceptionV3 and VGG16. The new architecture and the dataset 6.8 outperformed all the previously implemented algorithms (EVM and Multimodal Algorithm) with a inference accuracy of 92.23% and a speed of 25 FPS. The next chapter will provide the simulations results of the algorithms used in Chapter 4, Chapter 5 and Chapter 6.

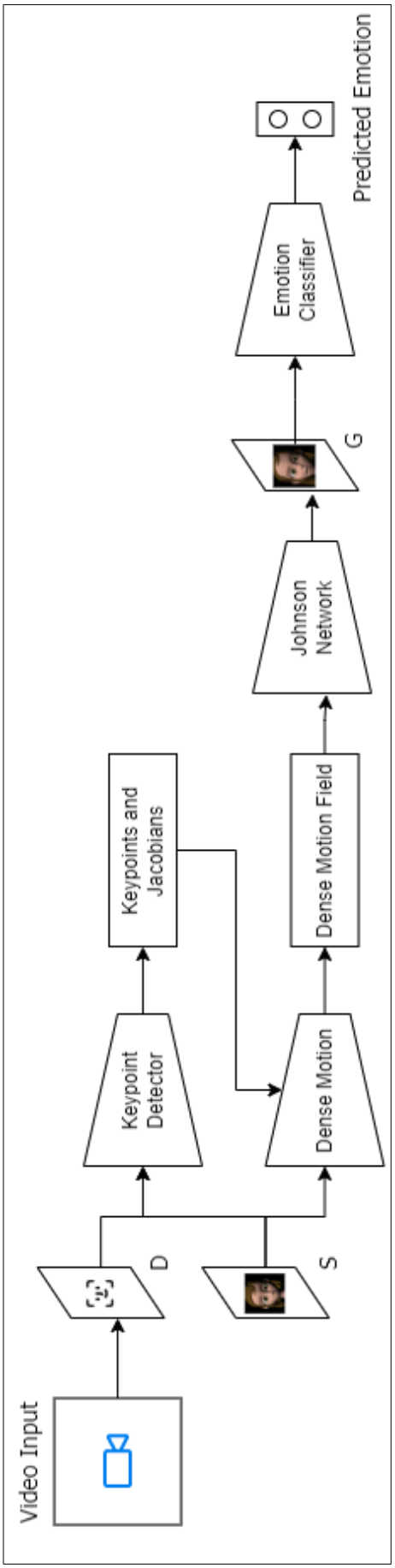

FIGURE 6.8: Architecture of Image Animation based Emotion Classifier

Chapter 7

Simulation Results

7.1 Experimental Results of EVM based Emotion Classifier

In this section, the results gathered from tests and performance obtained from the new architecture and algorithms devised and implemented in Chapter 4 to Chapter 6 including, are presented. A confusion matrix is the ideal choice for a performance metric while solving a classification problem. This provides the classification results in a matrix format which contains all the predicted and the wrongly predicted classes. The average of the primary diagonal gives the validation accuracy of the model. The diagonal elements of the confusion matrix represent the total true positives (t_p) and true negatives (t_n) of the classifier. The values of t_p and t_n are calculated using the equations depicted in eq. 5.10 Fig. 7.4 shows the results from the EVM based emotion classifier inferred on the RAVDSS Dataset.

TABLE 7.1: Performance Accuracy for basic emotions on RAF-DB

——	Accuracy
SCNN. [132]	60
DLP-CNN [88]	84.22
IPA2LT [133]	86.77
gaCNN [134]	85.07
EVM Emotion Classifier	**88.47**

Compared with the traditional models such as DLP-CNN, SCNN and gaCNN as shows in Table 7.1 and Table 7.2 to detect micro-expressions, this approach

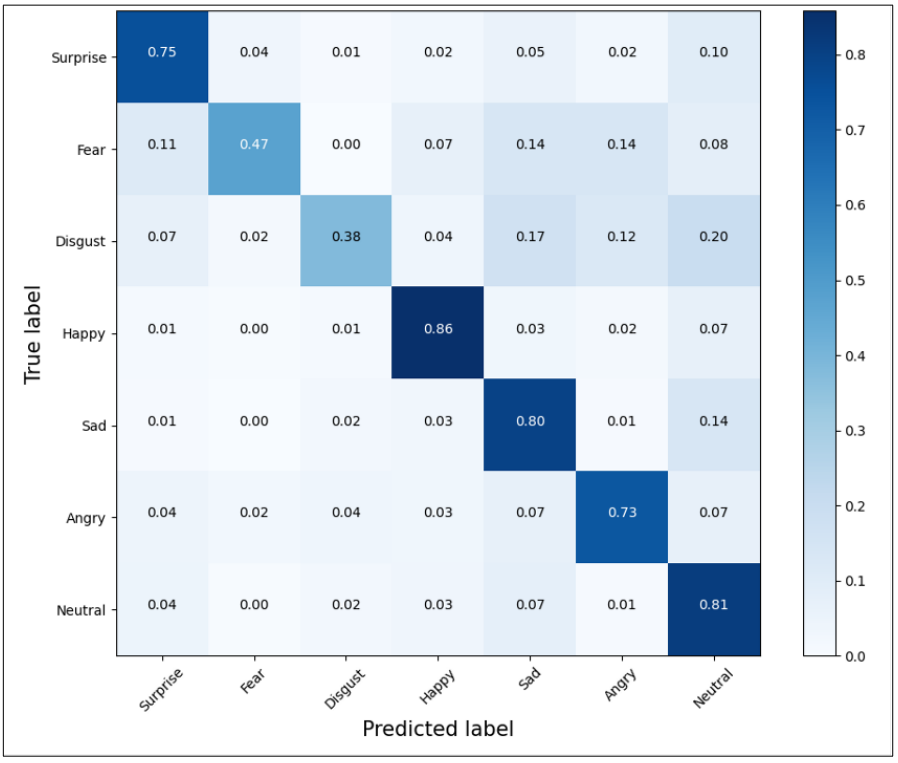

FIGURE 7.1: Confusion Matrix for Basic Emotions

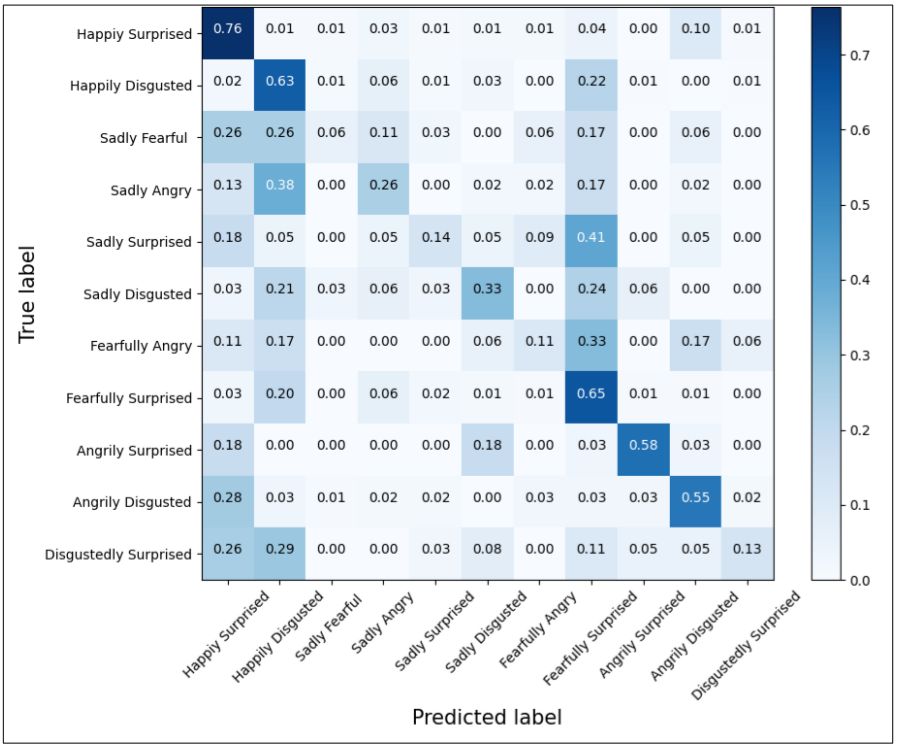

FIGURE 7.2: Confusion Matrix for Compound Emotions

TABLE 7.2: Performance Comparison for Compound emotions on RAF-DB.

——	Mean of Confusion Matrix
DLP-CNN [88]	32.29
Emotion Classifier	38.36
EVM Emotion Classifier	**42.61 ± 2**

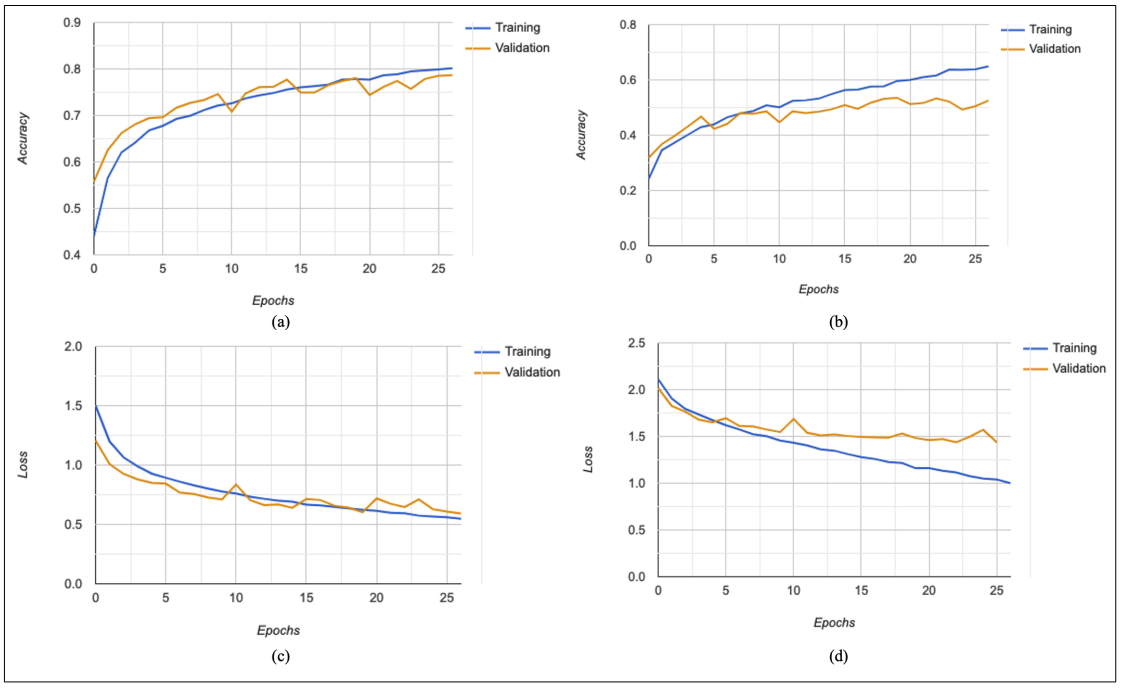

FIGURE 7.3: (a) and (c) show the accuracy and loss graphs of the emotion classifier when trained on basic emotions, (b) and (d) show the accuracy and loss graphs of the emotion classifier when trained on compound emotion.

surpasses them by achieving accuracy of 88.47%. Fig. 7.1 and Fig. 7.2 shows the confusion matrix for the CNN trained on RAF-DB for the emotion classification.

7.2 Experimental Results of Multimodal Network based Emotion Classifier

The accuracy and loss curves for the analyzer of the speech emotion recognition are depicted by Figure 7.6 and the accuracy and loss curves for both training and validation of the CNN's used for blink detection and gaze tracking are depicted in Fig. 7.5.

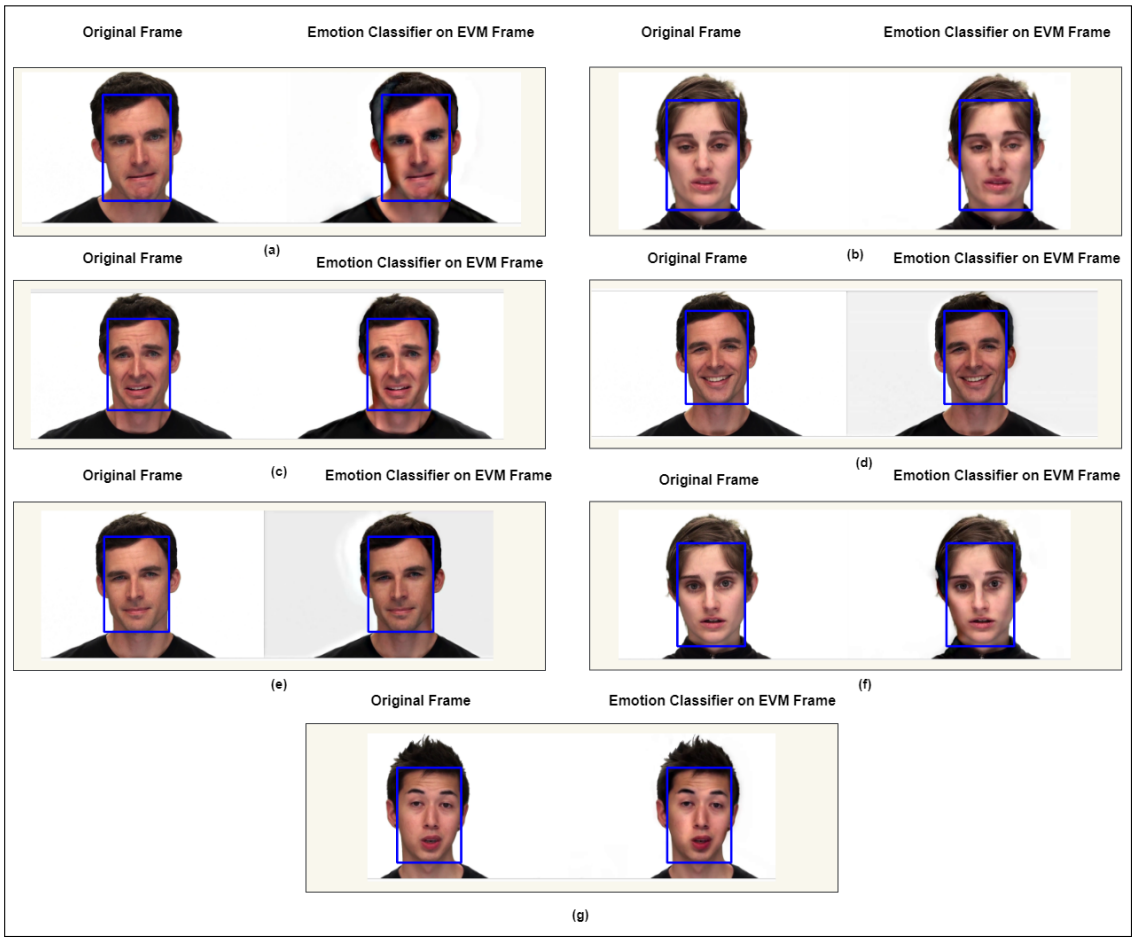

FIGURE 7.4: Results of the Emotion Classifier using Amplitude Eulerian Video Magnification.(a) shows the angry emotion, (b) shows the disgust emotion, (c) shows the fear emotion, (d) shows the happy emotion, (e) shows the neutral emotion, (f) shows the sad emotion and (g) shows the surprised emotion

Table 7.3 shows the validation accuracies of different convolutional neural network models used as the backbone for the multimodal aggregator network. The results generated by the speech synthesis module were obtained by using a custom CNN architecture as discussed in Section 5.2. Table 7.4 shows the precision, recall and the F1 score of the 7 classes in the speech emotion analyzer.

Angry Speech has the highest precision whilst disgust speech had the lowest, similar to the results obtained by the confusion matrix for the facial emotion classification. Fig. 7.7 shows the output from the CNN used for the blink detection and gaze tracking. The weights of these CNNs are carried forward to develop architecture of the multimodal network.

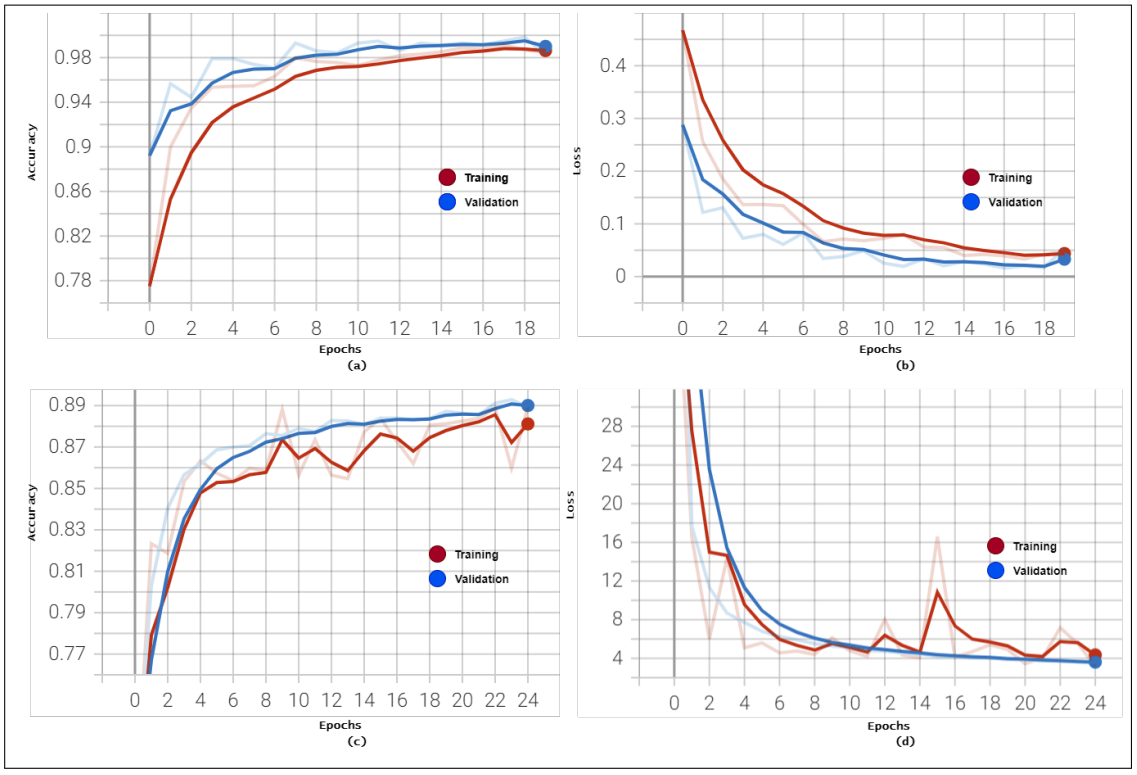

FIGURE 7.5: (a) and (b) show the Accuracy and Loss curves for the Eye State CNN Classifier. (c) and (d) show the Accuracy and Loss curves for the CNN based Gaze Tracking.

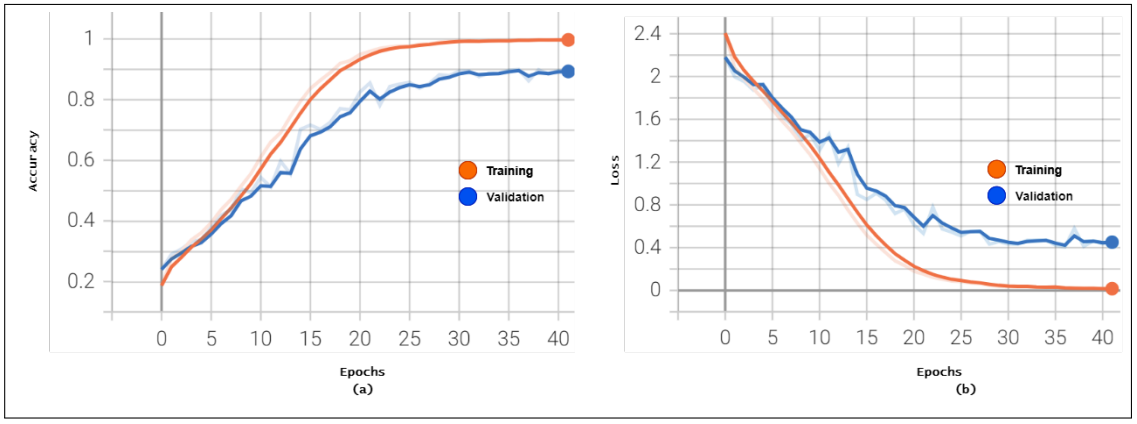

FIGURE 7.6: (a) and (b) show the Accuracy and Loss Curves for Speech Emotion Analyzer

TABLE 7.3: Performance Accuracies for the Multiple CNNs in the Multimodal Network Architecture

CNN Network Model	Accuracy	FPS
SSD Face Detector	95.2%	60
Facial Landmarks Mesh	98.6%	32
Eye-Blink Detector	99.2%	60
Gaze Tracking	89.9%	48
FER Aggregator	82.1%	-
Speech Synthesis	89.42%	-

TABLE 7.4: Performance Metrics for the Speech Emotion Analyzer

Emotion	Precision	Recall	F1-Score
Angry	0.93	0.93	0.93
Calm	0.90	0.85	0.87
Disgust	0.88	0.94	0.90
Fearful	0.92	0.88	0.89
Happy	0.89	0.91	0.90
Surprised	0.90	0.98	0.93
Sad	0.89	0.88	0.89

7.3 Experimental Results of Image Animation based Emotion Classifier

TABLE 7.5: Performance Accuracy for the Image Animation based Emotion Classifier

Backbone Network	Accuracy
VGG-16	93%
InceptionResnet	86.5%
InceptionV3	87.8%
Resnet50	**98.7**%

Transfer learning backbone networks such as VGG16, InceptionResNet, InceptionV3 and ResNet50 were used by the new architecture for emotion detection and identification in order to train the FERG-2D based emotion classifier. The results are tabulated as shown in Table 7.5. The performance accuracy is best when ResNet50 as the backbone network is used. ResNet50 not only enhances the validation accuracy but also being a small scale model, the inference speed satisfy real-time constraints as the model performs each frame in 0.04 s equivalent

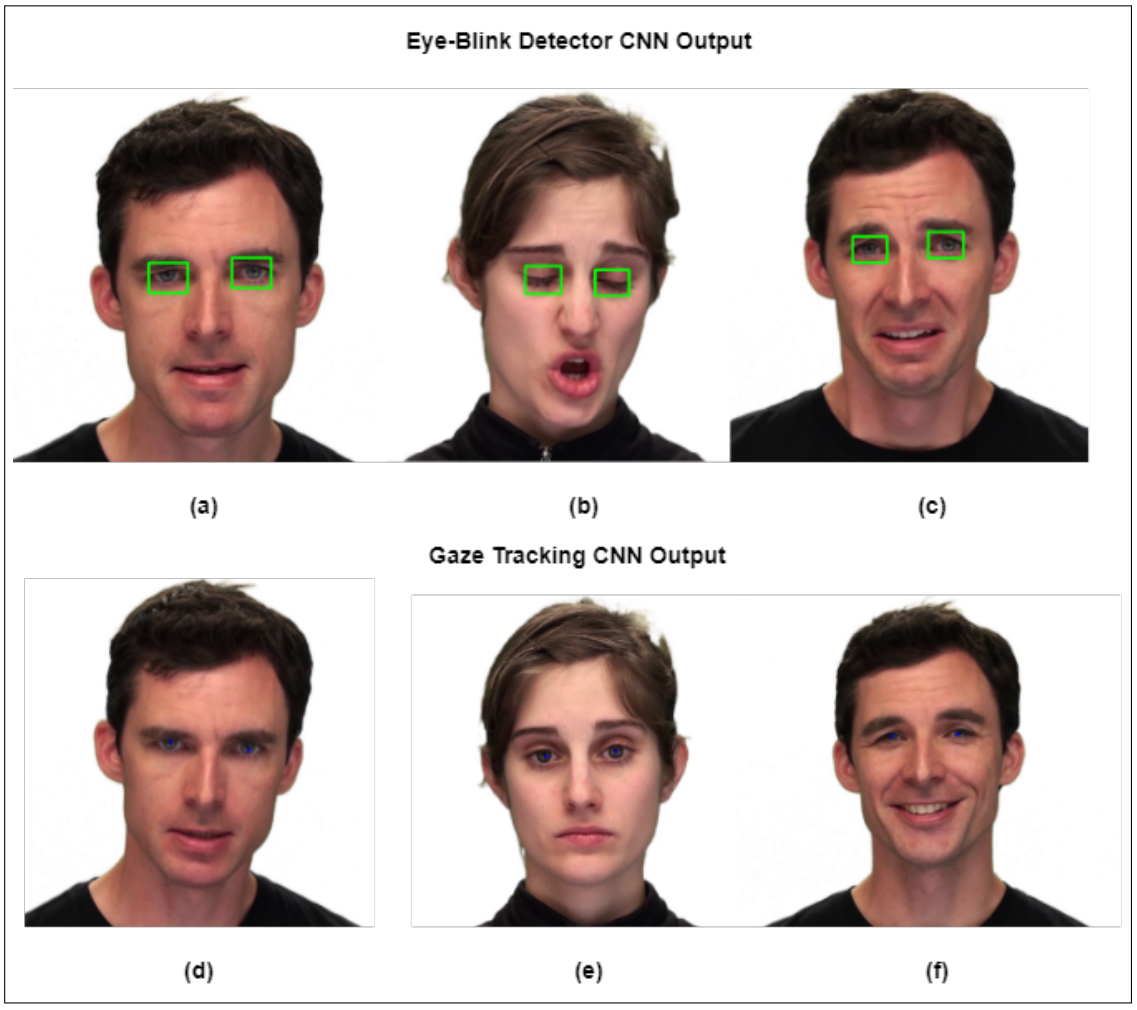

FIGURE 7.7: (a) and (c) show the image frames from the blink detector which predicts the eye state as open and (b) shows the eye state as closed. (d), (e) and (f) show the gaze vectors from the gaze tracking CNN.

to 25FPS. The training and validation curves of the different transfer learning methods used are depicted in Fig. 7.8.

TABLE 7.6: Performance Accuracy Comparison for Emotion Classifiers

——	Accuracy
SCNN. [135]	60
DLP-CNN [88]	84.22
IPA2LT [133]	86.77
gaCNN [136]	85.07
Motion Magnification based Emotion Classifier 4	88.47
Multimodal Network based Emotion Classifier 5	89.47 ±2
Image Animation based Emotion Classifier 6	**92.3** ±1

Fig. 7.9 and Fig. 7.10 show the results generated from the Stage 2 of various mixed emotions on the RAVDSS dataset. Results generated from the Stage-3 6.4 is shown in Fig. 7.11. The blue outline depicts the face region being cropped by the SSD Face detector and the image adjacent to it is the 2D generated image using the Image Animation Module. The Emotion classifier predicts all the 7 emotion classes with high inference speed of 25 FPS. Table 7.6 shows the performance accuracy of the image animation model with previously implemented algorithms in Chapter 4 and Chapter 5.

7.4 Summary

This Chapter provides the experimental results of the simulations for the implemented algorithms for emotion classification such as EVM, MAN, Image-Avatar Animation Algorithm. The performance accuracies of the algorithms are tabulated 7.6 and the model which has the highest inference speed and accuracy is the Image-Avatar Animation with an accuracy of 92.23%. By further summarizing the main research contributions of this thesis, Chapter 8 will draw conclusions and provide an outlook for future work.

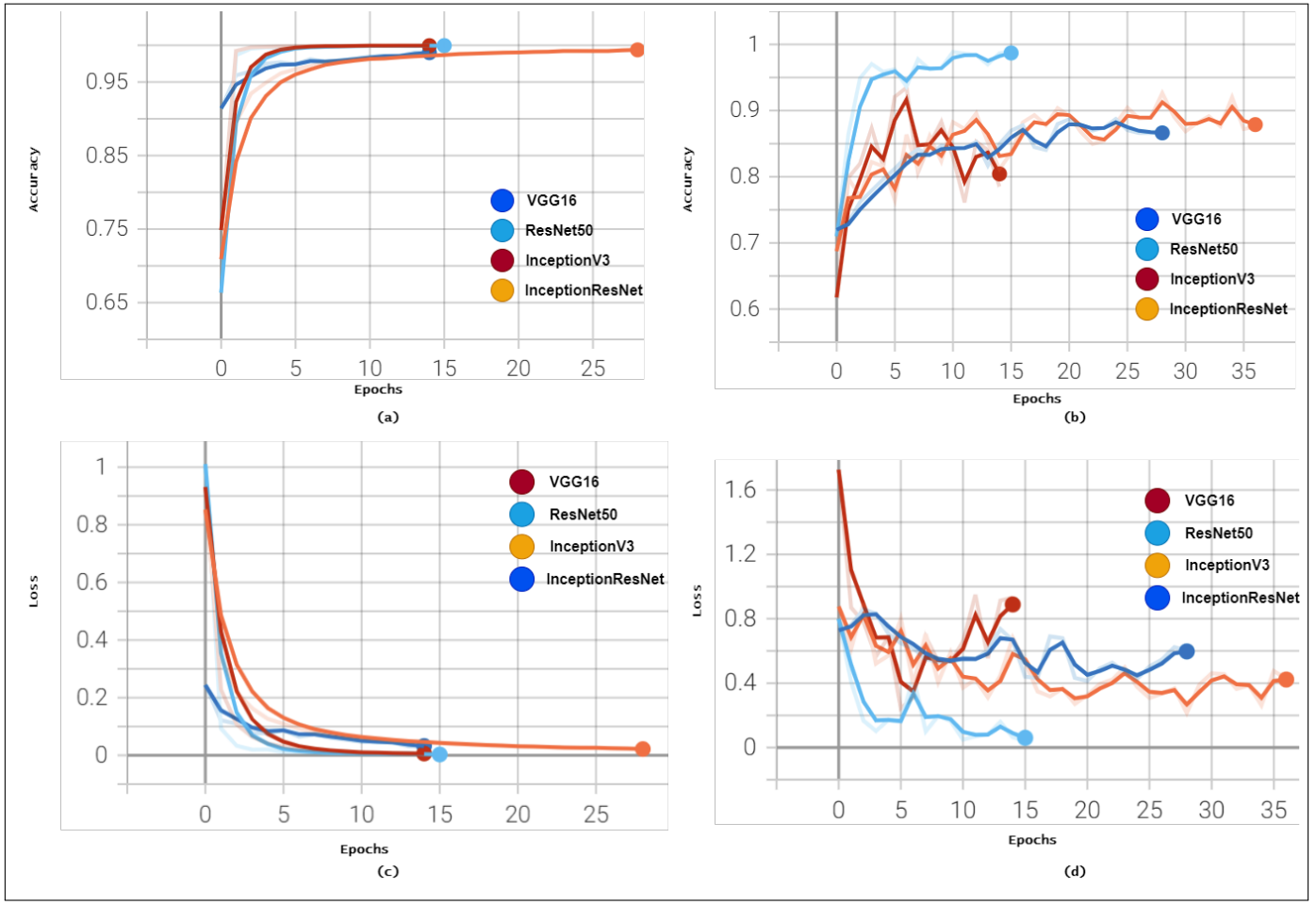

FIGURE 7.8: (a) and (b) shows the training and validation accuracy curves of different networks when used for transfer learning as a backbone to train the FERG emotion classifier. (c) and (d) shows the training and validation loss curves for the former.

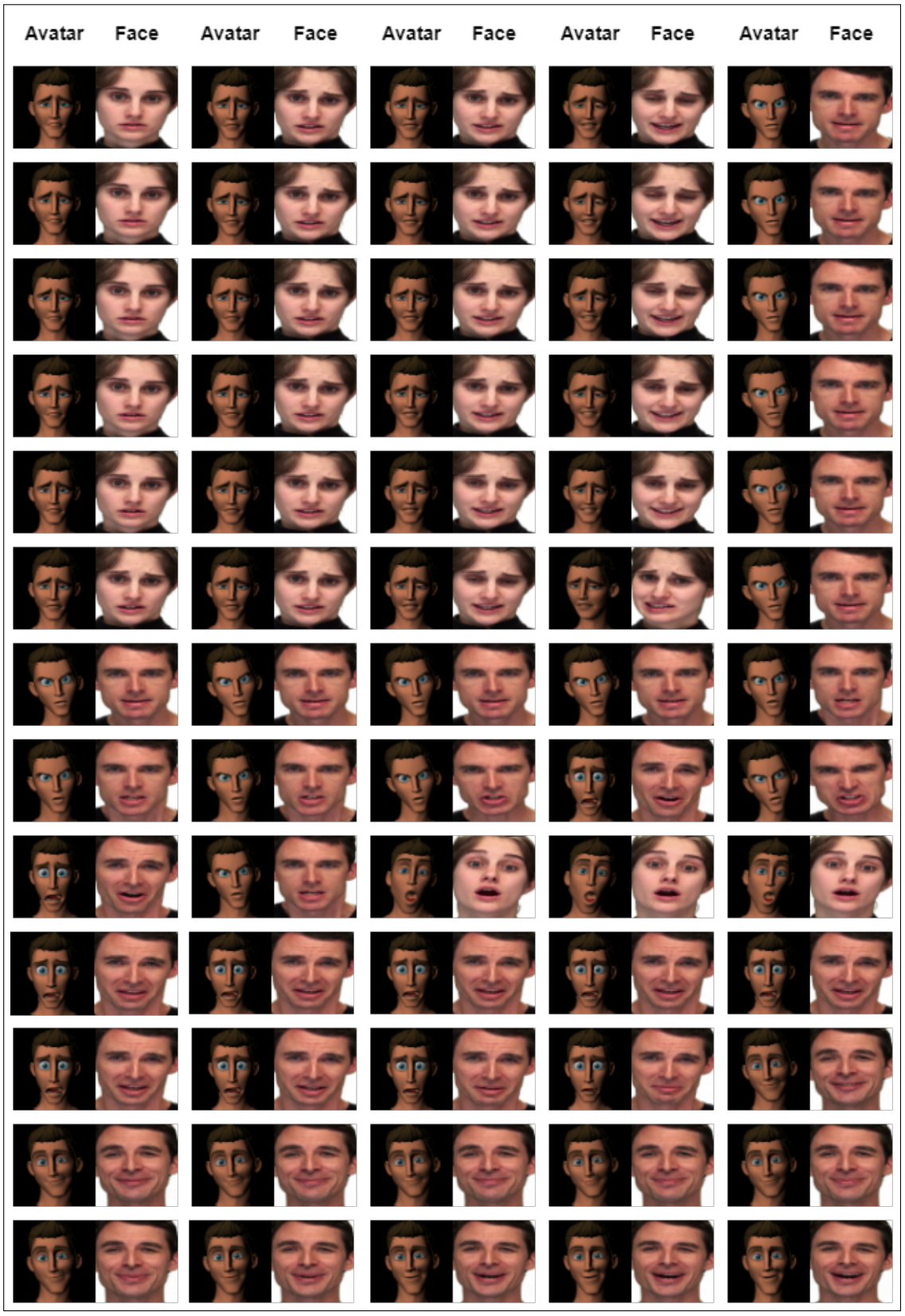

FIGURE 7.9: Results of the Image to Avatar Generation Module, Part (a).

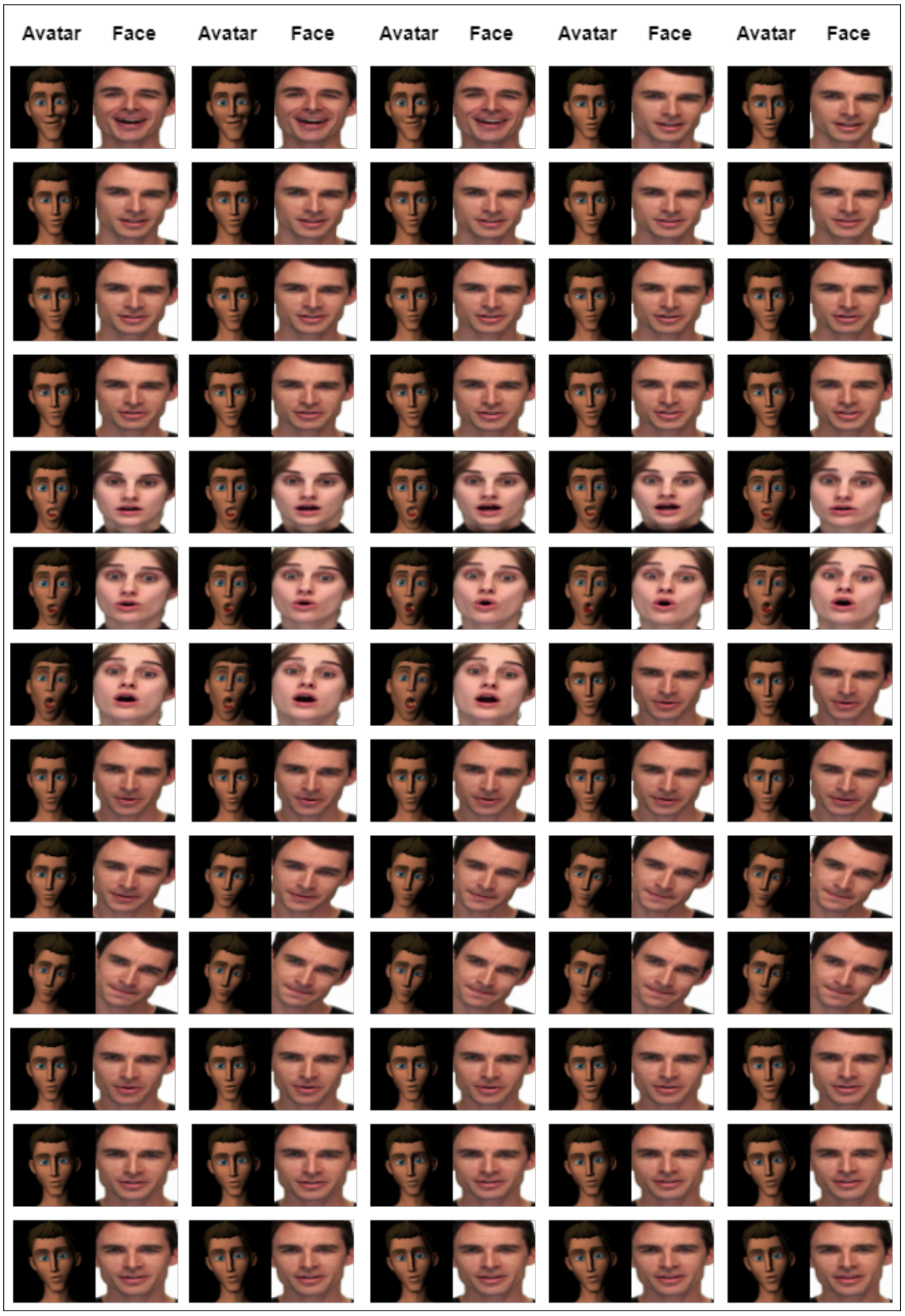

FIGURE 7.10: Results of the Image to Avatar Generation Module Part (b).

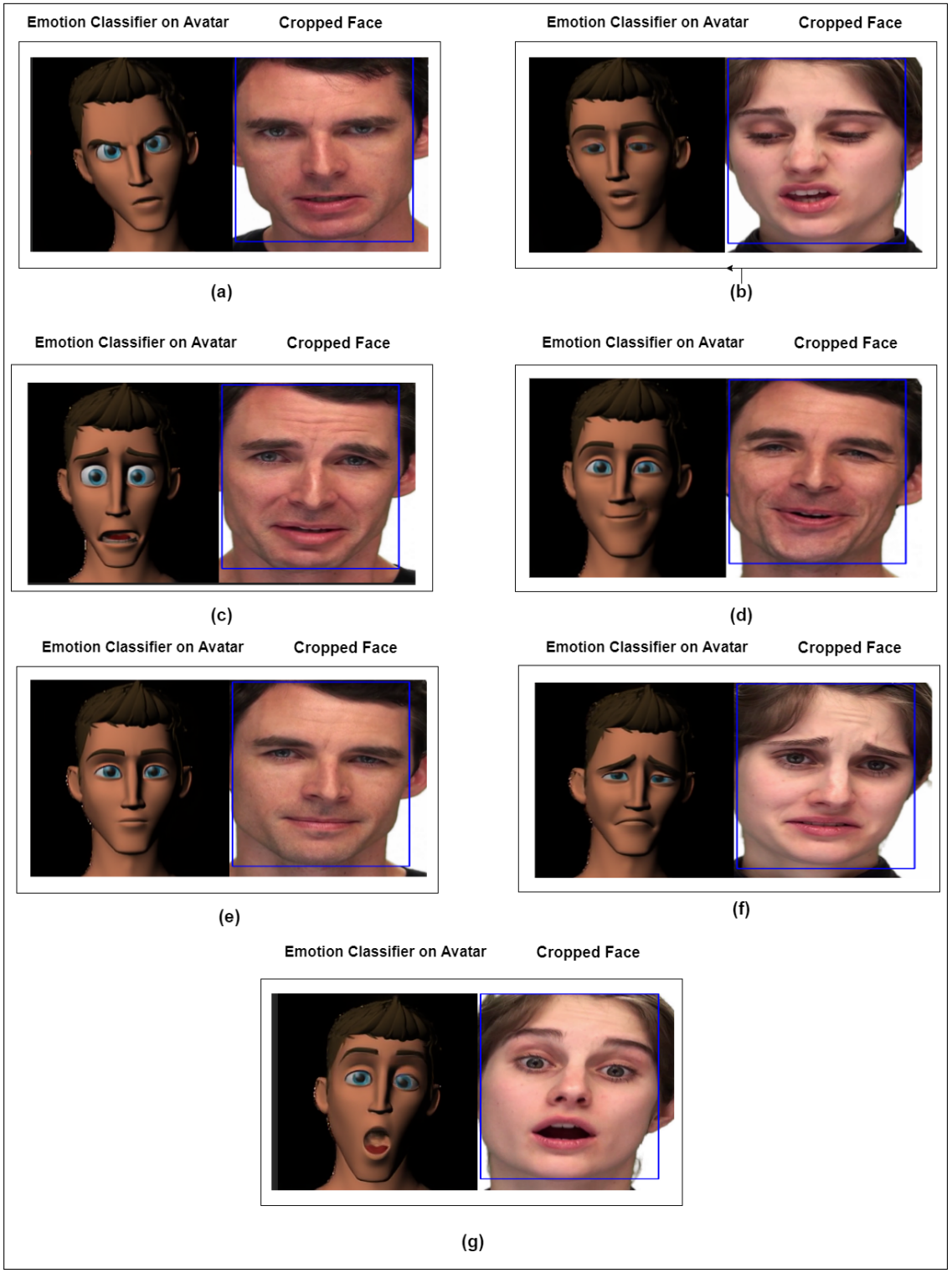

FIGURE 7.11: Results of the Emotion Classifier using Image Animation.(a) shows the angry emotion, (b) shows the disgust emotion, (c) shows the fear emotion, (d) shows the happy emotion, (e) shows the neutral emotion, (f) shows the sad emotion and (g) shows the surprised emotion

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Facial Micro-Expressions detection and emotion classification systems are very demanding and require substantial efforts in enhancing the performance metrics of the trained models therefore the precision of the emotion identifier. Due to the copious applications of the emotion recognition in real-world scenarios, affective computing is gaining a lot of attention and numerous studies are being conducted in detecting the real-time ME's due to its practical importance in diverse fields like robotics, medical treatments, Advanced Driver-Assistance Systems (ADAS) and other such human-computer interactive systems.

This thesis introduces different implementations of architectures and algorithms for achieving a stable emotion classifier with high inference accuracy of 92.3%. The Eulerian Motion Magnification increases the intensity of the subtle expressions which stimulates the emotion classification elevating the accuracy to detect ME's but is computationally heavy because of which a multimodal approach was devised and experimented on. The multimodal approach was based on the study which showed emotions are dependent on audio-visual cues. The experiment analyzed the emotion using the speech synthesizer and the facial micro-expression network which further was passed on the aggregator network to obtain the final result. This showed automated emotion is possible and can be vital in not only analyzing the basic emotions but also the cognitive assessment and the physiological response which was validated using the BEQ. Even though this model performed with a

high accuracy, the inference was still in the near-real time phase because of which an image animation based emotion classifier was devised.

Due to the absence of datasets which could be used to train more efficient models, there was a need to clean the datasets accessible. A Non-Local Means algorithm was used to remove the excess noise in the datasets which affected the performance of the trained model. The image animation algorithm was used to animate the real-time video using one of the driving images from the test dataset which removed all the excess noise from the facial terrain. This experiments showed an improvement in both the performance metrics and the accuracy tested in real-time making it suitable to implement in day-to-day applications and scenarios.

8.2 Future Work

The devised architectures and algorithms for emotion classification aim to increase the performance metrics by making them easier to implement and deploy. The work presented in this thesis is worth pursuing further in the direction of incorporating the emotion classifier with the face recognition systems. This section identifies some of the avenues for potential improvements and enhancements to the work presented in this thesis.

8.2.1 Micro-Expressions Datasets

Emotion classification being a data-driven task and training a neural network to capture the subtle expressions in the facial regions do require a large amount of training data without which FER systems lack quality. FER models are biased because of people with different ages, cultures and displays and an ideal dataset is expected to inculcate a wide range of samples with not only precise facial attribute labels but also with other attributes such as age, gender and ethnicity [31]. Also, there are some other factors which can contribute to the construction of these datasets like face occlusions and head-pose annotations, data bias and inconsistent annotations which is a fundamental issue causing discrepancies of the trained models evaluated on different datasets making the models lack of generalizability on unseen test data.

Due to the different dataset collections environments, the FER models can't achieve the desired performance on cross datasets [133] [137]. Automatic labelling tools can also be an alternative to provide efficient annotations even though they are approximations. Overall, a subsequent and reliable estimation is required to denoise the annotations and images. The data pre-processing and denoising can also solve the problem of class imbalance.

8.2.2 Multiple Affective Models

Micro-Expressions are often associated with the pre-defined emotional landscape of the FACS model, where the various different facial muscle AUs are combined to describe the overall visible appearance change of the micro-expression. This model consists of two parameters valence and arousal which continuously encode the subtle changes further defining the emotional intensity. Du et al. [138] stated that a majority of facial expressions are actually combinations of more than once basic emotions which improves the overall definitions of the facial expression and can even elevate the model. Therefore, new filters of neural networks can be defined whom can be associated weights in accordance with the importance of different facial muscle.

References

[1] S. Freud, "The standard edition of the complete psychological works of sigmund freud," 1953.

[2] P. Ekman, "An argument for basic emotions," *Cognition & Emotion*, vol. 6, pp. 169–200, 1992.

[3] ——, "Emotions revealed : Understanding faces and feelings," 2003.

[4] J. Tao and T. Tan, "Affective computing: A review," 10 2005, pp. 981–995.

[5] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "Affectnet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 18–31, 2019.

[6] N. Jadhav and R. Sugandhi, "Survey on human behavior recognition using affective computing," in *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)*, 2018, pp. 98–103.

[7] Z. Zhi and H. Jinde, "Emotion computing method based on knowledge representation," in *2020 International Conference on Computer Engineering and Application (ICCEA)*, 2020, pp. 368–372.

[8] G. Du, S. Long, and H. Yuan, "Non-contact emotion recognition combining heart rate and facial expression for interactive gaming environments," *IEEE Access*, vol. 8, pp. 11 896–11 906, 2020.

[9] H. Zhang, "Expression-eeg based collaborative multimodal emotion recognition using deep autoencoder," *IEEE Access*, vol. 8, pp. 164 130–164 143, 2020.

[10] S. K. Jarraya, M. Masmoudi, and M. Hammami, "Compound emotion recognition of autistic children during meltdown crisis based on deep spatio-temporal analysis of facial geometric features," *IEEE Access*, vol. 8, pp. 69 311–69 326, 2020.

[11] P. Ekman and E. Rosenberg, "What the face reveals : basic and applied studies of spontaneous expression using the facial action coding system (facs)," 2005.

[12] K.-Y. Huang, C.-H. Wu, Q.-B. Hong, M.-H. Su, and Y.-H. Chen, "Speech emotion recognition using deep neural network considering verbal and nonverbal speech sounds," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5866–5870.

[13] Y. Yang, Q. Wu, M. Qiu, Y. Wang, and X. Chen, "Emotion recognition from multi-channel eeg through parallel convolutional recurrent neural network," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–7.

[14] P. Fusar-Poli, A. Placentino, F. Carletti, P. Landi, P. Allen, S. Surguladze, F. Benedetti, M. Abbamonte, R. Gasparotti, F. Barale, J. Perez, P. McGuire, and P. Politi, "Functional atlas of emotional faces processing: a voxel-based meta-analysis of 105 functional magnetic resonance imaging studies." *Journal of psychiatry & neuroscience : JPN*, vol. 34 6, pp. 418–32, 2009.

[15] M. Bernstein and G. Yovel, "Two neural pathways of face processing: A critical evaluation of current models," *Neuroscience & Biobehavioral Reviews*, vol. 55, pp. 536–546, 2015.

[16] A. O'Toole, D. Roark, and H. Abdi, "Recognizing moving faces: a psychological and neural synthesis," *Trends in Cognitive Sciences*, vol. 6, pp. 261–266, 2002.

[17] J. Haxby, E. Hoffman, and M. I. Gobbini, "The distributed human neural system for face perception," *Trends in Cognitive Sciences*, vol. 4, pp. 223–233, 2000.

[18] O. Zinchenko, Z. Yaple, and M. Arsalidou, "Brain responses to dynamic facial expressions: A normative meta-analysis," *Frontiers in Human Neuroscience*, vol. 12, 2018.

[19] G. Warren, E. Schertler, and P. Bull, "Detecting deception from emotional and unemotional cues," *Journal of Nonverbal Behavior*, vol. 33, pp. 59–69, 2009.

[20] W. Yan, X. Li, S.-J. Wang, G. Zhao, Y.-J. Liu, Y. Chen, and X. Fu, "Casme ii: An improved spontaneous micro-expression database and the baseline evaluation," *PLoS ONE*, vol. 9, 2014.

[21] F. Qu, S. Wang, W. Yan, and X. Fu, "Cas(me)2: A database of spontaneous macro-expressions and micro-expressions," in *HCI*, 2016.

[22] A. K. Davison, C. Lansley, N. Costen, K. Tan, and M. H. Yap, "Samm: A spontaneous micro-facial movement dataset," *IEEE Transactions on Affective Computing*, vol. 9, no. 1, pp. 116–129, 2018.

[23] L. S. Chen, H. Tao, T. S. Huang, T. Miyasato, and R. Nakatsu, "Emotion recognition from audiovisual information," *1998 IEEE Second Workshop on Multimedia Signal Processing (Cat. No.98EX175)*, pp. 83–88, 1998.

[24] M. Peng, C. Wang, T. Chen, G. Liu, and X. Fu, "Dual temporal scale convolutional neural network for micro-expression recognition," *Frontiers in Psychology*, vol. 8, 2017.

[25] X. Hao and M. Tian, "Deep belief network based on double weber local descriptor in micro-expression recognition," in *MUE/FutureTech*, 2017.

[26] M. H. Yap, H. Ugail, and R. Zwiggelaar, "Facial behavioral analysis: A case study in deception detection," *British Journal of Applied Science and Technology*, vol. 4, pp. 1485–1496, 2014.

[27] N. Sebe, I. Cohen, and T. Gevers, "Multimodal approaches for emotion recognition: A survey," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5670, 12 2004.

[28] ——, "Multimodal approaches for emotion recognition: A survey," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5670, 12 2004.

[29] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, pp. 1 – 62, 2020.

[30] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (gans): A survey," *IEEE Access*, vol. 7, pp. 36 322–36 333, 2019.

[31] S. Li and W. Deng, "Deep facial expression recognition: A survey," *ArXiv*, vol. abs/1804.08348, 2018.

[32] D. Hebb, "The organization of behavior: A neuropsychological theory," 1949.

[33] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biology*, vol. 52, pp. 99–115, 1990.

[34] D. Hubel and T. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of Physiology*, vol. 195, 1968.

[35] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 2004.

[36] 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, pp. 611 – 629, 2018.

[37] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. A. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[38] Y. Cun, "A theoretical framework for back-propagation," 1988.

[39] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "1 efficient backprop," 2012.

[40] L. Datta, "A survey on activation functions and their relation with xavier and he normal initialization," *ArXiv*, vol. abs/2004.06632, 2020.

[41] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *IWANN*, 1995.

[42] B. Karlik and A. Olgac, "Performance analysis of various activation functions in generalized mlp architectures of neural networks," 2011.

[43] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[44] A. L. Maas, "Rectifier nonlinearities improve neural network acoustic models," 2013.

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ArXiv*, vol. abs/1502.03167, 2015.

[46] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate Research*, vol. 30, pp. 79–82, 2005.

[47] N. Zhang, S. Shen, A. Zhou, and Y.-S. Xu, "Investigation on performance of neural networks using quadratic relative error cost function," *IEEE Access*, vol. 7, pp. 106 642–106 652, 2019.

[48] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision," in *A Guide to Convolutional Neural Networks for Computer Vision*, 2018.

[49] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, p. 265–292, Mar. 2002.

[50] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *ArXiv*, vol. abs/1702.05659, 2017.

[51] Y. Guo, Y. Tian, X. Gao, and X. Zhang, "Micro-expression recognition based on local binary patterns from three orthogonal planes and nearest neighbor method," in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 3473–3479.

[52] C. Pelachaud, N. I. Badler, and M. Steedman, "Generating facial expressions for speech," *Cognitive Science*, vol. 20, no. 1, pp. 1–46, 1996.

[53] L. De Silva and P. C. Ng, "Bimodal emotion recognition," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, 2000, pp. 332–335.

[54] D. H. Kim, W. J. Baddar, and Y. M. Ro, "Micro-expression recognition with expression-state constrained spatio-temporal feature representations," *Proceedings of the 24th ACM international conference on Multimedia*, 2016.

[55] W. Merghani, A. K. Davison, and M. H. Yap, "A review on facial micro-expressions analysis: Datasets, features and metrics," *ArXiv*, vol. abs/1805.02397, 2018.

[56] B. M. Talukder, B. Chowdhury, T. Howlader, and S. M. Rahman, "Intelligent recognition of spontaneous expression using motion magnification of spatio-temporal data," in *Proceedings of the 11th Pacific Asia Workshop on Intelligence and Security Informatics - Volume 9650*. Berlin, Heidelberg: Springer-Verlag, 2016, p. 114–128.

[57] X. Li, T. Pfister, X. Huang, G. Zhao, and M. Pietikäinen, "A spontaneous micro-expression database: Inducement, collection and baseline," in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2013, pp. 1–6.

[58] L. Chen and T. Huang, "Emotional expressions in audiovisual human computer interaction," in *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, vol. 1, 2000, pp. 423–426 vol.1.

[59] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *INTERSPEECH*, 2012.

[60] Z. Lu, Z. Luo, H. Zheng, J. Chen, and W. Li, "A delaunay-based temporal coding model for micro-expression recognition," in *ACCV Workshops*, 2014.

[61] S. K. A. Kamarol, N. S. Meli, M. H. Jaward, and N. Kamrani, "Spatio-temporal texture-based feature extraction for spontaneous facial expression recognition," *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pp. 467–470, 2015.

[62] S.-T. Liong, J. See, R. C.-W. Phan, and K. Wong, "Less is more: Micro-expression recognition from video using apex frame," *Signal Process. Image Commun.*, vol. 62, pp. 82–92, 2018.

[63] S.-T. Liong, J. See, R. C.-W. Phan, Y.-H. Oh, A. C. L. Ngo, K. Wong, and S.-W. Tan, "Spontaneous subtle expression detection and recognition based

on facial strain," *Signal Process. Image Commun.*, vol. 47, pp. 170–182, 2016.

[64] Y.-H. Oh, A. C. L. Ngo, R. C.-W. Phan, J. See, and H.-C. Ling, "Intrinsic two-dimensional local structures for micro-expression recognition," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1851–1855, 2016.

[65] X. Ben, X. Jia, R. Yan, X. Zhang, and W. Meng, "Learning effective binary descriptors for micro-expression recognition transferred by macro-information," *Pattern Recognit. Lett.*, vol. 107, pp. 50–58, 2018.

[66] D. Jain, Z. Zhang, and K. Huang, "Random walk-based feature learning for micro-expression recognition," *Pattern Recognit. Lett.*, vol. 115, pp. 92–100, 2018.

[67] C.-W. Huang and S. S. Narayanan, "Deep convolutional recurrent neural network with attention mechanism for robust speech emotion recognition," *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 583–588, 2017.

[68] P.-W. Hsiao and C.-P. Chen, "Effective attention mechanism in dynamic models for speech emotion recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2526–2530, 2018.

[69] Y. Li, T. Zhao, and T. Kawahara, "Improved end-to-end speech emotion recognition using self attention mechanism and multitask learning," in *INTERSPEECH*, 2019.

[70] B. J. Abbaschian, D. Sierra-Sosa, and A. S. Elmaghraby, "Deep learning techniques for speech emotion recognition, from databases to models," *Sensors (Basel, Switzerland)*, vol. 21, 2021.

[71] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2227–2231, 2017.

[72] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial networks," *ArXiv*, vol. abs/1406.2661, 2014.

[73] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *AISTATS*, 2010.

[74] S. Latif, R. K. Rana, and J. Qadir, "Adversarial machine learning and speech emotion recognition: Utilizing generative adversarial networks for robustness," *ArXiv*, vol. abs/1811.11402, 2018.

[75] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[76] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[77] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[78] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2016.

[79] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *ArXiv*, vol. abs/1411.1784, 2014.

[80] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *NIPS*, 2016.

[81] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *ICML*, 2017.

[82] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow, "Adversarial autoencoders," *ArXiv*, vol. abs/1511.05644, 2015.

[83] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," *ArXiv*, vol. abs/1611.02163, 2017.

[84] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," *ArXiv*, vol. abs/1612.02136, 2017.

[85] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *ICML*, 2017.

[86] S. Livingstone and F. 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, 2018.

[87] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[88] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 2584–2593.

[89] S. Li and W. Deng, "Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 356–370, 2019.

[90] F. Song, X. Tan, X. Liu, and S. Chen, "Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients," *Pattern Recognition*, vol. 47, no. 9, pp. 2825–2838, 2014.

[91] R. Fusek, "Pupil localization using geodesic distance," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11241 LNCS, pp. 433–444, 2018.

[92] M. J. Lyons, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets (ivc special issue)," *ArXiv*, vol. abs/2009.05938, 2020.

[93] M. J. Lyons, ""excavating ai" re-excavated: Debunking a fallacious account of the jaffe dataset," *ArXiv*, vol. abs/2107.13998, 2021.

[94] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge," *Image Vision Comput.*, vol. 47, no. C, p. 3–18, Mar. 2016.

[95] ——, "300 faces in-the-wild challenge: database and results," *Image and vision computing*, vol. 47, pp. 3–18, Mar. 2016.

[96] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 896–903.

[97] V. Le, J. Brandt, Z. L. Lin, L. D. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *ECCV*, 2012.

[98] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," in *CVPR*, 2011.

[99] D. Aneja, A. Colburn, G. Faigin, L. Shapiro, and B. Mones, "Modeling stylized character expressions via deep learning," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 136–153.

[100] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Comput. Speech Lang.*, vol. 60, 2020.

[101] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *Lecture Notes in Computer Science*, p. 21–37, 2016.

[102] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. T. Freeman, "Eulerian video magnification for revealing subtle changes in the world," *ACM Transactions on Graphics (Proc. SIGGRAPH 2012)*, vol. 31, no. 4, 2012.

[103] S. C. Ayyalasomayajula, B. Ionescu, and D. Ionescu, "A cnn approach to micro-expressions detection," *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 345–350, 2021.

[104] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[105] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *ArXiv*, vol. abs/1704.04861, 2017.

[106] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. Ieee, 2001, pp. I–I.

[107] D. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 07 2009.

[108] A. C. Le Ngo, Y. Oh, R. C. . Phan, and J. See, "Eulerian emotion magnification for subtle expression recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 1243–1247.

[109] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. Senior, "Recent advances in the automatic recognition of audiovisual speech," *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1306–1326, 2003.

[110] T. Küntzler, T. T. A. Höfling, and G. W. Alpers, "Automatic facial expression recognition in standardized and non-standardized emotional expressions," *Frontiers in Psychology*, vol. 12, 2021.

[111] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2017.

[112] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015.

[113] J. Cech and T. Soukupova, "Real-time eye blink detection using facial landmarks," *Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague*, pp. 1–8, 2016.

[114] M. S. Likitha, S. R. R. Gupta, K. Hasitha, and A. U. Raju, "Speech based human emotion recognition using mfcc," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 2257–2260.

[115] M. R. Sampford *et al.*, "An introduction to sampling theory with applications to agriculture." *An introduction to sampling theory with applications to agriculture.*, 1962.

[116] C. Huang, C. Guoming, Y. Hua, B. Yongqiang, and Z. Li, "Speech emotion recognition under white noise," *Archives of Acoustics*, vol. 38, no. 4, pp. 457–463, 2013.

[117] M. Laghari, M. J. Tahir, A. Azeem, W. Riaz, and Y. Zhou, "Robust speech emotion recognition for sindhi language based on deep convolutional neural network," in *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2021, pp. 543–548.

[118] S. Alim and N. K. Alang Md Rashid, *Some Commonly Used Speech Feature Extraction Algorithms*, 12 2018.

[119] S. Chakroborty, A. Roy, and G. Saha, "Fusion of a complementary feature set with mfcc for improved closed set text-independent speaker identification," in *2006 IEEE International Conference on Industrial Technology*, 2006, pp. 387–390.

[120] J. Gross and O. John, "Revealing feelings: facets of emotional expressivity in self-reports, peer ratings, and behavior." *Journal of personality and social psychology*, vol. 72 2, pp. 435–48, 1997.

[121] ——, "Mapping the domain of expressivity: multimethod evidence for a hierarchical model." *Journal of personality and social psychology*, vol. 74 1, pp. 170–91, 1998.

[122] J. Gross, O. John, and J. Richards, "The dissociation of emotion expression from emotion experience: A personality perspective," *Personality and Social Psychology Bulletin*, vol. 26, pp. 712 – 726, 2000.

[123] Şebnem Tunay Akan and E. Barışkın, "[reliability and validity indicators of berkeley expressivity questionnaire in the context of culture and gender]." *Turk psikiyatri dergisi = Turkish journal of psychiatry*, vol. 28 1, pp. 43–50, 2017.

[124] X. Wang, J. Gong, M. Hu, Y. Gu, and F. Ren, "Laun improved stargan for facial emotion recognition," *IEEE Access*, vol. 8, pp. 161 509–161 518, 2020.

[125] A. A. Dixit and A. C. Phadke, "Image de-noising by non-local means algorithm," in *2013 International Conference on Signal Processing , Image Processing Pattern Recognition*, 2013, pp. 275–277.

[126] B. Coll and J.-M. Morel, "A non-local algorithm for image denoising," vol. 2, 07 2005, pp. 60– 65 vol. 2.

[127] X. Zhang, G. Hou, J. Ma, W. Yang, B. Lin, Y. Xu, W. Chen, and Y. Feng, "Denoising mr images using non-local means filter with combined patch and pixel similarity," *PloS one*, vol. 9, p. e100240, 06 2014.

[128] K. Leng, "An improved non-local means algorithm for image denoising." in *IEEE International Conference on Signal and Image Processing (ICSIP)*, 2017.

[129] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," *ArXiv*, vol. abs/2003.00196, 2019.

[130] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[131] J. I. Tam, "Exploring fast style transfer," 2016.

[132] S. Adiga, D. Vaishnavi, S. Saxena, and S. Tripathi, "Multimodal emotion recognition for human robot interaction," in *2020 7th International Conference on Soft Computing Machine Intelligence (ISCMI)*, 2020, pp. 197–203.

[133] J. Zeng, S. Shan, and X. Chen, "Facial expression recognition with inconsistently annotated datasets," in *ECCV*, 2018.

[134] Y. Li, J. Zeng, S. Shan, and X. Chen, "Occlusion aware facial expression recognition using cnn with attention mechanism," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2439–2450, 2019.

[135] S. Adiga, D. Vaishnavi, S. Saxena, and S. Tripathi, "Multimodal emotion recognition for human robot interaction," in *2020 7th International Conference on Soft Computing Machine Intelligence (ISCMI)*, 2020, pp. 197–203.

[136] Y. Li, J. Zeng, S. Shan, and X. Chen, "Occlusion aware facial expression recognition using cnn with attention mechanism," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2439–2450, 2019.

[137] X. Wei, H. Li, J. Sun, and L. Chen, "Unsupervised domain adaptation with regularized optimal transport for multimodal 2d+3d facial expression recognition," *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 31–37, 2018.

[138] S. Du, Y. Tao, and A. M. Martinez, "Compound facial expressions of emotion," *Proceedings of the National Academy of Sciences*, vol. 111, pp. E1454 – E1462, 2014.