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

Deep Learning Algorithms for Mood Detection and Appropriate Quranic Verses Suggestion

A Thesis

Submitted to the Council of the College of Information Technology for Postgraduate Studies of the University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology/Software

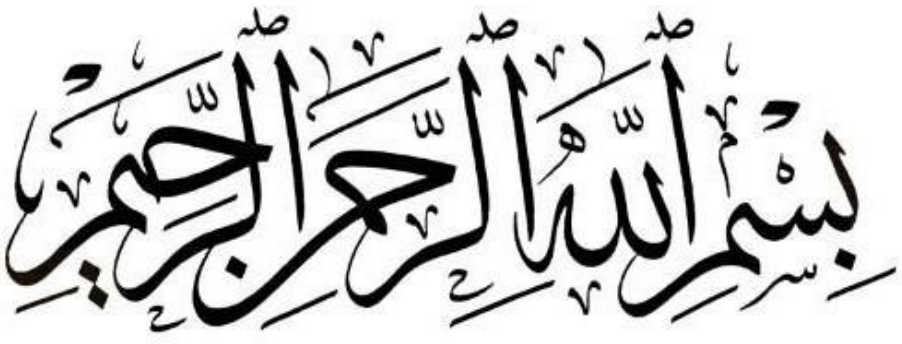
By

Anwar Salah Mishaan Prism

Supervised by

Asst. Prof. Dr. Nashwan Jassim Hussain

2023A.D. **1445 A.H.**



قُلْ هَلْ يَسْتَوِي الَّذِينَ يَعْلَمُونَ وَالَّذِينَ لَا يَعْلَمُونَ إِنَّمَا

يَتَذَكَّرُ أُولُوا الْأَلْبَابِ

سورة الزمر الآية 9

Declaration

I as a result of this declare that this thesis entitled "**Deep Learning Algorithms for Mood Detection and Appropriate Quranic Verses Suggestion**", submitted to the University of Babylon in partial fulfilment of requirements for the degree of Master in Information Technology \ Software, has not been proposed as an exercise for a similar degree atany other University. I also certify that this work described here is entirely my own except for experts and summaries whose source is appropriately cited in the references.

Signature:Name:

Date: / / 2023

Supervisor Certification

I certify that this thesis is prepared under my supervision at the Department of Software / Collage of Information Technology / Babylon University, by **Anwar Salah Mishaan** as a partial fulfillment of the requirements for the degree of Master in Information Technology

Signature:

Supervisor Name: Dr. Nashwan Jassim Hussain

Date: / / 2023

The Head of the Department Certification

In view of the available recommendations, I forward this thesis for debateby the examination committee.

Signature:

Assist. Prof. Dr. Ahmed Saleem Abbass

Head of Software DepartmentDate: / / 2023

Acknowledgements

In the name of Allah, the Most Gracious, the Most Merciful, I begin by expressing my deepest gratitude and praise to Allah Almighty for His guidance and blessings that enabled me to successfully complete this task.

I would like to extend my sincere appreciation and respect to my supervisor, Dr. Nashwan Jassim Hussain, for his invaluable guidance, support, and supervision throughout this endeavor. His expertise and wisdom have been instrumental in shaping this thesis and guiding me towards its successful completion.

I am also grateful to the staff of the College of Information Technology for their unwavering support, cooperation, and assistance throughout my academic journey. Their dedication and commitment to providing a conducive learning environment have greatly contributed to the quality of this work.

Furthermore, I would like to express my heartfelt thanks to all the individuals who have offered their support, whether directly or indirectly, throughout this research. Although it is not possible to mention everyone by name, their contributions and encouragement have been truly invaluable and have played a significant role in my personal and academic growth.

Abstract

The recognition of facial expressions is of paramount importance in comprehending human emotions and has garnered considerable interest within the realm of computer vision and deep learning. Detecting facial expressions in videos is considered a difficult and interesting task since the face is the main means of communication and the most communicative part of the body to display emotions. The current thesis provides a detailed investigation of facial expression detection and its application to a proposed Quranic system that relies on emotions. This thesis aims to apply deep learning approaches to efficiently recognize and classify emotions shown in face photographs. This thesis leverages the capabilities of convolutional neural networks (CNN) (one dimension and two dimensions) and time-distributed layers in order to effectively capture temporal dependencies within video sequences on the CREMA-D dataset. The system has undergone extensive training and testing, resulting in a high level of accuracy in the identification of six fundamental emotions: anger, disgust, fear, happiness, neutrality, and sadness.

This thesis improves its field by developing a new architecture for video-based emotion recognition. Preprocessing, feature extraction, and classification utilizing 1D-CNN and 2D-CNN models comprise the architectural framework. The 1D-CNN model classifies features after being extracted by the histogram of oriented gradients (HOG), while the 2D-CNN model features extraction and classification simultaneously. The accuracy of the CNN 1D model is 0.99 , which indicates that it produces high results. Additionally, the 2D CNN worked superbly, with an accuracy score of 0.82. These results show the system's capability to identify facial expressions ,detecting emotion, and designing emotion-based systems for Suggest Quranic Verse.

Table of Contents

List of Tables

List of Figure

List of Algorithms

List of Abbreviations

Abbreviation	Meaning
Acc	Accuracy
ANN	Artificial Neural Networks
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DT	Decision Tree
FN	False Negative
FP	False Positive
KNN	K-Nearest Neighbors
LR	Logistic Regression
LR	Linear Regression
ML	Machine Learning
MLR	Multivariate Linear Regression
NB	Naïve Bayes
NN	Neural Network
PCA	Principle Component Analysis
RF	Random Forest
SVM	Support Vector Machines
SVR	Support Vector Regression
TN	True Negative
TP	True Positive
VAE	Variational Auto Encoder

Chapter One
General Introduction

Chapter One
General Introduction

1.1. Introduction

Facial expressions are essential indicators of our emotional states and serve as a primary mode of communication, it plays a significant role in communicating emotion [1]. In recent times, deep learning algorithms have shown great potential for improving the accuracy and efficiency levels of facial expression recognition systems significantly [2]. The recognition of facial expressions has become increasingly precise and reliable in recent times due to advancements made in machine learning and computer vision [3].

The ability to recognition of facial expressions has several practical uses. In the medical field, it can help doctors better understand their patients' mental health issues and formulate effective treatment plans. Emotion recognition may be used in the field of customer service to get a feel for how happy customers are and adjust strategies accordingly. User experiences may be improved through human-computer interaction when emotion detection allows for more natural and individualized interactions with technology [4].

Gaming and entertainment could employ emotional reaction recognition to tailor and engage players based on their mood. Emotion detection can also improve situational awareness and identify high-stress situations in law enforcement and security. Recognition of pupils' facial expressions can assist teachers improve classroom instruction and comprehend their emotions. Deep learning models provide an advantage over prior emotion-recognition systems, but they need additional research for high-robust expression detection [5].

In addition, deep learning-based algorithms for Quranic verse recommendation can be used to examine user emotions and behavior and

provoke a wide spectrum of emotional reactions beyond verbal or textual interactions. In contrast to past initiatives that merely classified general human emotions indicated by facial expressions, such as joy or sadness, these ideas are personalized. The program recommends Quranic verses based on users' emotions[6–9].

1.2. Thesis Motivation

1. With the development of deep learning models, progress has been made in the area of facial emotion recognition. Many existing approaches to emotion identification also depend on antiquated methodologies or on relatively tiny datasets filled with performed emotions, which may restrict their usefulness and accuracy.
2. Use deep learning algorithms to create a system that can accurately identify emotions. This system will not only be able to categorize photos into the six universal emotions, but it will also be at recognizing and analyzing minute differences in face expressions.
3. The Quran Verse Suggest system is able to tailor Verse to each individual user depending on their current emotional state by using deep learning techniques trained on previously observed emotional patterns. By doing away with potentially erroneous self-reporting and user biases, this new method improves and personalizes the Quranic experience for each user.

1.3. Problem Statement

Facial expression recognition is difficult to define and categorize, making it a difficult research topic. Teaching machines to decipher facial expressions is difficult. Dynamic facial expressions include modest muscular movements that shape their appearance. Advanced algorithms and methods are needed to capture and analyze these minute differences. Deep learning's popularity has expanded emotion recognition applications.

However, choosing a Quranic verse based on emotion and necessity is difficult. There are many Quranic verses with different storylines, purposes, and morals. This makes choosing a Quran verse to match user feeling difficult. This will be done by sensing user emotions from their faces.

1.4. Research Questions

The following research questions will be expanded academically in this thesis:

1. How can deep learning algorithms consistently and accurately recognize and categorize facial expressions?
2. What is the mechanism employed for interning a sequence of frames into deep learning models?
3. What is the influence of preprocessing and feature extraction approaches on the performance of the system in boosting emotion identification and communication?
4. How could verses in the Quran be selected to correspond with the identified emotional states?

1.5. Thesis Aim and Objectives

The aim of this thesis is to develop and train a Deep Learning model for accurate emotion recognition and suggest Quranic varse based on detected emotions. While the objectives of this thesis are:

1. to develop a comprehensive system architecture capable of accurately and efficiently detecting and recognizing facial emotions.
2. To combine (detecting emotions and suggesting Quranic verses) opens new possibilities for study and application to the science of facial expressions, emotion identification, and deep learning.

1.6. Related Works

Deep learning methods have been used in numerous studies on facial expression recognition as illustrated in Table (1.1).

Ristea et al. (2019) [10] demonstrated a real-time emotion recognition system based on deep Convolutional Neural Networks. By analyzing speech data as well, and fusing visual and aural cues, they were able to improve the recognition system's precision. The value of merging visual and aural input is demonstrated by the experimental findings, demonstrating the efficacy of the suggested strategy for emotion identification. Facial expression recognition accuracy was 69.42% using the CREMA-D datasets.

Ryumina and Karpov (2020) [11] tested a distance importance scoring based feature extraction strategy for face emotion recognition using CREMA-D and RAVDESS databases. Facial expression recognition accuracy was 79.1% on the CREMA-D database and 98.9% on the RAVDESS database, compared to methods based on facial graphical regions.

Birhala et al (2020) [12] proposed a multimodal fusion strategy for emotion identification was presented. This technique combines audio-visual modalities from a temporal window with distinct temporal offsets for each modality. They demonstrate that the suggested strategy performs better than other methods found in the literature as well as the human accuracy rating. Experiments are carried out using CREMA-D, a multimodal dataset that is freely available to the public.

Ghaleb et al (2020) [13] proposed a Multimodal Emotion Recognition Metric Learning to achieve a discriminative score and robust latent-space representation for both modalities (MERML). Radial Basis Function (RBF)-based Support Vector Machine (SVM) kernel makes efficient use of the acquired measure. Evaluation on the eNTERFACE and CREMA-D datasets shows that prposed performs better than the state-of-the-art.

Sujanaa and Palanivel (2020) [14] identified feelings by suggesting technique

employs a Histogram of Oriented Gradients (HOG). A Haar-Based Cascade classifier is used to identify mouth areas at a frame rate of 20.Next, a One-Dimensional Convolutional Neural Network is trained using the HOG features as input (1D-CNN). Based on testing data, the suggested system has a 90.23 percent success rate in recognizing the three emotions, making it a clear winner over its predecessors.

Hans and Rao (2021) [15] used 75 frames of masked facial photos for each prediction and offers a CNN-LSTM based Deep Neural Network architecture to do so. On the CREMA-D dataset, a 6-layer CNN-LSTM trained with a learning rate of 0.0001 yielded a test accuracy of 78.53 percent, while the same model tested on the RAVDEES dataset yielded a result of 63.35 percent. The model's accuracy could be enhanced by incorporating speech characteristics and Facial Action Units into the same architecture and then stacking these features together before sending them to the LSTM layer.

Sujanaa et al (2021) [16] employed a dataset that consists of still frames of videos showing images of people's mouths conveying various emotions. Images of people's faces have their mouths removed using a Haar-based cascade classifier, and frames are taken from the video at a rate of 20 per second. Each histogram in the feature set represents a different image of a person's mouth, and each histogram is based on using tools like HOG (histogram of oriented gradients) and LBP (local binary pattern) (LBP). Two methods, accelerated robust features (SURF) and scale-invariant feature transform (SIFT), are utilized to separate out individual data points. Texture features are used in the training of a support vector machine (SVM) and a one-dimensional convolutional neural network (1D-CNN). Test video frames are fed into trained models that look for signs of emotion, and the results of the experiments show that the accuracy of SVM is 97.44% and that of 1D-CNN is 98.51%.

Zamani and Wulansari (2021) [17] studied how emotions are classified and proposed two models that combine the best features of the In order to achieve this

objective, utilizing the One-Dimensional Convolutional Neural Network (CNN-1D) as well as the Recurrent Neural Network (RNN).The RNN design includes Gated Recurrent Units (GRUs) and Long Short-Term Memories (LSTMs) to address the vanishing gradient issue common to time series data. High-Value-High-Arousal (HVHA), Low-Value-High-Arousal (HVLA), Low-Value-Low-Arousal (LVHA), and Low-Value-Low-Arousal (LVLA) are the four emotional zones that our model distinguishes (LVLA).The popular DEAP dataset was used in this experiment. According to the experiments, the training accuracy of the suggested approaches is 96.3% for the 1DCNN-GRU model and 97.8% for the 1DCNN-LSTM model. This emotion classification task is therefore well within the capabilities of both models.

Vijay and Yasutomo (2022) [18] suggested a transformer-based paradigm improves audio-visual emotion identification. This novel model has three multimodal transformer branches: one for audio processing and one for visual processing. Cross attention between auditory and visual stimuli is computed by the third strand. These branches detect relevant information in both modalities and any interactions that might affect users' emotional state analysis. The study's best results came from ablation of these three locations. They also propose block embedding, a new temporal embedding method that uses time information from many video frames to improve visual characteristics. Public audio-visual datasets RAVDESS, CREMA-D, and SAVEE were used to validate the design.

Comparisons with other models and a detailed ablation investigation were also done. Based on observations, our multi-modal transformer architecture is more successful than baseline techniques.

Srinivas and Mishra (2022) [19] introduced a multimodal system for emotion recognition that integrates characteristics from disparate modalities, such as audio and video. Energy, zero crossing rate, and Mel-Frequency Cepstral Coefficients (MFCC) are all strategies taken into account while extracting audio features. The findings from MFCC are very encouraging. First, using a spatial temporal Gaussian Kernel, the films are split into frames and saved in a linear scale space. Applying a Gaussian weighted function to the second momentum matrix of linear scale space further extracts characteristics from the photos. The audio and video features are fused using the Marginal Fisher Analysis (MFA) fusion method, and the combined features are then sent into the FERCNN model for analysis. Experiments employ audio and video data from the RAVDESS and CREMAD databases. Performance is improved over previous multimodal systems, with accuracy values of 95.56, 96.28, and 95.07 on the RAVDESS dataset and 80.50, 97.88, and 69.66 on the CREMAD dataset in audio, video, and multimodal modalities.

Table (1.1): Summary of the Related Works.

Reference and year	Dataset	Model	Accuracy
Ristea et al. (2019) [10]	CREMA-D	CNN	(62.48%, CREMA-D)
Ryumina and Karpov (2020) [11]	CREMA-D RAVDEES	LSTM	(79.1%, CREMA-D) (98.9%, RAVDESS)
Birhala et al (2020) [12]	CREMA-D	CNN	(68.4%, CREMA-D)
Ghaleb et al (2020) [13]	CREMA-D eNTERFACE	(MERML)	(66.5, CREMA-D) (91.5, eNTERFACE)
Sujanaa and Palanivel (2020) [14]	dataset is collected using a web camera	1D-CNN	90.23%
Hans and Rao (2021) [15]	CREMA-D RAVDEES	CNN-LSTM	(78.53%, CREMA-D) (63.35%, RAVDESS).
Sujanaa et al (2021) [16]	collected using a web camera	SVM and 1D-CNN	(97.44%, SVM) (98.51%,1D-CNN)
Zamani and Wulansari (2021) [17]	DEAP	1DCNN-GRU 1DCNN-LSTM	(96.3%, 1DCNN-GRU) (97.8%, 1DCNN-LSTM)
Vijay and Yasutomo (2022) [18]	RAVDESS, CREMA-D, and SAVEE	CNN	(RAVDESS ,93.59) (CREMA-D ,72.45) (SAVEE ,99.17)
Srinivas and Mishra (2022) [19]	RAVDESS, CREMA-D	FERCNN model	(RAVDESS ,96.28) (CREMA-D, 97.88)

1.9. Thesis Outline

The subsequent sections of this research work are structured as follows:

- **Chapter 2 :** This chapter presents in-depth assessment of the pertinent

literature on facial expression recognition, mood detection and Quran recommendation systems. It assesses various deep learning techniques used for analyzing facial expressions and evaluates different approaches to detect moods with respect to recommending suitable Quran.

- **Chapter 3 :** This chapter introduce the proposed methodology for recognizing facial expressions and detecting moods incorporating a comprehensive explanation of dataset integration, architecture design of Deep Learning Models. Furthermore, it outlines the training process employed in carrying out face expression analysis coupled with emotion estimation using AI technology integrating harmony vectors towards improving personalized Quran Recommendations Systems.
- **Chapter 4 :** Contains the outcomes and discussions derived from the proposed methodology. The chapter aims to determine how effective a deep learning model is in recognizing facial expressions and detecting moods through an evaluation of accuracy, precision, recall, and F1-score metrics. Furthermore, user feedback will be used to evaluate the Quran recommendation system's efficiency based on satisfaction levels. Additionally, it analyzes the strengths and limitations of the method utilized while identifying areas for improvement along with applications that come with its implementation.
- **Chapter 5:** Provides a summary of research findings based on results obtained earlier; conclusions drawn are also specified here accordingly while elaborating potential avenues for future exploration within facial expression recognition systems alongside mood detection mechanisms including Quran recommendation software development as parting note.

Chapter Two
Theoretical Background

Chapter Two

Theoretical Background

2.1. Introduction

This chapter provides a comprehensive overview of the literature on human emotions, with a focus on facial expression analysis. It will explain how various computer programs can be utilized for recognizing facial expressions. The current state of research and practical applications for identifying these feelings will be given via a review of the literature. Since deep learning (DL) has emerged to help put ML into practice, particularly through the application of neural networks for both learning and prediction [20] . Various deep learning algorithms that can be used for face expression recognition are covered in this section.

2.2. Human Emotions Representation Overview

In the twenty-first century, computers are ubiquitous and play a crucial role in society. The ability to recognize emotion is being built into computers, and one day they may even be programmed to experience emotions. Emotional theory has a long and illustrious history, dating back to the Stoics, Plato, and Aristotle of Ancient Greece [21]. The classic Aristotelian theory of emotions by Aristotle examines the evolution of his thinking on emotions by defining and explaining a wide range of emotions, contrasting and comparing them, and characterizing the emotions themselves. Remarkable insights emerged from his ideas, such as the following: Emotions, such as anger, pity, fear, and its opposites, are the reasons people undergoing transformations have different perspectives and experience both positive and negative feelings [5].

The usage of biometric data is becoming increasingly commonplace, with examples ranging from fingerprint scanning technology for logging into secure databases to face recognition via passport photos for gaining entry to a nation.

The primary goal is to provide a more foolproof method of identification than,

say, a password. By removing the need for human intervention, this technology streamlines the process and increases security, reducing the likelihood of forgery or fraud. Biometric identification can be accomplished using a variety of human characteristics, including the face, iris, voice, and fingerprints [22, 23].

A further subfield of HCI is devoted to processing feelings; this area Is known as affective computing. Researchers in the field of computer vision use these datasets and attempt to decipher their machine-readable meaning. Many academics are focusing on this problem, hoping to find a way to automatically identify affective feelings [24, 25]. One's choices in many domains can be influenced by awareness of one's emotional condition.

- Recognizing pain when there are no words to express it. When a doctor isn't present, a radiotherapy patient may feel such intense pain that he can't even utter a sound or move his eyes from where they're fixed ahead.
- Identifying symptoms of subject despair and hostility to provide early warning and avert potential incidents. If staff have access to information on patients' affective emotions, they will be better equipped to assess the patients' behavior.
- Machines can be used to detect the earliest stages of autism and other disorders that manifest as weaknesses or observable differences from a healthy individual.
- Service to Customers To ensure customer satisfaction, it's important for operators to gauge their customers' emotional states so they can avoid asking questions that could inflame the situation.
- If a car is able to identify when its driver is getting tired or sleepy, it can issue warnings to the driver to pull over or find the closest rest stop for them.

To reach a future when machines can comprehend how to communicate and handle humans, further investigation is required in this area to enhance the naturalness of human-computer interaction and assist whenever it is required .

2.2.1. Facial Expressions

Expressions of emotion and mental condition are frequently conveyed through a person's each emotion that determines how strongly an expression will be shown. Intentions, action tendencies, assessments, other cognitions, neuromuscular and physiological changes, expressive behavior, and subjective emotions all fall under this category [2]. Because of these factors, the face muscles contract to produce an expression that may be seen by other people. Ekman [26] identified six core facial expressions representing the full range of human emotions (happiness, surprise, fear, sadness, anger, and disgust; see Figure 2.1). The basic displays of emotion can be expanded upon to convey a wide range of nuanced sentiments.

Figure 2.1: A screenshot of six essential lines from the Bosphorus corpus [26].

2.2.2. Image Attributes

The image can be normalized into a vector space and various features extracted from it. There are a number of methods we may use to deduce the feeling, such as calculating the ellipses on the face or the angles between the various features. The following are examples of salient characteristics that can be utilized to educate AI systems:

A. Faces

To assign a numerical value to a facial expression, the Facial Action Coding System is employed. An "action unit" is any one of these numbers. A face expression is the consequence of a combination of action units. A facial activity unit is a measure of the minute contractions and relaxations of facial

muscles. A pleasant expression, such as a smile, can be broken down into the coordinated movements of six action units (AU6) and twelve (AU12) of muscle. The cheek riser is represented by Action Unit 6, and the lip corner puller by Action Unit 12. An action unit-based facial action coding system is useful for identifying which face muscles contribute to a given expression. They can be used to create face models that can be used in real time[27].

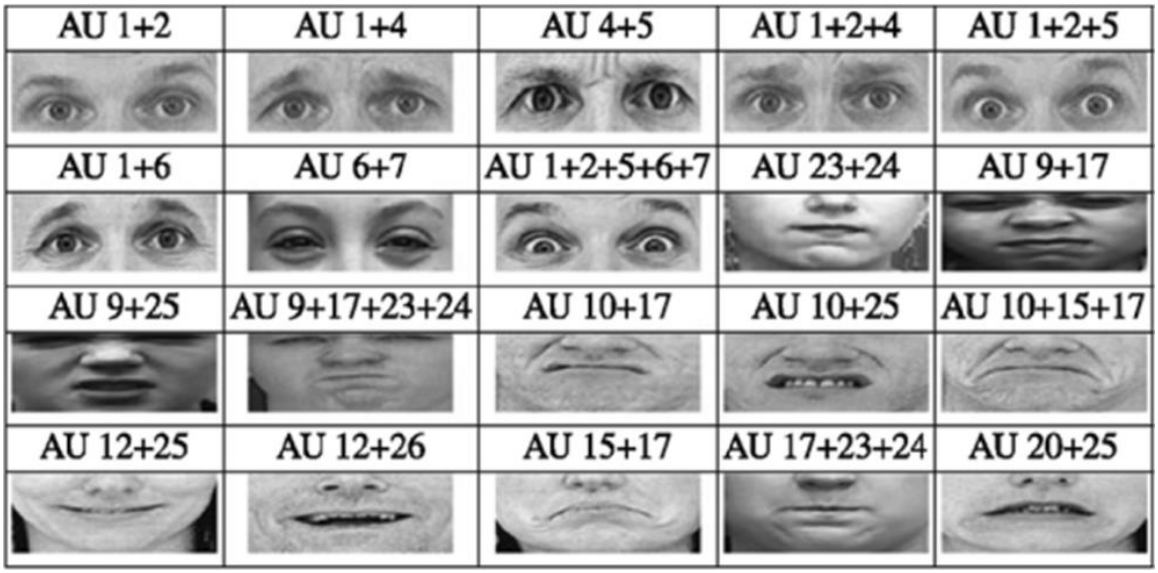


Figure 2.2: Action Units corresponding to different movements in face [27].

B. Landmarks

Important and useful facial landmarks can be used for face recognition and identification. The same anchors apply to expressions as well. The 68 facial landmark detector that pinpoints the location of 68 facial landmarks. Figure (2.3) shows the 68 different facial landmarks. The x,y coordinates of each facial point can be retrieved with the help of the dlib library. Each of the 68 points corresponds to a distinct body part, such as the left or right eye, the left or right eyebrow, the mouth, the nose, or the jaw[1].

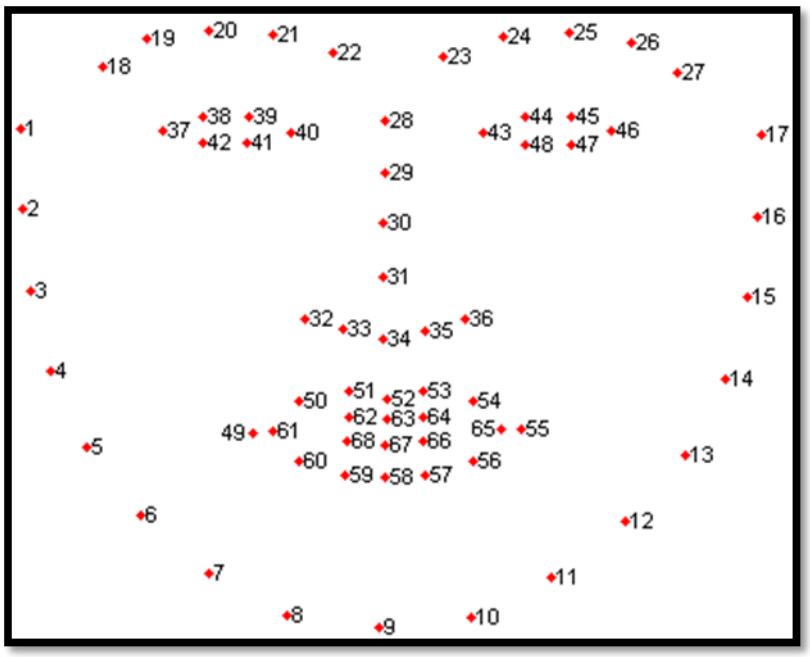


Figure 2.3: Landmarks on face [1].

2.3. Models for Color Digital Images

The use of color digital images is extensive across multiple fields, such as computer vision, image processing, and multimedia applications. Various models have been created to handle the representation and manipulation of color information in digital images. One commonly employed model is the RGB model which combines different intensity levels of these three primary colors to represent colors. The RGB model is widely supported by imaging software and devices due to its intuitive nature. Another well-known color model is CMYK, frequently utilized in printing and graphic design practices [28, 29].

The CMYK color model is utilized to represent colors by utilizing the concept of subtractive color mixing. This involves combining different inks to achieve the desired colors. Moreover, there are alternative color models such as HSV, Lab, and YUV/YcbCr that offer different perceptual properties and aid specific image processing tasks. These models have significant importance in analyzing, manipulating, and comprehending color images. They enable the creation of algorithms and techniques for various applications related to color

analysis[30]. Using a Deep Learning Model for Emotion Recognition necessitates the following preprocessing steps:

1. **RGB to Gray Conversion:**

By linearly combining the red, green, and blue primary colors, the RGB color model can be used to define any conceivable color. Computers and televisions both employ this color display technology. A digital color picture is composed of three RGB 2-D matrices, one for each basic color. The visual on the screen is constructed by multiplying the values of these three matrices together[29], [30]. Typically, 8 bits are used to represent each of the three components of the three matrices. As can be seen in Figure(2.4), there are a total of 24 bits in a color pixel (3 x 8).

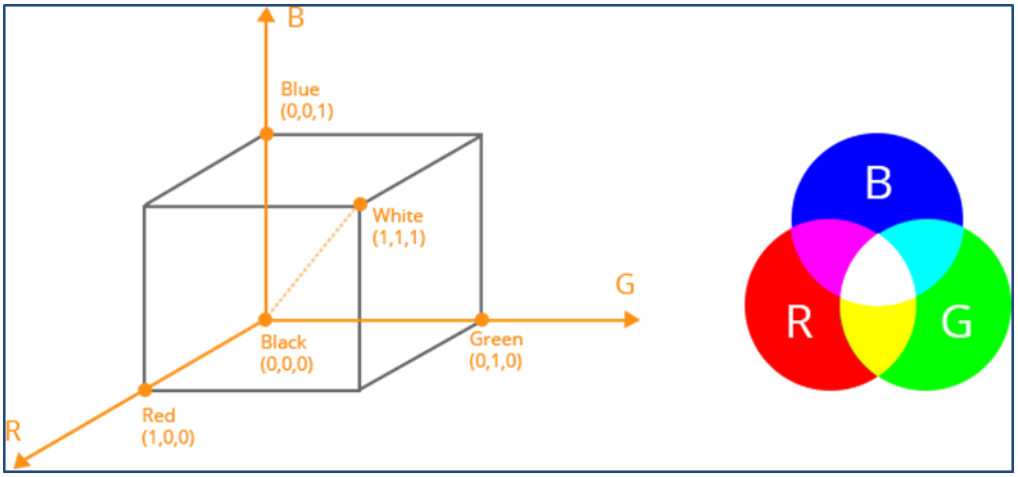


Figure 2.4: RGB to Gray[31].

2. **Grey color model:**

Gray scale images only have one color channel, while RGB (Red Green Blue) images have three. It is possible to lower the processing burden and simplify the data by converting the RGB image to grayscale. The formula for the transformation is as follows [29], [30]:

$$Grayscale\ Value = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (2.1)$$

2.4. Enhancement of Digital Images

The visual quality of photographs as well as their interpretability may be significantly improved by the application of various digital image enhancement techniques. This section will focus on a method known as histogram equalization, which is one of these techniques. The process of histogram equalization is one of the most common techniques for boosting the contrast of digital photos as well as improving their overall look.

Histogram Equalization

Histogram equalization dispersing intensity values improves visual contrast. Histogram equalization is performed on an input picture having pixel values from 0 to L-1 (where L is the number of intensity levels, generally 256 for an 8-bit image)[32, 33]:

1. Create a histogram of the supplied image to see how often each brightness level appears. The histogram value at intensity level I will be denoted by H(i).

2. Calculate the cumulative distribution function (CDF) of the histogram. The CDF, denoted by CDF(i), represents the cumulative probability of each intensity level up to level i. It is computed using the following equation [32]:

$$CDF(i) = \Sigma H(j) \; for \; j = 0 \; to \; i \qquad (2.2)$$

Map intensity values from [0, L-1) to [0, L-1] by normalizing the cumulative distribution function (CDF). The new intensity values are comprehensive because of the normalizing procedure. The following formula is used to calculate the intensity at any level I [32]:

$$NewValue(i) = round((CDF(i) - CDF(min)) / (M * N - 1) * (L - 1)) \qquad (2.3)$$

Where M*N is the number of pixels in the picture, L is the number of

intensity levels, and NewValue(i) is the new intensity value CDF(i) is the cumulative probability of intensity level I is the minimal cumulative probability among all intensity levels; and.

3. Apply the following equation to each pixel in the supplied picture, replacing its value with NewValue(i).

By using the histogram equalization method, the final image will have better aesthetic appeal and be more appropriate for further image analysis and processing because of the increased contrast and more even distribution of intensity values [34].

Algorithm (2.1): Histogram Equalization

Input: The pixel values of a grayscale facial image span from 0 to 255.

Output: An enhanced grayscale facial image.

BEGIN

Step 1: Understanding the dimensions of the grayscale image (E*F) requires pixel values between 0 and 255. Set every element of a matrix G of size 256 to 0.

Step 2: To create an image histogram, the corresponding elements must be updated in the matrix. – The image matrix is updated by scanning each pixel to create the histogram.

G[gray_value(pixel)] = G[gray_value(pixel)] + 1

Step 3: Using Eq. (2.2), CDF calculations can be performed.

CH[0] = H[0]

Each pixel (1 to 255) has CH[i] equal to CH[pixel-1] plus H[pixel].

Step 4: Referring to Eq (2.3), compute the updated pixel values through the process of general histogram equalization.

Step 5: Update image-wise, new values replace the original grayscale image.

NewImage[E][F] = T[OldImage[E][F]]

END

2.5. Standard Scaler

The standard scaler is a core preprocessing method in machine learning, with the goal of normalizing a dataset's features such that they all have the same mean value of zero and the same standard deviation of one. For many algorithms, especially those that are sensitive to the magnitude of input characteristics, this change is crucial for boosting convergence and speed. By subtracting the mean (μ) of the feature's values from each individual value and dividing by the standard deviation (σ) of the feature's values, the standard scaler calculates the Z-score for each feature[35]:

$$z = \frac{x-\mu}{\sigma} \quad (2.4)$$

Each data point is denoted by x, the feature mean is mu, the feature standard deviation is sigma, and the converted Z-score is z. In order to prevent bigger magnitude characteristics from dominating the learning process, the data is transformed so that it is centered around 0. This transformation also accounts for differences in scale. Therefore, the standard scaler is an important preprocessing step that contributes to the development of more accurate machine learning models across a variety of datasets.

2.6. Face Detection Algorithms

In the modern world, face identification jobs are needed increasingly frequently. It results from the creation of security measures in response to terrorist attacks. The speed of technological advancement is a result of computers' increasing intelligence. This new era of computer-human connection brought about by these intelligent computer's aids in the exploration of many different subjects [36]. One of the areas of computer-human interaction is face detection, a branch of object detection. The technique of object detection involves finding instances of items belonging to a specific class (such as people, cars, buildings, or

faces) in an image or video. In this new era, object detection has a wide variety of uses, including pedestrian and face detection [37].

Viola & Jones Algorithm

Facial expression recognition systems rely heavily on the face detection phase, which entails finding and pinpointing facial regions within an input image or video frame. The Viola and Jones technique is extensively used because it efficiently detects faces in real-time using a cascade of weak classifiers. Due to its high detection accuracy and fast processing speed, the Viola & Jones algorithm is well-suited for use in real-time settings [38]. Simple rectangular filters that record local intensity variations in the image form the basis of its operation, and these filters are known as Haar-like features. The system can recognize faces of diverse sizes and orientations thanks to the computation of these features at many scales and locations across the image [39].

The Viola and Jones face detection method consists of a number of sub-steps, each of which aids in the recognition and localization of individual facial features. In brief, the algorithm consists of the following steps [40]:

1. Haar-like features are basic rectangular filters that capture local picture intensity fluctuations. These traits allow face recognition. The method chooses Haar-like characteristics that can distinguish face and non-facial areas.

2. Calculating the integral picture accelerates Haar-like feature calculation. The integral picture sums pixel intensities in a rectangle region from the top-left corner. This equation efficiently computes [40]:

$$\text{Integral Image} = \text{I}_{\text{Integ}(x_2, \text{y}_2)} + \text{I}_{\text{Integ}(x_1-1, \text{y}_1-1)} - \text{I}_{\text{Integ}(x_1-1, \text{y}_2)} - \text{I\_Integ}(x_2, \text{y}_1 - 1) \quad (2.5)$$

where Integral Image is the integral picture value at coordinate (x_2, y_2) and Image(x_1, y_1) is the pixel intensity in the original image.

3. The approach uses the Adaboost machine learning technique to combine numerous weak classifiers into a strong one. The system picks a subset of training pictures and iteratively modifies the weights of weak classifiers depending on their performance in distinguishing face and non-facial areas.

4. The Viola & Jones technique uses a cascade of classifiers to efficiently reject non-facial areas and focus computation on possible face regions. Multiple weak classifiers make up each cascade level. An area is deleted if it doesn't fulfill a stage's criteria, decreasing computation for succeeding stages.

5. The technique uses a sliding window to scan the picture at multiple sizes and places. At each window point, the integral image computes Haar-like characteristics, and the cascade of classifiers sequentially determines if the region includes a face.

6. Eliminates redundant face detections and improves accuracy. Selecting the region with the greatest detection confidence score eliminates overlapped detections.

The Viola and Jones approach detects faces in real time using Haar-like features, integral pictures as illustrated in Algorithm (2.2), Adaboost training, cascade classifiers, sliding window detection, and non-maximum suppression[40].

Algorithm (2.2): Viola & Jones

Input: Image I

Output: A collection of image rectangles that correspond to the identified faces.

BEGIN

Step1: Initialize Haar-like features for face recognition

Step2: Calculate Integral Image for accelerated Haar-like feature calculation

```
for x from 1 to imageWidth:
    for y from 1 to imageHeight:
        use Eq.(2.5)
```

Step3: Implement Adaboost for combining weak classifiers into a strong one

Initialize weights for weak classifiers

for each training iteration:

- Normalize weights
- Train weak classifiers on subset of training images
- Calculate weak classifier error and importance
- Update weights based on classifier performance

Step4: Implement Viola & Jones cascade of classifiers for efficient face detection

```
for each cascade level:
    for each classifier:
        if region doesn't fulfill criteria:
            reject area and move to next stage
        else:
            continue to next stage
```

Step5: Perform sliding window face detection

for each window size:

```
                for x from 1 to imageWidth – windowWidth:
                    for y from 1 to imageHeight – windowHeight:
                        Compute Haar-like features using Integral Image
                        for each cascade level:
                            if region doesn't pass current cascade level:
                                break
                        if all cascade levels passed:
                            mark region as potential face

Step6: Eliminate redundant face detections and improve accuracy
                for each potential face region:
                    calculate detection confidence score
                eliminate overlapping detections by selecting regions with highest
            confidence scores
End
```

2.6. Resizing:

The computational cost of running the model can be decreased by making the image lower in size. Based on the needs of the model, the new size can be established. To scale an image, use this formula[41]:

$$new\_width = (scale\_x * old\_width)$$
$$new\_height = (scale_y * old_{height}) \quad (2.6)$$

where the width and height of the previous image are old width and old height, respectively, and the height of the new image is desired height. The input image for emotion recognition with a deep learning model can be optimized with the help of these preprocessing processes.

2.7. Feature Extraction

Human face detection, facial feature extraction, and facial expression recognition (FER) technologies all rely heavily on feature extraction. There are a

number of different approaches that can be used to extract useful and distinguishing characteristics from facial photos. Several feature extraction strategies currently in use within FER are discussed below[42].

1. Region and geometric-based techniques for face feature extraction collect global and local details. Global features consider the entire face, whereas local ones focus on specific areas. Region-based and geometric features are extracted using various approaches. Characteristics, including Gabor features, Local Binary Patterns (LBP), Local Ternary Patterns (LTP), Linear Discriminant Analysis (LDA), Histogram of Oriented Gradients (HOG), Active Shape Model (ASM), and Active Appearance Model (AAM). These techniques permit the description of facial features, textures, poses, and patterns[43].
2. Appearance-based feature extraction methods are useful for a variety of tasks, including facial emotion classification. Important facial cues can be captured by these techniques thanks to their use of texture-based, pose-based, and pattern-based representations. One such method that examines changes in facial texture is the Gabor wavelet representation. The accuracy of expression classification can be enhanced by using these strategies to extract change-sensitive elements from facial expressions[44].
3. Deep learning has made CNN-derived deep features common in FER. CNNs may learn hierarchical representations from facial pictures and discover highly discriminative features. Deep learning features might create missing data due to location and illumination changes or overlapping and non-overlapping areas. Deep learning features may need significant computational resources[42, 45].

In conclusion, FER's feature extraction methods cover a wide range of approaches, from those based on regions and geometry to those based on appearance to those based on deep learning. Different approaches have different advantages and disadvantages, thus picking the right one for a given facial expression identification task is essential.

Histogram of Oriented Gradients

Facial expression detection is just one application of computer vision and image processing where the Histogram of Oriented Gradients (HOG) is a common feature extraction approach. By evaluating the distribution of gradients in different orientations, the HOG method extracts local texture and shape information from an image. HOG feature extraction, when applied to facial emotion recognition, seeks to extract discriminative aspects of the face. The usual procedures are as follows[46, 47]:

- The quality of the supplied facial image is improved through preprocessing to remove noise and artifacts. To do this, it may be necessary to perform actions such as face detection and alignment.
- Computation of Image Gradients Entails Determining the Magnitude and Direction of Gradients for Each Individual Pixel in a Prepared Image. Each pixel's intensity shift is represented by the magnitude of the gradient, while the orientation of the gradient shows the direction of the shift [46, 47].

$$M(x,y) = (gx2 + gy2)1/2 \quad (2.7)$$

$$\theta\,(x,y) = atan\,(gy/\,gx) \quad (2.8)$$

- The picture is broken up into cells, often 8x8 or 16x16 pixels in size. Cell creation serves to preserve spatial linkages by encapsulating localized information within individual cells.
- A histogram of gradient orientations is calculated for each cell. Quantizing the gradient orientations into a set number of bins for the

various angles is done. The histogram represents local texture information concisely by capturing the distribution of gradient orientations Histogram(cell) = [H(0), H(1), ..., H(n-1)] [46, 47].

$$H(i) = \Sigma|G| \; for \; pixel \; with \; \theta \; in \; the \; i-th \; bin \quad (2.9)$$

- Using a technique called "block normalization," adjacent cells are clustered together to account for differences in illumination and contrast. To make the feature descriptor more resilient to these fluctuations, normalization techniques are applied to each block, such as contrast normalization or block-wise normalization.
- The histogram values of each block are combined to create a feature vector for the complete face image. Input to a classifier or machine learning system for facial expression recognition, this feature vector reflects the face expression.

Face expression recognition algorithms benefit greatly from HOG feature extraction because it accurately captures key face traits associated to expressions such wrinkles, eyebrow movement, and mouth shape. Classifiers that can tell the difference between neutral emotions and those conveying joy, sadness, rage, or surprise can be trained using the resulting feature vectors [48], [49].

When it comes to recognizing facial expressions, HOG feature extraction is invaluable because it provides a reliable and interpretable representation of local facial features that can be put to good use in a number of contexts, such as human-computer interaction, emotion analysis, and behavior understanding.

Algorithm (2.3): HOG feature extraction

Input: Image I

Output: HOG feature vector .

BEGIN

Step 1: Use a gradient operator to determine the magnitude and direction of the gradient for each pixel in the image (e.g., Sobel operator) using Eqs (2.7) and (2.8).

Step 2: Create a grid by slicing the image into cells of uniform size.

For each cell:

- Generate a histogram of cellular gradient orientations.
- The values in the histogram can be normalized by using histogram normalization (for instance, L2-norm) using Eq (2.9).

Step 3: Bring together the histograms of all cells in a set region.

Step 4: The block histograms should be normalized using a block normalization technique (such L2-norm).

Step 5: The final HOG feature vector is obtained by concatenating all the normalized block histograms.

Step 6: HOG feature vector should be returned.

END

2.8. Deep Learning

Deep learning (DL) is a new area of study focused on developing theories and algorithms that mimic human neural networks to enable machines to learn new information autonomously. Deep learning is a subfield of ML that was originally developed as an AI technique to mimic human learning processes in a particular domain [50]. Deep learning algorithms are layered in a complex and abstract hierarchy comprising of layers; each higher layer is built on the previous lower layer, hence the name hierarchical learning . Traditional machine learning algorithms follow a linear structure. In 2007, deep learning made its debut.

Microsoft, Facebook, Amazon, Google, and Baidu all utilize it regularly because of its capacity to process large datasets and solve difficult problems [51]. In order to describe complex structures in vast and massive datasets, deep learning uses computer models comprised of multiple processing layers to represent the data at varying degrees of granularity. As a novel method of machine learning, deep learning bridges the gap between traditional ML and AI. Object detection, speech recognition, and even medicine is just a few of the many applications of deep learning [52]. Learning is aided when the characteristics are informative and accurately characterize the database. In machine learning, feature extraction is a crucial process. However, natural information may be beyond the capabilities of typical machine learning to process and deal with in their raw form [53].

Since deep neural networks are superior to shallow ML methods in most applications involving the processing of text, picture, video, voice, and audio data, DL is also very beneficial in many fields with huge, high-dimensional data. When only a little amount of training data is available, shallow ML typically outperforms deep neural networks and is ideal for low-dimensional data entry. The rapid development of machine learning algorithms and processing is a major factor in the widespread use and success of deep learning[54, 55].

2.8.1. Convolutional Neural Network

The diagnosis and treatment of brain tumor rely heavily on accurate prognoses of the condition. There have been multiple attempts to develop deep learning models for this problem. When it comes to machine vision, convolutional neural networks (CNNs) are among the most popular forms of deep neural networks . The Convolutional Neural Network (CNN) is a deep learning approach that attempts to mimic how the brain processes information[56]. It is a subset of the feed-forward neural network family used in AI. CNN networks are quite similar to multi-layer networks (Perceptron), with the exception that they can integrate multiple locally connected networks into a single one. In addition to the layers employed for feature extraction, there are also layers employed for

categorization. Increased accuracy in automated diagnostic systems and disease prediction are two areas where CNN shows great promise. Due to their high data-processing capability, CNNs have become increasingly popular in the field of artificial intelligence[57].

In a convolutional neural network (CNN), data flows from one layer to the next, with each layer's output feeding into the next layer's input. The input layer is the first in the network, and the output layer is the final. The network's hidden layers are found between the input and output nodes. Each "layer" consists of a single "activation function" algorithm. Overall, the CNN model improved prediction accuracy by identifying high-level interactions between genes and the target collection[58].

1. Basic Components of CNN Architecture

It is possible to create a convolutional neural network that acts and thinks like a human brain. Prediction is a major advantage of this network in general. There are two primary elements that make up a CNN[59]:

A) Feature Extractor

Feature extraction and feature map creation is CNN's first step in processing data. CNN is made up of many filters, each of which performs a specific function. This led to the development of a plethora of feature maps, each of which represents a certain set of filters. The features vector with its low dimensionality is the output of the features extraction technique and is fed into a classifier. The feature extractor is built up from several layers (multiple convolution layers with optional pooling layers). First, the input and filter are convolved in a convolution layer to generate feature maps, which are then pooled down in a reduction layer. The output feature maps are then fed back into the system as input feature maps, where the process is repeated layer by layer in order to extract more sophisticated features. Finally, the feature maps with

reduced dimensions are flattened to produce a low-dimensional feature vector[60].

B) Classifier

After feature maps have been extracted, the best features from each have been selected to create a low-dimensional feature vector, which is then fed into a classifier. The likelihood of an input belonging to a certain class is reported by the classifier. One or more fully linked layers make up the classifier to do this [61].

2. Activation Functions

To introduce non-linearity into the activation map, non-linearity layers are typically placed immediately after the convolutional layer. This is because the convolution process is linear, while images are nonlinear. Some of the more common types of non-linear operations are outlined here[62]:

- **Sigmoid:** The sigmoid function is represented by the mathematical formula [63]:

  $f(x) = 1 / (1 + exp(-x))$ (2.10)

  It is used to convert a real number into a value between zero and one. However, when it comes to backpropagation, the vanishing gradient problem can occur with Sigmoid activation if the local gradient becomes extremely small and eventually disappears.
- **Tanh:** Tanh activation "squishes" a real number into the range of -1 to 1. Similar to sigmoid activation, this neuron's output also saturates; however, instead of being centered around an axis, it is zero-centered.
- **ReLU:** The Rectified Linear Unit activation function, denoted as [64]:

  $$f(\kappa) = max\ (0, \kappa)$$ (2.11)

  Can be calculated to determine the value of κ. In simpler terms, ReLU applies a threshold set at zero for activation. It is worth noting that ReLU has

several advantages over sigmoid and tanh functions. Firstly, it induces convergence six times faster than these other activation functions. Additionally, ReLU provides enhanced reliability in deep learning models[65].

- **Leaky ReLU:** Another variation of the ReLU function is called Leaky ReLU. Similar to its predecessor, Leaky ReLU sets a threshold at zero but incorporates a slight slope for negative values instead of being flat like regular ReLU[66]. The coefficient determining this slope is predetermined and remains constant throughout training iterations. Leaky ReLU may be formally defined as follows in Eq. (2.12):

$$f(x)_{LeakyReLU} = \begin{cases} x & if\ X > 0 \\ mx & if\ x \leq 0 \end{cases} \qquad (2.12)$$

If x is higher than zero, as in the input, the LeakyReLU acts like a regular ReLU function and lets the input through unmodified; otherwise, it does its usual thing when x is less than zero. A slight slope, m, is added to the output by the LeakyReLU when x is less than or equal to zero. Unlike ReLU, which completely shuts off the input, m-ReLU instead lets some of the input in. This solves the "dying ReLU" problem, which occurs when neurons get stuck during training and never update their weights.

3. Softmax Function

When given a vector of K real values, the softmax function will return another vector of K real values where each component sums to 1. The softmax transforms the input values, which might be positive, negative, zero, or greater than one, into a range from 0 to 1 so that they can be used to represent probabilities[67]. If one of the inputs is small or negative, Softmax will scale the probability down to 0 and scale it up to 1 if the other input is large.

Softmax is another term for multi-class logistic regression. Because of its similarity to the sigmoid function used in logistic regression, the softmax can be

applied to multi-class classification. The classifier's softmax function can be utilized only when the classes are completely unrelated[68].

In multi-layer neural networks, the intermediate layer is responsible for generating real-valued scores, which are not easily scaled. Because it transforms the scores into a probability distribution, the softmax is useful here because it may be shown to the user or used as input by other systems. The data from this distribution can be used in other applications as well. To this end, a softmax function is often used to finalize the neural network in Eq. (2.13) [69].

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j\equiv 1}^{k} e^{z_j}} \qquad (2.13)$$

For i=1,……,K and z=(z1,……,zk) €R. Here, zi is a numeric value that might be included in the input vector. As can be seen in Figure (2.5), a legitimate probability distribution necessitates the use of a normalization term at the very end of the computation to guarantee that the sum of the function's output values equals 1.

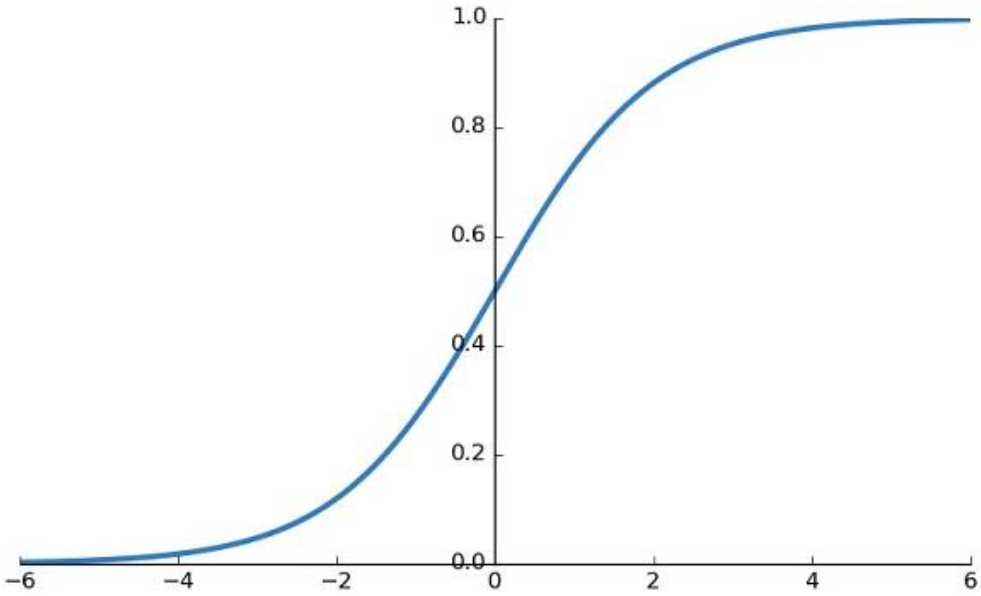


Figure 2.5: Softmax function [69].

5. Loss Function

A loss function can be used to find the best possible settings for the parameters of a neural network model. When applied to a set of network parameters, loss functions aggregate them into a single metric that quantifies the network's efficiency in completing the task at hand. Simply put, a loss function evaluates how well your prediction model performs in comparison to the desired result (or value). Through the introduction of a loss function, the original learning issue is transformed into one of optimization[70].

6. Adam Optimization Algorithm

Adam can change network weights iteratively based on training data, unlike stochastic gradient descent. Adam combines AdaGrad with RMS Prop to solve sparse gradients in noisy situations [71]. Adam dynamically scales learning rates for each parameter to increase convergence and speed. Averaging previous gradients and squared values yields adaptive learning rates for each parameter[72, 73]. Through two main equations:

1. The momentum term (first moment estimation) calculates the exponential moving average of gradients [71]:

$$m = \beta 1 * m + (1 - \beta 1) * g \quad (2.14)$$

Where m is the first moment estimation, which is essentially an exponentially weighted moving average of the gradients. It represents the average gradient values encountered so far during training. $\beta 1$ is the exponential decay rate for the first moment. It's a hyperparameter usually set to a value close to 1 (e.g., 0.9) that controls how much historical gradient information is retained. A higher value makes the moving average smoother. g is the current gradient. It represents the gradient of the loss function with respect to the model parameters at the current iteration.

2. The second moment estimation weights squared gradients exponentially [71].

$$v = \beta 2 * v + (1 - \beta 2) * g2 \quad (2.15)$$

v is the second moment estimation, which is also an exponentially weighted moving average, but of the squared gradients. It represents the average of the squared gradient values encountered during training. $\beta 2$ is the exponential decay rate for the second moment. Like $\beta 1$, it's a hyperparameter typically set close to 1 (e.g., 0.999). It controls how much historical squared gradient information is retained. $g2$ is the element-wise square of the current gradient.

These equations let Adam fine-tune each parameter's learning rate by estimating the gradient's momentum and variance over time. Last update iteration [71]:

$$\theta = \theta - \alpha * m / (\sqrt{v} + \varepsilon) \quad (2.16)$$

where m and v are the first and second moments, α is the learning rate, and is a tiny constant to avoid division by zero. The Adam optimizer combines momentum-based optimization with adaptive learning rates to improve deep learning model training by dynamically adjusting each parameter's learning rate depending on its past gradients[71, 74].

7. Learning Rate

The weights of the neural network adapt to the loss gradient based on the value of the hyper-parameter learning rate. The parameter controls how often the network reviews previously learned material. In order for the network to converge on a viable solution, The best pace of learning is somewhere between very slow and very fast (so that it can be taught in a practical length of time) [75].

Less time is needed to train a model with a higher learning rate since more substantial changes are made to the weights with each update. When training speeds are increased, the resulting final weight set is generally subpar. Epoch denotes the total number of weight updates applied to the model since the first time each training vector was utilized.

When training in batches, the learning algorithm processes all of the training data in a single epoch before any weights are adjusted [76].

8. Training Parameters

Training in deep learning can be enhanced in terms of accuracy and training time by modifying a number of parameters. The following are examples of training variables:

a) Batch Size

The amount of time it takes to train a model depends on the batch size, or how many datasets are supplied to the network during each iteration. If the complete dataset is given to the network during each iteration, the training time may be lowered. The accuracy of the network will suffer if it is made too generalized with regard to the dataset [77, 78].

b) Epoch

If the dataset has 1100 rows and the batch size is 100, then one epoch will require 11 iterations to finish feeding the full dataset into the network. Epochs allow for the recycling of previously used training data sets.

2.8.1.1. CNN Architecture

In face expression recognition, 2D CNNs with time distribution efficiently capture spatial and temporal information from video sequences. 2D CNNs with time distribution differ from image-based CNNs by include the temporal dimension by using a sequence of frames as input [79]. The two-dimensional convolutional neural network (CNN) can learn spatiotemporal patterns and reliably record changing face expressions by adding time distribution. 2D convolutions in the spatial (height, width) and temporal (frames) axes of the input video achieve this.

The time distribution lets the network exploit temporal relationships in face expression sequences to completely understand expression dynamics. This lets the model track expression evolution. A 2D CNN with temporal distribution typically has many convolutional layers followed by pooling layers for spatial

down sampling. These layers develop hierarchical face expression representations at various abstractions. Next, fully linked layers with softmax activation classify[80].

A CNN's input layer reflects the model's input (explicitly specified features) and is independent of the network's size. When processing gene expression data, a convolutional neural networks (CNN) input layer typically consists of a two-dimensional (n m) matrix, where n is the sample size and m is the number of features [81, 82]. Figure shows a neural network's layers (2.6). Levels are:

a) Convolutional layer.
b) Max pooling layer (or Sub Sampling layer).
c) Fully Connected Layer (Classification layer).
d) Dropout Layer

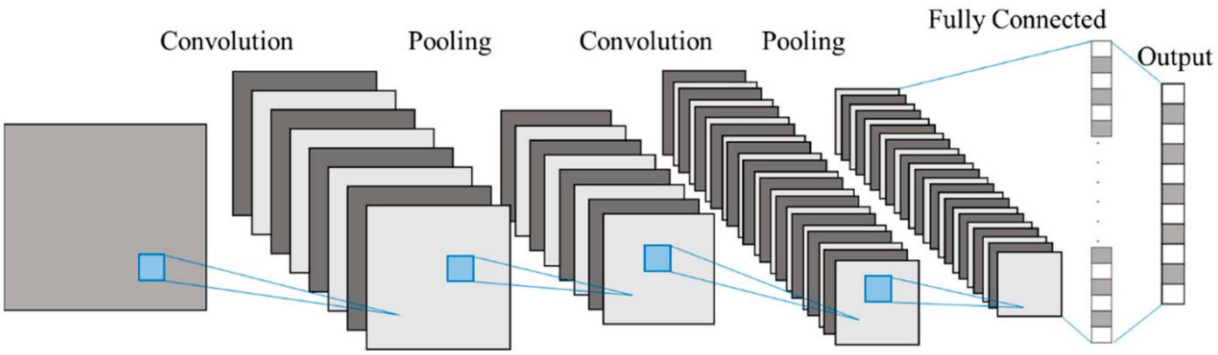


Figure 2.6: Example of a 2D-CNN Structure[82].

a) Convolutional layer

The convolution layer is the foundation of the CNN architecture and can process data in high dimensions. The convolution layer is the first layer of a convolutional neural network and it is partially connected to the second layer (the pooling layer). For example, a window of input neurons (3x3)

will be connected to the pooling layer, and this window moves forward in the data from the top left corner to the bottom right corner at regular intervals (the "stride" value, which is typically 1).The kernel shifts right by one cell until it reaches the end of the columns, then shifts down by one cell until it reaches the end of the rows, and so on, until all the information has been recorded[83].

The receptive field is a tiny window region created from the input data. To extract features, a tiny section of the input data will be convolved with a shared weights window called kernel or filter . Figure (2.7) depicts the convolution process.

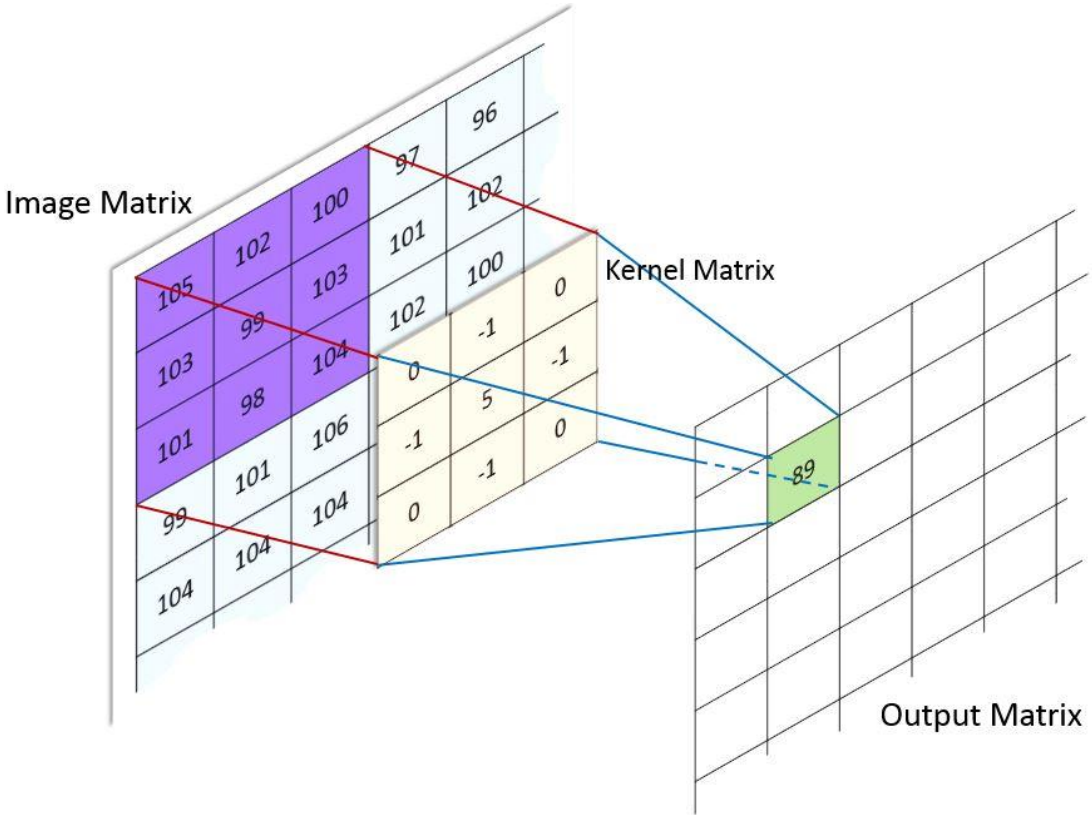


Figure 2.7: Input image after being processed with a Convolution filter [69].

A single entry is wrapped with many different filters. In the convolutional layer, the activation maps are merged to produce a single output file, which serves as the input data for the subsequent layer. The default weights

accurately represent the values in the filter matrix. Each filter needs to have unique values for these parameters in order to endow its output matrices with distinctive characteristics or features [84].

Furthermore, a Convolutional Neural Network's (CNN's) architecture has several parameters that are used to regulate things like the model's behavior, the size of its output, and how long it takes to operate (hyperparameters). These hyperparameters are crucial for convolutional neural networks (CNN):

1. **Number of filters:** It is possible to utilize a wide variety of filters, each of which comes in a slightly varied size.
2. **Filter size:** The filter or kernel size, expressed as L2 or L2, must be less than or equal to the X2 size of the input data.
3. **Stride:** Number of cells that must be shifted simultaneously to produce a filter's local receptive field. In one fluid motion, a single cell travel both horizontally and vertically. Overlap occurs when the stride is too short, and vice versa.
4. **Padding :**In order to improve accuracy, the CNN architecture incorporates the idea of padding. In order to regulate the contraction of the convolutional layer's output, padding is used.

The convolutional layer produces a feature map that is much more compact than the original image. The produced feature map emphasizes central pixels and thus downscales the prominence of information at the map's periphery. To stop the feature map from getting too small, blank rows and columns are appended to the image's borders. When determining the size of the output feature map, the equations (2.17) and (2.18) define the relationship between the feature map size, the kernel size, and the stride[85].

$$W_{nx} = W_{(n-1x)} - F_{nx} . S_{nx} + 1 \quad (2.17)$$

$$W_{ny} = W_{(n-1y)} - F_{ny} . S_{ny} + 1 \quad (2.18)$$

Equation (2.17) calculates the width (W_{nx}) of the output feature map at layer n. It does this by taking the width of the input feature map at layer $n-1$ (W_{n-1x}), subtracting the width of the kernel (F_{nx}) multiplied by the stride (S_{nx}), and then adding 1. In essence, this equation computes how much the kernel "slides" across the width of the input, taking into account the kernel size and the step size determined by the stride. The "+1" accounts for the starting position of the kernel.

Equation (2.18) is analogous to Equation (2.17), but it calculates the height (W_{ny}) of the output feature map at layer n based on the height of the input feature map at layer $n-1$ (W_{n-1y}), the height of the kernel (F_{ny}), and the vertical stride (S_{ny}).

b) Max pooling layer or Sub Sampling layer

CNN achieves its outcomes via a combination of convolution and pooling layers. This layer's major goal is to produce reduced-dimensional output by reducing the input dimensions while keeping the most crucial information. This layer uses maximum and average pooling to accomplish its dimensionality reduction. When in max or average pooling mode [86], the pooling layer divides the input feature map into non-overlapping blocks and returns a single value for each block. Maximal pooling is shown in Figure (2.8).

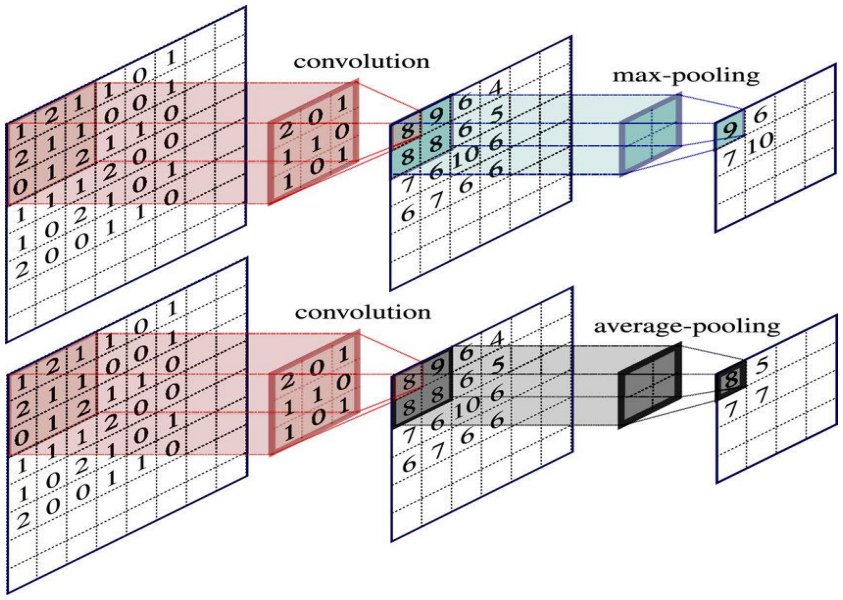


Figure 2.8: Max and Average Pooling Patterns[87].

The maximum pooling layer performs the same operation on each of the feature maps that were previously produced from the preceding convolution layer.

c) Fully Connected Layer (Classification layer) (FC).

As the final layer of the network, a fully connected layer (FC), also commonly referred to as a dense layer, is present. A fully linked layer is one in which every neuron from the layer below it is coupled to every neuron in the layer above it. Before the resulted feature map from the previous layer can be fully connected with the output layer, it must first be flattened into the form of a feature vector. The output layer is comprised of neurons that are equal in number to the number of classes according to the softmax or sigmoid activation functions that are used in the final CNN layer to classify the trained data. These activation functions are optimized for multi-class and binary class classification, respectively [88]. Figure (2.9) presents a visual representation of the relationship that exists between completed feature maps and a layer that is fully connected.

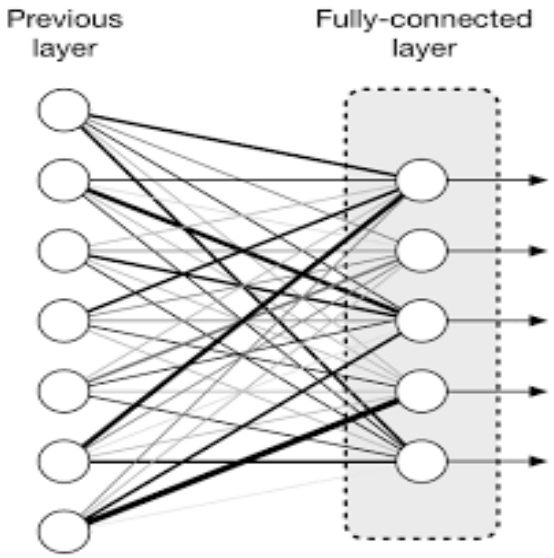


Figure 2.9: Connection Between convolution layer and Fully Connected Layer [88] .

d) Dropout Layer

Overfitting the training dataset occurs when all features are connected to the fully connected layer. When a model does exceptionally well on training data, but then struggles to generalize to novel data, this phenomenon is known as overfitting. To address this issue, a dropout layer is used to remove neurons and their associated connections from the network at random during training[61]. The procedure of dropping out is shown in Figure (2.10).

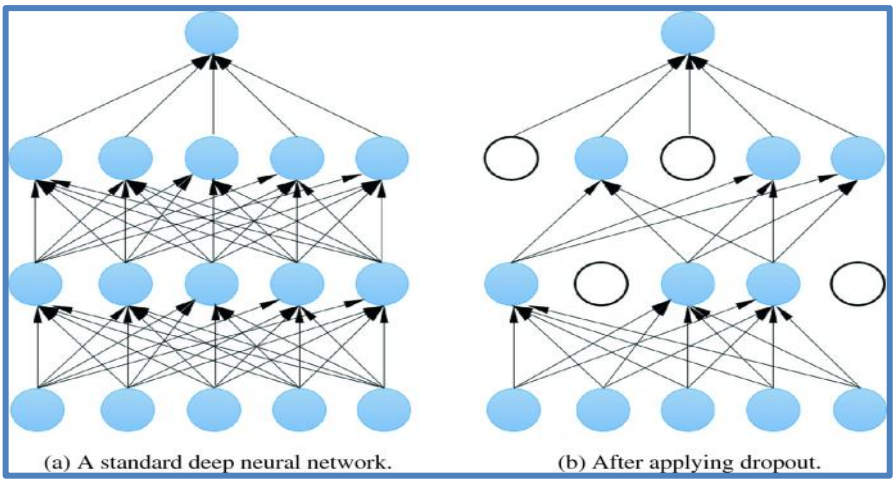


Figure 2.10 : Schematic comparison of (a) a regular neural network and (b) a neural network trained with Dropout[89]

Using 2D CNNs with time distribution has the benefit of allowing for more accurate and robust recognition of facial expressions in video sequences because of its capacity to capture both spatial and temporal cues. In the realm of face expression analysis, this method has been widely used since it outperforms those that focus just on the spatial information of individual frames. Facial expression identification relies on modelling spatiotemporal patterns, and 2D CNNs with time distribution provide a powerful tool for doing so. These models are able to comprehend the dynamic character of expressions and increase the precision of emotion categorization in video data by include the temporal dimension in the network architecture.

2.9 Time Distributed layer

The Time Distributed layer is a crucial component in deep learning architectures for handling sequential data. It is specifically designed to process sequential information in recurrent neural networks (RNNs) or convolutional neural networks (CNNs) [90]. The main purpose of the Time Distributed layer is to apply a specific layer or operation to each time step or frame of the input sequence independently. The Time Distributed layer plays a vital role in capturing temporal dependencies and patterns within a sequence. By applying a layer or operation to each time step individually, the model can effectively learn and understand the sequential nature of the data [91]. The algorithm for the Time Distributed layer can vary depending on the specific layer or operation being used. For instance, let's consider an example where we have an input sequence with the shape (batch_size, sequence_length, input_dim), and aim to apply a fully connected layer independently to each time step[92]. The algorithm for the Time Distributed layer would be as follows:

Algorithm (2.4): Time Distributed Algorithm

Input: Sequence of inputs X of shape (batch_size, time_steps, input_dim)

Output: Collection of outputs Y of shape (batch_size, time_steps, output_dim)

BEGIN

Step 1: Initialize an empty collection of outputs Y.

Step 2: For each time step t in the range from 0 to time_steps-1:

- Select the input at time step t from the sequence X[:, t, :] of shape (batch_size, input_dim).

Step 3: Apply the designated layer or operation to the selected input at each time step:

- Pass the input at time step t through the designated layer or operation.
- Obtain the output O_t of shape (batch_size, output_dim) at time step t.

Step 4: Collect the output at each time step and append it to the collection of outputs Y:

- Append the output O_t to the collection Y.

Step 5: Repeat steps 2-4 for each time step in the range from 0 to time_steps-1.

Step 6: Return the collection of outputs Y as the final output of the TimeDistributed layer.

END

By utilizing this algorithm, the model can effectively capture temporal patterns and dependencies within the sequence. The Time Distributed layer is commonly employed in tasks such as sequence classification, sequence-to-sequence prediction, and video analysis, where individual time steps or frames require separate processing to comprehend the sequential information effectively.

2.10. Performance Evaluation

The classification precision of the model was calculated at this stage. The performance of the classifier was evaluated by comparing the actual to the predicted labels for each category. Accuracy can be determined by comparing the percentage of correctly categorized cases (true positives) against the percentage of correctly classified but irrelevant examples (true negatives), correctly

classified but inaccurate examples (false positives), and unclassified instances (false negatives)[93 - 94].

2.10.1 Confusion Matrix:

Classification algorithms can be evaluated scientifically using a number of different metrics. Accuracy, f1-score, precision, and recall are all examples. These metrics are derived from a confusion matrix, which is a table summarizing the proportion of correct and incorrect predictions made by a specific classification model (see Figure (2.11)). Detailed explanations of the table's values follow below[88]:

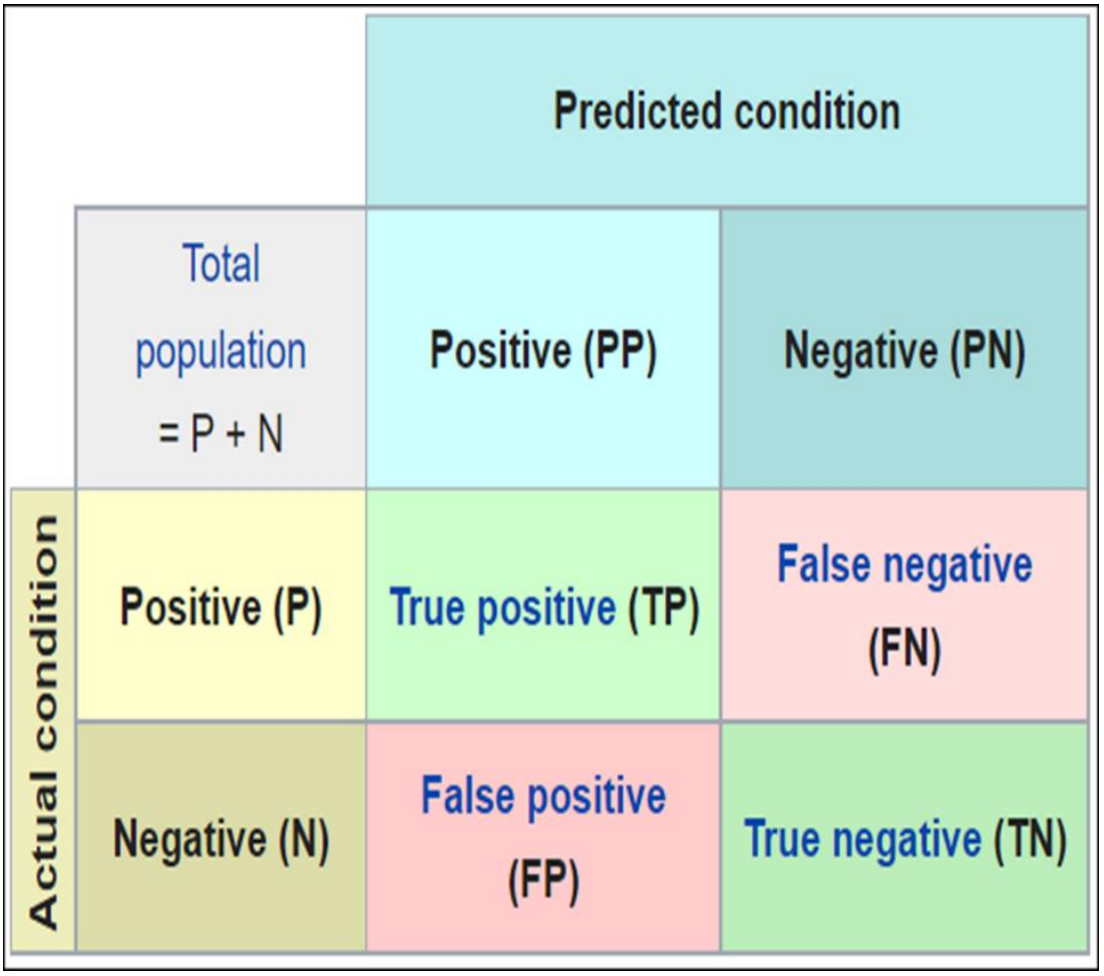


Figure 2.11: Confusion matrix [88].

2.10.2 Performance Metrices

Metrics that measure performance are crucial for figuring out how efficient and trustworthy a given system, algorithm, or model is. They provide us a numerical basis for judging performance and making educated choices. These measurements shed light on how precise, effective, and high-quality a process or system is. Performance metrics let us to objectively evaluate and contrast several solutions, allowing us to pick the best one for a given job by assessing specific parameters such as accuracy, precision, recall, and F1 score.

- **Accuracy:** As demonstrated in Eq. (2.19), accuracy can be defined as the fraction of predictions that turn out to be correct relative to the total number of forecasts[94].

$$Accurcy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.19}$$

- **Precision :** The precision of a document set is measured by how well it describes its topic and, by extension, how exactly the documents were categorized. As demonstrated in Eq. (2.20), the precision of the class ci denoted by the symbol (Pi) is as follows [94]:

$$P_i = \frac{TP_i}{TP_i+FP_i} \tag{2.20}$$

- **Recall:** To what extent a classifier is able to correctly identify documents as belonging to a particular class is quantified by recall (as shown by Eq (2.21). The formula for class ci recall (Ri) is [94]:

$$R_i = \frac{TP_i}{TP_i+ FN_i} \tag{2.21}$$

In this case, TP_i points to a true-positive value. FP_i stands for false positives and FN_i represents false negatives.

- **F1-score:** The correlation between precision and recall is denoted by F1. If F1 is high, the performance of the whole system improves. The following is a description of function F1 given Eqs. (2.22) and (2.23) [94]:

$$\boldsymbol{F1} = \frac{\boldsymbol{2 \times precision \times recall}}{\boldsymbol{precision + recall}} \tag{2.22}$$

$$= \frac{\boldsymbol{2TP}}{\boldsymbol{2TP + FP + FN}} \tag{2.23}$$

Chapter Three
The Proposed System

Chapter Three

The Proposed System

3.1. Introduction

This chapter presents a summary of the proposed system that uses deep learning techniques to recognize facial expressions to establish a person's attitude and then tailors recommendations from the Quran to that mood. The system is comprised of multiple stages, including as Dataset, preprocessing, feature extraction, and classification are just few of the processes in the system. These phases can be used with still images or video clips.

3.2 Proposed System

The method proposed in this thesis employs a systematic approach, as illustrated in Figure (3.1) , to efficiently analyze image frames and videos and precisely classify facial expressions. The workflow is partitioned into two distinct models: a one-dimensional convolutional neural network (1D CNN) and a two-dimensional convolutional neural network (2D CNN) . The first stage of the process focuses on pre-processing processes designed to improve the quality and interpretability of the supplied data. Following the feature extraction step, the use of the 1D CNN model is implemented, capitalizing on its ability to proficiently categorize emotions in video sequences. Furthermore, the utilization of a 2D Convolutional Neural Network (CNN) model enables the concurrent extraction of features and classification, hence facilitating a thorough assessment and efficient categorization of emotions depicted in videos. The suggested technique employs two models to compare the accuracy of various models in the classification of facial emotions and subsequently select Quranic verses from the higher model accuracy on the classified emotions., as described in Algorithm (3.1) and (3.2).

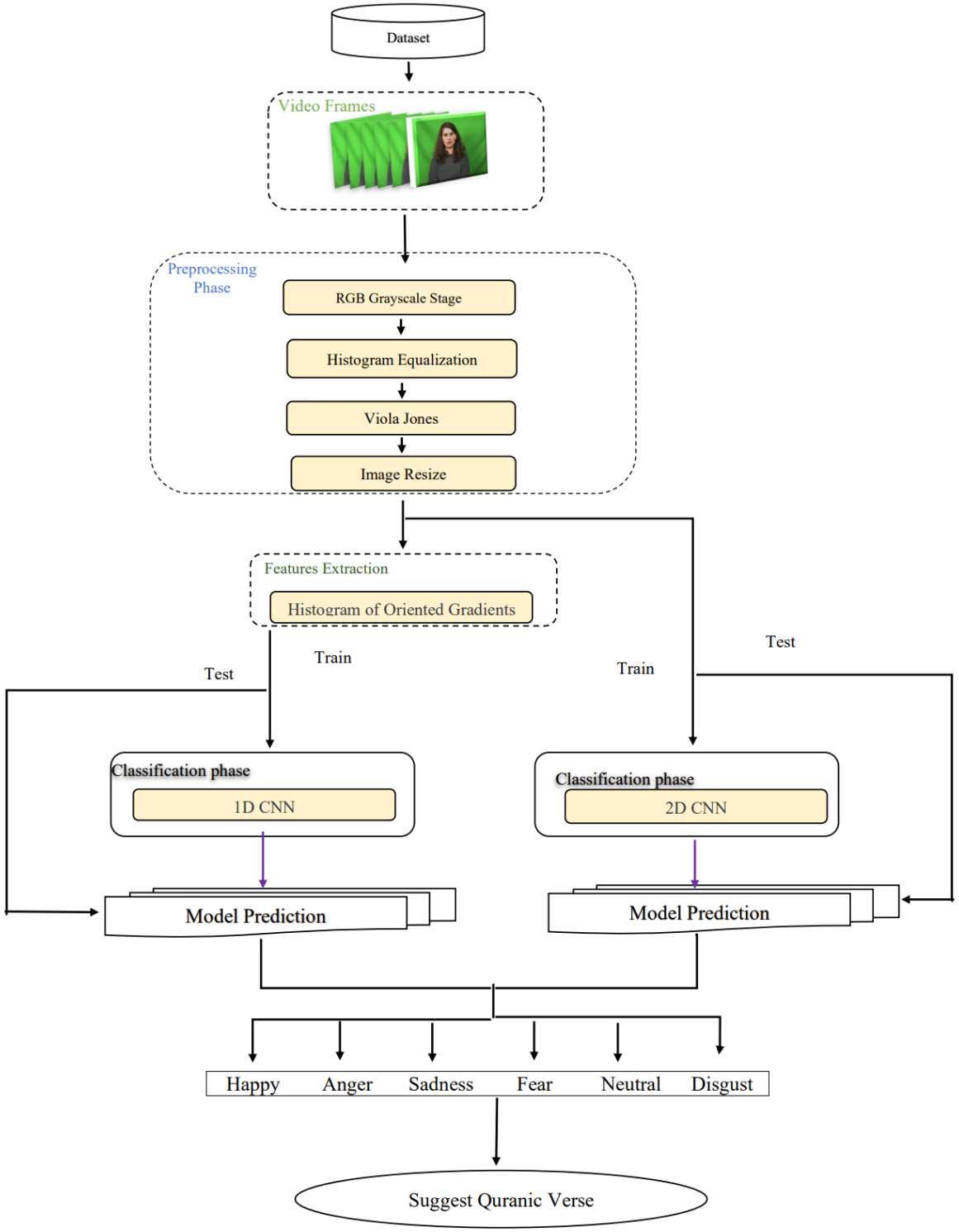


Figure 3.1 : Proposed System.

Algorithm (3.1): 1D CNN Model

Input: Video sequence or Image frames

Output: Predicted facial expression category

BEGIN

Step 1: Insert Dataset

Step2:Preprocessing

- Apply preprocessing steps to enhance the quality of the input video sequence or image frames (RGB to Gray Conversion).
- Apply Histogram Equalization using algorithm (2.1).
- Apply Viola and Jones using algorithm (2.2).
- Apply Resizing using Eq. (2.6).

Step 3:HOG Feature Extraction + 1D CNN Model

- For each frame in the video sequence or image frames:
  - Apply the HOG algorithm to extract hand-crafted features from the frame using algorithm (2.3).
  - Store the extracted HOG features in a feature vector for each frame.
- Construct the input data for the 1D CNN model using the feature vectors obtained from the HOG algorithm.
- enter the input data into the 1D CNN model.
- The 1D CNN model performs classification based on the learned features, predicting the facial expression category for each frame.

Step 4: 1D CNN model : Once the model has classified an emotion, it chooses appropriate verses from the Quran that relate to that feeling.

End of Algorithm.

Algorithm (3.2): 2D CNN Model

Input: Video sequence or Image frames

Output: Predicted facial expression category

BEGIN

Step 1: Insert Dataset

Step2:Preprocessing

- Apply preprocessing steps to enhance the quality of the input video sequence or image frames (RGB to Gray Conversion).
- Apply Histogram Equalization using algorithm (2.1).
- Apply Viola and Jones using algorithm (2.2).
- Apply Resizing using Eq. (2.6).

Step 3: 2D CNN Model

- For each frame in the video sequence or image frames:
  - Apply preprocessing steps to the 2D CNN model (e.g., resizing to the desired input size for the 2D CNN).
- Construct the input data for the 2D CNN model using the preprocessed frames.
  - Use the time-distributed layer to feed the input data into the 2D CNN model as a sequence of frames.
  - The 2D CNN model applies a series of convolutional layers to extract spatial patterns and hierarchical representations from the raw image data.
- The 2D CNN model performs classification based on the learned features, predicting the facial expression category for each frame.

Step 4:The 2D CNN model's suggests Quranic verses to the identified emotions after producing results in emotion classification.

End of Algorithm.

The system utilizes MP3 audio recordings to propose Quranic verses that correspond to the identified emotions. The utilization of both the 1D CNN and 2D CNN models in the proposed approach facilitates precise identification of facial expressions. Additionally, the incorporation of Quranic verses based on the user's emotions enhances the system, offering a distinctive and spiritually advantageous encounter for users.

3.2.1 Dataset Characterization

The Crema-D dataset, which is a Crowd-sourced Emotional Multi-modal Actors Dataset, comprises a total of 7442 clips. These clips were produced by theater directors and feature 91 actors, consisting of 48 males and 43 females. The actors' ages range from 20 to 74, and they represent diverse ethnicities.

3.2.2 Video Processing and Capturing

Every video or animation is composed of a series of static images. Subsequently, these images are sequentially displayed multiple times per second, thereby deceiving the visual perception into perceiving the object as being in motion. The video stream achieved a frame rate of approximately 30 frames per second, with each discrete image referred to as a frame. One of the essential prerequisites of this methodology entails the provision of a suitable environment for accessing the videos.

3.2.3 Preprocessing Step

Preprocessing image frames is crucial for analysis. This phase includes essential efforts to improve visual information quality and processing appropriateness. To enhance the incoming video sequence or picture frames, preparation begins by converting RGB photos to grayscale simplifies analysis by removing color information while keeping facial expression intensity fluctuations. Next, histogram equalization, improves pixel contrast and distribution, improving visual cues that aid emotion identification.

A standard scaler normalizes pixel values throughout the dataset. This stage provides a consistent data distribution and allows following models to learn from the input data without scale biases. Face identification and alignment are improved by using the Viola and Jones method . This method efficiently recognizes face areas in photos, leading to feature extraction and emotion categorization. To standardize picture dimensions, requires image resizing. This approach simplifies picture

comparison and feature extraction and preserves face expressions as image size changes.

3.2.3.1 RGB to Grayscale Conversion

The conversion from RGB to grayscale is a key stage in the image preparation process. This entails turning the color photos into the grayscale images, which simplifies the subsequent processing processes while maintaining the relevant facial expression information. The red, green, and blue color channels of the original image are taken into consideration throughout the conversion process, and weights are allotted to each channel so that a single intensity value may be obtained for each pixel. The color information in the image is lost during this conversion, but the overall structure and brightness of the picture is maintained. Grayscale photographs are frequently utilized in facial expression analysis because they decrease the complexity of the data and concentrate attention on the intensity fluctuations that occur in various facial regions.

3.2.3.2 Histogram Equalization

The histogram is then equalized to improve the contrast and details in the localized facial regions that were produced in the prior stage. The visibility of significant features can be improved with this strategy, letting such features stand out more clearly for later examination.

The contrast of an image can be improved by using a technique called histogram equalization, which involves redistributing the pixel intensities in the image across a larger range. This is accomplished by modifying the histogram of the image in order to obtain a more even distribution of intensities throughout the picture. The technique works to maximize the visibility of facial expression cues including wrinkles, contours, and other small variations in the facial areas. This is accomplished by equalizing the histogram. This step of preprocessing increases the quality of the

image frames as well as their capacity for discrimination in preparation for subsequent feature extraction and mood categorization.

The histogram equalization technique is employed when digital facial images exhibit low contrast values, such as inadequate allocation of image lighting or poor illumination. After converting the inputted colored images to grayscale, the resulting gray image's contrast is enhanced using the cumulative histogram equalization technique. This technique is chosen due to its effective performance in histogram equalization. The implementation of this stage of image preprocessing follows the steps outlined in section (2.5.1).

3.2.3.3 Viola-Jones Face Detection:

Within the grayscale images, facial regions are detected and localized with the help of the Viola-Jones method, which is utilized by the suggested system. This stage is essential for the feature extraction and categorization that will take place later. The efficient detection of faces in images is accomplished by the Viola-Jones algorithm, which makes use of Haar-like features and a learning framework based on AdaBoost. Developing a powerful face classifier requires training a hierarchy of somewhat ineffective classifiers using a substantial amount of both positive and negative examples. When the grayscale image frames are processed, the trained cascade is applied to them in order to accurately detect the presence of faces. Calculating the integral image requires only one pass through the original image. To compute this integral image serves as important means of effective feature computation. Four different metrics are used to refer to this region which includes Region D. These references correspond to specific points on the integral image: points 1, 2, and 3. Amplitude differences, i.e., integration of the image points for a period of time, enable accurate determination of pixels of interests. This involves significant regions including (A+B), (A+C) and A. Taken together, these actions help to determine the count of all pixels of region D. However, for some tasks such as face detection one

must have very quick and precise calculations, therefore, this method becomes especially helpful.

Haar-like characteristics are calculated from the Haar wavelet transform, which breaks down an image into wavelet patterns. The designs are rectangular with dark and bright sections. Calculating the difference between dark and bright pixel intensities creates Haar-like characteristics. The system can recognize face traits by applying these attributes to picture areas.

The method slides rectangular filters across the picture and studies intensity differences to build Haar-like features. Two-rectangle, three-rectangle, and four-rectangle patterns provide versatile picture data processing. Each feature is a visual pattern, and their existence or absence in an image can reveal its content. The Haar-like properties offer a complete set of templates for rectangular filter setups. These templates help the program scan the picture at various sizes and places. The system can recognize face characteristics in the picture by calculating their responses at different locations and sizes. A strong technique, Haar-like features collecting, scans the image quickly in the Viola-Jones face identification procedure to discover prospective face locations.

3.2.3.4 Image Resizing:

Resizing the images is a crucial step in the preparation phase for image frames in the system that has been developed. The facial photographs are then scaled to a set dimension following the face detection and histogram equalization processes. The photos should be resized to a constant size before continuing, as this will assure uniformity and simplify the remaining processing procedures. The fixed dimension makes it possible to compare and extract features from multiple photos with greater ease. In addition, reducing the size of the photos by shrinking them to a smaller size lowers the amount of computational complexity and memory needs, which makes the future phases of feature extraction and mood categorization more effective.

The proposed system prepares the image frames for further analysis by resizing them to a standardized dimension before doing so. This ensures consistency and makes it easier to extract meaningful features that can detect facial expressions and moods by capturing the essential information related to those areas of the face.

3.2.4 Features Extraction and Classification Step

The proposed model comprises two discrete models, namely the HOG (Hand-crafted Features) + 1D CNN model and the 2D CNN model. Each model assumes responsibility for distinct stages within the comprehensive workflow, encompassing feature extraction and classification.

The initial step in the first model involves feature extraction utilizing the HOG (Hand-crafted Features) algorithm, followed by the application of a 1D CNN. The algorithm employed in this thesis effectively captures and represents the local gradient patterns present in the input image. These patterns are subsequently converted into a feature vector for further analysis and processing. The features obtained from the Histogram of Oriented Gradients (HOG) algorithm are subsequently inputted into a one-dimensional Convolutional Neural Network (CNN) model. The architectural design of the 1D Convolutional Neural Network (CNN) is carefully constructed to effectively analyze and interpret the facial expressions in video sequences. This particular architecture enables the model to accurately perceive and understand the dynamic changes in facial expressions that occur over a period of time. The system utilizes the retrieved characteristics as input and proceeds with the classification job, intelligently assigning the input sequences to their corresponding facial expression categories.

In the second model, the 2D Convolutional Neural Network (CNN), the process of feature extraction and classification is integrated. The proposed model utilizes input images and applies a sequence of frames by using TimeDistributed to convolutional layers to extract spatial patterns and hierarchical representations of facial expressions. The 2D convolutional neural network (CNN) model effectively

leverages its inherent capacity to acquire discriminative features from unprocessed image data in order to extract pertinent information. Subsequently, the system proceeds to conduct classification utilizing the acquired features, effectively assigning appropriate categories to the facial expressions depicted in the input images.

The proposed method effectively tackles the feature extraction and classification stages of facial expression recognition through the utilization of these two models. After feature extraction using HOG then send features to 1D CNN model utilizes manually designed features and temporal analysis, whereas the 2D CNN model directly extracts features from processed image data. The utilization of a two model approach allows the system to proficiently encompass temporal and spatial attributes, resulting in precise and resilient classification of facial expressions.

3.2.4.1 Histogram of Oriented Gradients (HOG) Features Extraction Step

The Histogram of Oriented Gradients (HOG) technique is used in the system that has been proposed in order to extract features from the video frames. The HOG technique records the distribution of gradients within the localized facial areas, which are key cues for understanding facial expressions. These gradients serve as essential cues since they vary from region to region on the face. In order for the HOG method to function, the facial region is first segmented into smaller cells, and then the gradient magnitude and orientation within each cell is calculated. The gradient orientations are then arranged in a histogram, with the histogram bins being determined by the magnitude of the gradients as shown in Figure (3.2). The HOG method is able to capture local patterns of edge directions and gradients that are characteristic of various face expressions as a result of this .

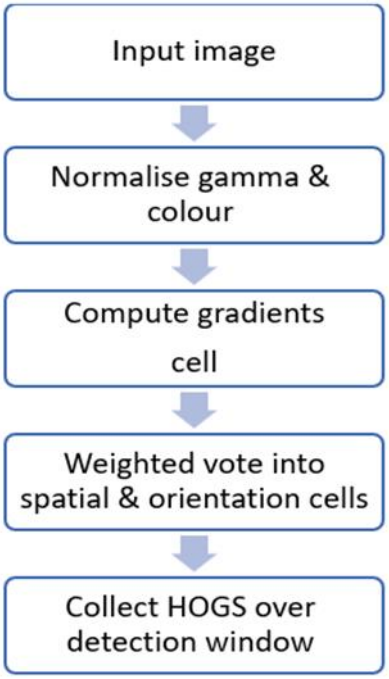


Figure 3.2: Histogram of Oriented Gradients Algorithm's.

The information on the shape and texture of the facial areas in the video frames is efficiently captured by the suggested method, which does this by extracting HOG features from the frames. Different facial expressions reveal varied patterns and arrangements of edges and gradients, and these traits serve a significant part in differentiating between the many expressions that can be found on the human face. In order to facilitate future categorization and mood detection, the HOG features offer a condensed and informative representation of the localized face regions.

3.2.4.2 Convolutional Neural Network (1D CNN) Classification Stage

The suggested system for facial expression recognition and mood detection relies heavily on the 1D Convolutional Neural Network (1D CNN). In-depth discussion of the 1D CNN model's layers and the forms their outputs take is provided here. In the first layer, denoted 'conv1d 1,' a 1D convolution is carried out, with the following output shape: (None, 574, 16). To the input data, it applies 16 filters, each with a size of 3x3. With a pool size of 2x2, the next layer, "max pooling1d 1,"

completes max pooling to get an output shape of (None, 574, 16). By picking the highest value in each pooling region, max pooling minimizes the input's dimensionality in space.

In the 'conv1d 2' layer, another one-dimensional convolution is applied, this time using 32 filters with a size of 3x3. This operation results in an output shape of (None, 572, 32). The same output shape is maintained in the subsequent 'max pooling1d 2' layer by applying max pooling. This pattern of convolution followed by max pooling is repeated in the subsequent layers ('conv1d 3', 'max pooling1d 3', 'max pooling1d 4', 'conv1d 4', 'max pooling1d 5', 'conv1d 5', 'max pooling1d 6', 'conv1d 6', 'max pooling1d 7', 'conv1d 7', 'conv1d 8', 'max pooling1d 8'). These layers progressively reduce the spatial dimensions of the data while increasing the number of filters. This process helps capture and extract higher-level features from the input data, allowing the network to learn and represent complex patterns in the data.

The 'dense 1' layer is a completely linked 128-neuron layer that preserves the input-to-output mapping. Subsequent "dense 2," "dense 3," and "dense 4" layers are all fully linked and feature 64, 64, and 16 neurons, respectively. Another 1D convolution operation is carried out by the 'conv1d 9' layer, this time using 55 filters to provide an output shape of (None, 560, 55). This layer is responsible for capturing finer patterns and adding additional nuance to the feature representation.

The input is transformed into a one-dimensional vector of shape (None, 30800) via the 'flatten 1' layer. Six neurons, one for each of the six mood categories, are represented in the final layer ('dense 5'), which acts as the output layer. This layer's output is shaped like (None, 6).

The 1D CNN model has a total of 247,037parameters, all of which can be modified by training. During the training phase, these settings are fine-tuned to increase the model's precision when classifying facial expressions. In order to learn complicated patterns and collect discriminative characteristics from the preprocessed

facial images, the 1D CNN classification phase makes use of the capabilities of convolutional operations, max pooling, and fully connected layers. The model's ability to extract hierarchical representations and make predictions based on the observed mood is made possible by the sequential arrangement of layers.

Table 3.1: Representation of the layers in the 1D Convolutional Neural Network (1D CNN) model.

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 574, 16)	64
max_pooling1d_1	(None, 574, 16)	0
conv1d_2 (Conv1D)	(None, 572, 32)	1568
max_pooling1d_2	(None, 572, 32)	0
conv1d_3 (Conv1D)	(None, 570, 64)	6208
max_pooling1d_3	(None, 570, 64)	0
conv1d_4 (Conv1D)	(None, 568, 32)	6176
max_pooling1d_4	(None, 568, 32)	0
conv1d_5 (Conv1D)	(None, 566, 64)	6208
max_pooling1d_5	(None, 566, 64)	0
conv1d_6 (Conv1D)	(None, 564, 64)	12352
max_pooling1d_6	(None, 564, 64)	0
conv1d_7 (Conv1D)	(None, 562, 32)	6176
conv1d_8 (Conv1D)	(None, 560, 32)	3104
max_pooling1d_7	(None, 560, 32)	0
dense_1 (Dense)	(None, 560, 128)	4224
dense_2 (Dense)	(None, 560, 64)	8256
dense_3 (Dense)	(None, 560, 64)	4160
dense_4 (Dense)	(None, 560, 16)	1040

conv1d_9 (Conv1D)	(None, 560, 55)	2695
flatten_1 (Flatten)	(None, 30800)	0
dense_5 (Dense)	(None, 6)	184806

The table provides a clear overview of each layer's type, output shape, and the number of trainable parameters (param #). It helps visualize the architecture and the flow of data through the network.

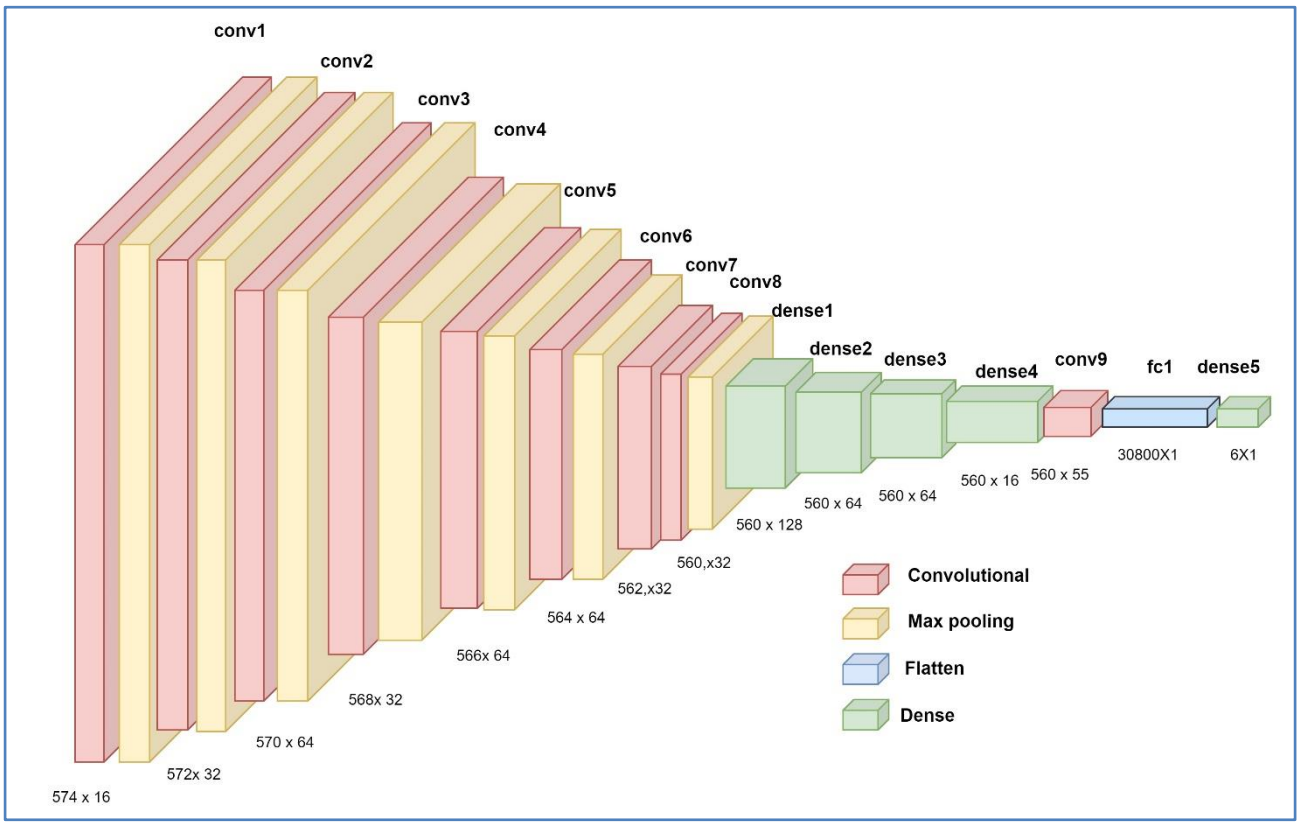


Figure 3.3: 1D CNN classification model layers.

3.2.4.3 Convolutional Neural Network (2D CNN)

The first phase in the process of emotion classification entails employing a two-dimensional convolutional neural network (CNN) to simultaneously carry out feature extraction and classification. The 2D convolutional neural network (CNN) is of special significance because to its capacity to capture the feature emotions in consecutive frames. Through the integration of spatial and temporal information, the

utilization of a two-dimensional convolutional neural network (2D CNN) facilitates the classification of facial expressions, thus enabling the precise identification of emotions. The model's dual functionality enables it to extract prominent characteristics and smoothly transition into the classification phase. This methodology effectively leverages the inherent benefits of deep learning models, enabling the assessment of the complex patterns that are unique to face emotions. As a result, this approach effectively captures the subtle changes in emotions displayed through facial expressions, thereby providing a strong foundation for reliable and precise emotion classification.

Table 3.2: Representation of the layers in the 2D Convolutional Neural Network (2D CNN) model.

Layer (type)	Output Shape	Param #
time_distributed_11	(None, 6, 25, 25, 64)	640
time_distributed_12	(None, 6, 13, 13, 64)	36928
time_distributed_13	(None, 6, 6, 6, 64)	0
time_distributed_14	(None, 6, 3, 3, 128)	73856
time_distributed_15	(None, 6, 2, 2, 128)	147584
time_distributed_16	(None, 6, 1, 1, 128)	0
flatten_1 (Flatten)	(None, 768)	0
dense_3 (Dense)	(None, 1024)	787456
dense_4 (Dense)	(None, 512)	524800
dropout_1 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 6)	3078

The provided layers constitute a deep learning model architecture with a TimeDistributed layer applied to handle sequential data. The input to the model is a tensor with a shape of (None, 6, 25, 25, 64). This tensor represents a sequence of six frames, each of size 25x25 pixels and with 64 channels. The TimeDistributed layers play a crucial role in processing the sequential data.

Then comes the next layer which successively reduces the size of the data, transforming it into new shapes gradually. The first is (None, 6, 13, 13, 64), the second is (None, 6, 6, 6, 64), the third is (None, 6, 3, 3, 128), This helps represent the changes happening at different stages in the sequence of frame without consuming a significant amount of computational power. The output from the final TimeDistributed layer is then flattened to a 1D tensor of shape (None, 768). This flattened representation is then passed through two fully connected layers, with dimensions (None, 1024) and (None, 512) respectively. The Dropout layer is applied after the second dense layer to mitigate overfitting.

Overall, this architecture leverages the TimeDistributed layers to effectively process and extract temporal features from the sequential data, followed by dense layers for classification. The model learns to capture the temporal dynamics and patterns within the input sequence, enabling accurate emotion recognition.

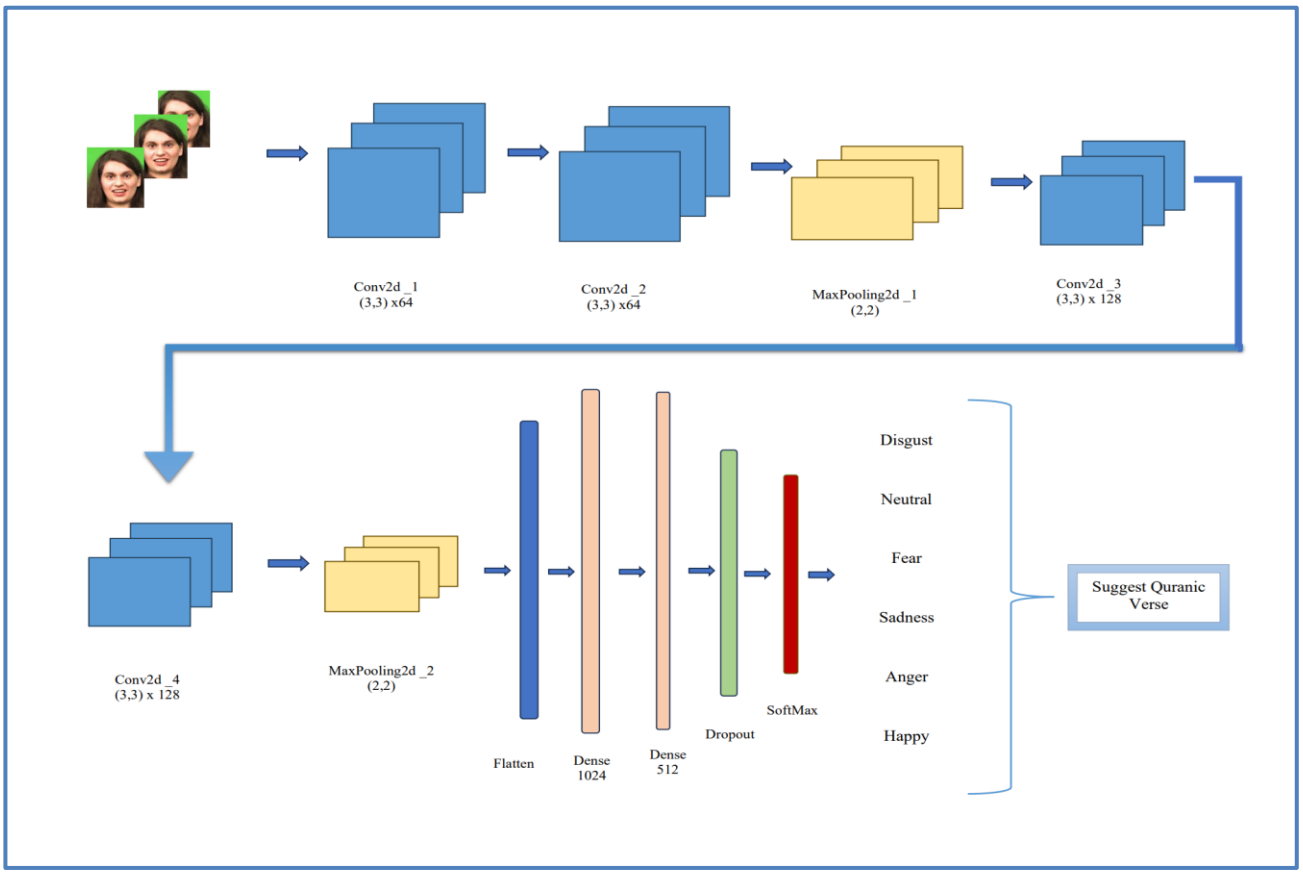


Figure 3.4: 2D CNN model .

3.2.5 Quranic Verses Based on Emotions

In addition to the facial detection and emotion recognition functionalities of the proposed system, also present a distinctive attribute wherein Quranic verses are recommended based on the identified emotions. Through the process of associating distinct Quranic verses with various emotions, individuals have the opportunity to access pertinent verses that can potentially provide comfort, motivation, or introspection.

The audio files are stored in the MP3 format to facilitate convenient retrieval and playback. When the system identifies a particular emotion based on the user's facial expression, it proceeds to query the database containing Quranic verses that are linked to that specific emotion. The proposed model subsequently chooses the Quranic verse that is most pertinent, taking into account the identified emotion. The chosen verse is delivered to the user via an auditory playback, enabling them to engage in active listening and contemplate the verse's inherent meaning and significance. The primary objective of this feature is to offer emotional and spiritual assistance to individuals experiencing various emotional conditions.

Quranic verses are integrated into the system to use technology to deliver emotional recognition, face identification, and spiritual direction. Adding Quranic verses to the proposed system makes emotional counseling unique. The website allows people to practice their religion, take consolation in the Quran, and heal emotionally by reciting sacred verses.

CHAPTER FOUR
EXPERIMENTAL RESULTS AND DISCUSSION

Chapter Four

Experimental Results and Discussion

4.1 Introduction

The steps and processes involved in the proposed emotion recognition system were explored in detail in the preceding chapter. The results and performance of the proposed system are discussed in this chapter. The system is split into two steps, the first of which uses HOG to extract characteristics from videos before classifying them with deep learning algorithms, and the second of which uses the convolutional neural network CNN algorithm to enter the videos directly to it for classification. These are the main points that are elaborated upon throughout this chapter.

4.2. Hardware and Software Requirements

To successfully carry out the process of designing, implementation, training, and testing the system, particular software and hardware requirements must be met. Some essential requirements include installing the right software programs and sufficient processing power and memory to handle the system's demands.

4.2.1. Hardware Requirements:

The proposed emotion recognition classification system employs a personal computer hp with parameters such as Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz (4 CPUs), ~2.7GHz for CPU, RAM 8GB,Windows 10, and 64-bit Operating System.

4.2.2. Software Requirements

The suggested system was developed using IDLE and Python version 3.9. The system uses open-source libraries (TensorFlow, Keras, open-cv, numpy, and pandas).

4.3.Description of CREMA-D dataset

The CREMA-D dataset [95], short for "Crowd-Sourced Emotional Multimodal Actors Dataset," is a widely used database of facial expressions, vocalizations, and emotional information. They created it to provide a comprehensive collection of multimodal data for research in affective computing, emotion recognition, and related fields.

The participants of the study were instructed to verbalize a set of 12 predetermined phrases while exhibiting one of six distinct emotional states, namely Disgust, Fear, Sadness, Anger, Neutrality, and Happiness as shown in Figure (4.1). Additionally, the actors were required to deliver the phrases in three varying intonations. (Low, Medium, High). The auditory, visual, and audiovisual recordings were evaluated by a group of 2,443 raters through crowdsourcing, with the assessment being based on the intensity of the emotion and sensation.

Figure 4.1 : Original Dataset.

The dataset contains videos and audio recordings of actors performing various emotional expressions. It features 91 actors (48 male and 43 female) from various ethnic backgrounds, providing a diverse representation of emotions across different

genders and cultures. The database contains videos of various sizes, where the smallest video size is (90 KB) with a frame count of (17 frames), and the largest video size in the database is (674 KB) and the total number of frames within the video is (169 frames). Table (4.1) provides an overview of the dataset utilized and can consult for further information.

Table 4.1: Brief description of the CREMA-D dataset

Characteristic	Values
Number of videos	7442
Number of actors	91
Age for actors	From 20 to 74
Number of sentences	12
Number of classes	6
Sentence levels	Three levels (low, middle, high)

The CREMA-D dataset provides a rich set of annotations for each video and audio sample, including categorical labels for the expressed emotion, dimensional ratings (e.g., valence and arousal), and transcriptions of the spoken sentences. Figure (4-1) shows the number of videos for each type of class and the number of classes in the database.

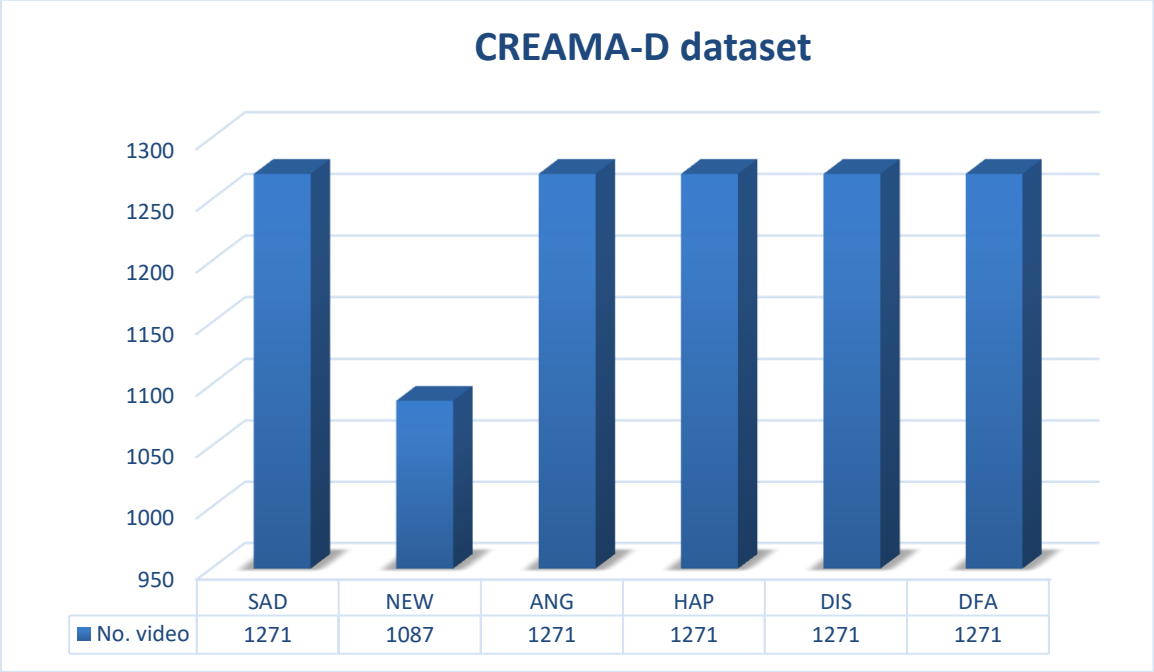


Figure 4.2: Number of videos for each class.

4.4. The Proposed System

This section of the thesis discusses an emotion recognition system based on images of human faces. Each stage of the system, including the preprocessing step, data split into (70% train and 30% test), face detection from the image, feature extraction, and classification using a conventional deep neural network, will be described in detail, along with a set of procedures that will be explained and put into practice. A sample of the first frame that was taken from random videos to illustrate the preprocessing before entering it into the proposed system is shown in Figure (4.3).

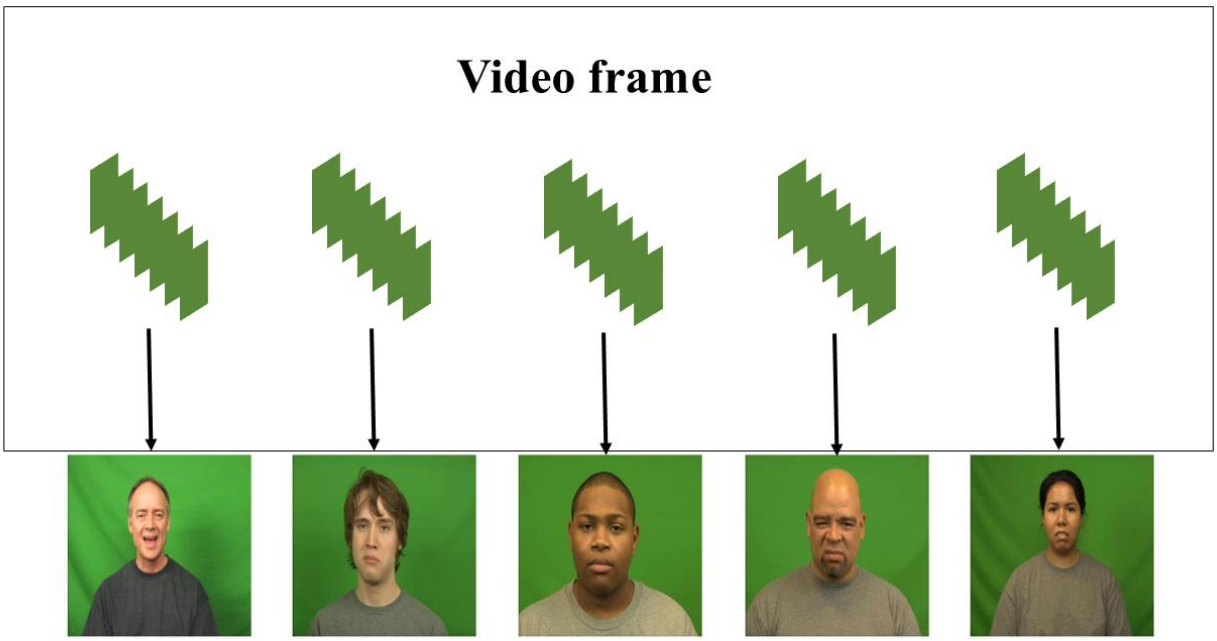


Figure 4.3: First frame from random video**.**

4.4.1. Data Preprocessing

The preprocessing data schedule for the two models included the following steps: (Extract frames, Frames to gray, Histogram Equalization,), Face detection based on (haar cascade), and resize), shown in Figure (4.4).

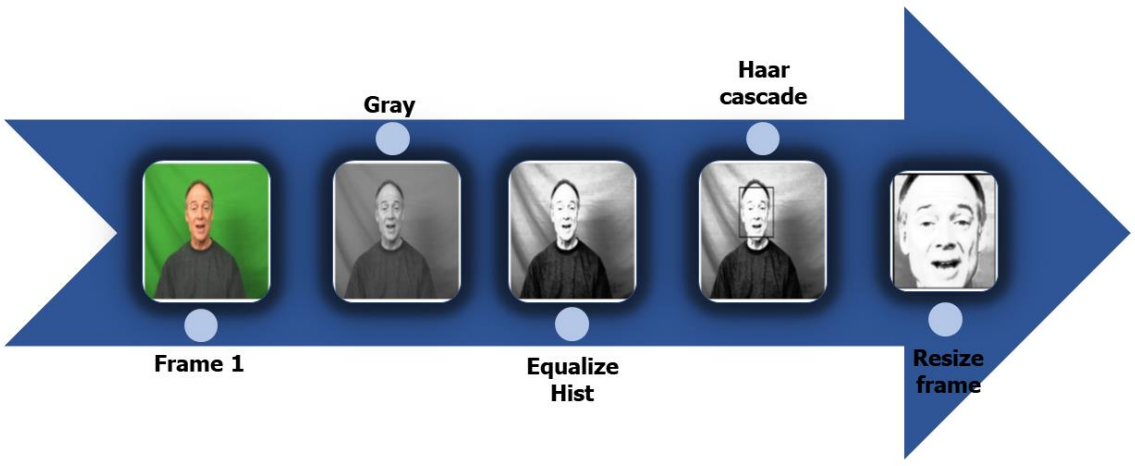


Figure 4.4: The preprocessing data scheduled**.**

The proposed recognition method begins with the first step, which involves preprocessing video frames. Preprocessing is crucial, and frames will go through numerous phases, each producing a different frame and having a different influence on the frame. Each of these steps will also have a different effect on the frame. The step of preprocessing, which consists of a set of steps to provide us with correct and ideal facial images to work within the stage of categorization, is broken down into the following four steps:

4.4.1.1. Extract Frames

In the preprocessing stage of video analysis or editing, extracting frames refers to capturing individual frames from a video file. As shown in Figure (4-2), Each frame represents a single image in the video sequence and can be treated as a standalone image for further processing or analysis. In order to access and process a video file by using the OpenCV library. Subsequently, the frames can be sequentially accessed. The present methodology accepts Frames as input and produces a boolean value that signifies whether the frame was successfully read or not, along with an accompanying visual representation. The individual frames of the movies may be readily extracted and manipulated according to your preferences, allowing for flexible usage.

Once extracted, a frame can perform various preprocessing operations on it if necessary. These operations might include resizing, filtering, or any other image processing technique requirements. Save the extracted frames to store them in memory for further analysis.

4.4.1.2. Convert Frames to Gray Level.

The image is initially changed from a color image to a grayscale image in the first stage of the preprocessing procedure. The process of transformation will result in the image requiring less space for storage and fewer channels in its representation. This is an essential aspect of the classification procedure because it allows the

classifier to complete recognition tasks with the proposed emotion recognition system more quickly and accurately while requiring less time overall. Samples of face photographs that have been transformed to the grayscale color space are displayed in Figure (4.5).

Figure 4.5: (a) represents RGB frames, and (b) represents the converted frame to a grayscale level**.**

In Figure (4.5), represent RGB frames by the OpenCV library in Python. It converts an image from the BGR color space to grayscale. In the RGB color space, an image is represented by three color channels: Blue (B), Green (G), and Red (R). Each channel contains intensity values ranging from 0 to 255, indicating that the colors contribute to the image's overall appearance. Grayscale images, on the other hand, have only one channel representing the intensity or brightness of each pixel. The intensity values typically range from 0 to 255, with 0 being black and 255 being white.

4.4.1.3. Apply Histogram Equalization

This step converts the face photographs to grayscale before applying the histogram equalization technique. This helps increase the contrast of the photos,

making the image details more transparent, which is helpful during the detection phase.

Histogram equalization is a method that redistributes the pixel intensities of an image to make full use of the available intensity range. It aims to improve the overall contrast and enhance the details in the image by stretching the intensity values across the entire range. Following histogram equalization, an example of the resulting face photographs is presented in Figure (4.6).

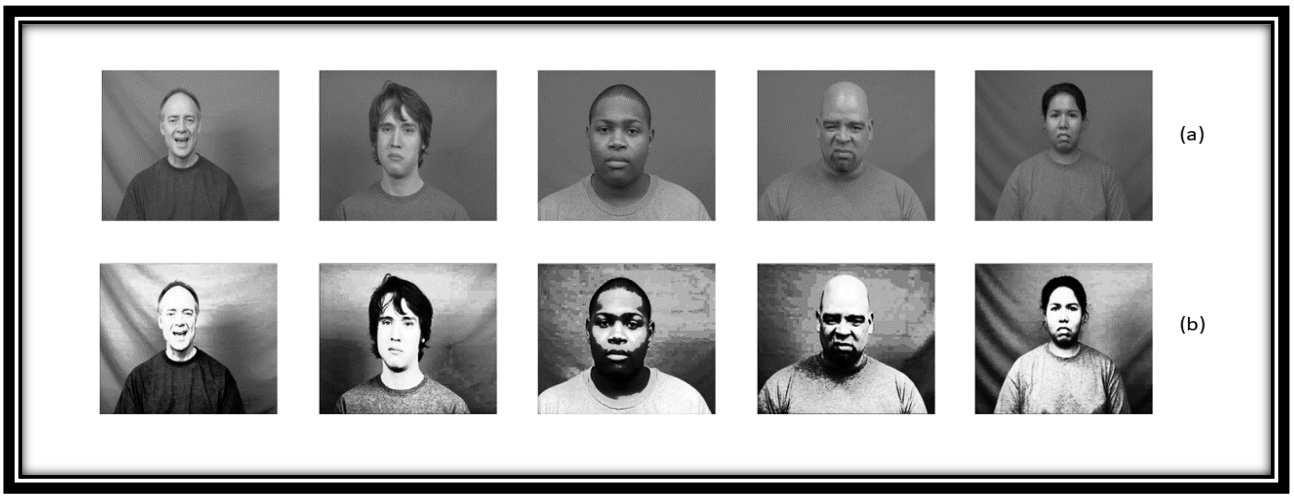


Figure 4.6: (a) Before applying Histogram Equalization (b) After applying Histogram Equalization.

Figure (4.6) shows that to enhance the contrast of a grayscale image using histogram equalization. The equalizeHist function takes a grayscale image as its input and applies histogram equalization to enhance the contrast. It automatically calculates the image's histogram and adjusts the pixel intensities accordingly. The result is an image with improved contrast and enhanced details. Histogram equalization is particularly useful when dealing with images with low contrast or limited dynamic range.

4.4.1.4. Face detection based on haar cascade (viola&jones)

After utilizing Adaboost machine learning to identify the relevant details in the face photos and remove all undesirable details from the photographs, this stage's goal

is to achieve the viola&jones detection technique. This is accomplished by removing all unnecessary details from the images. The examples of the results of the detection technique are shown in Figure (4.7).

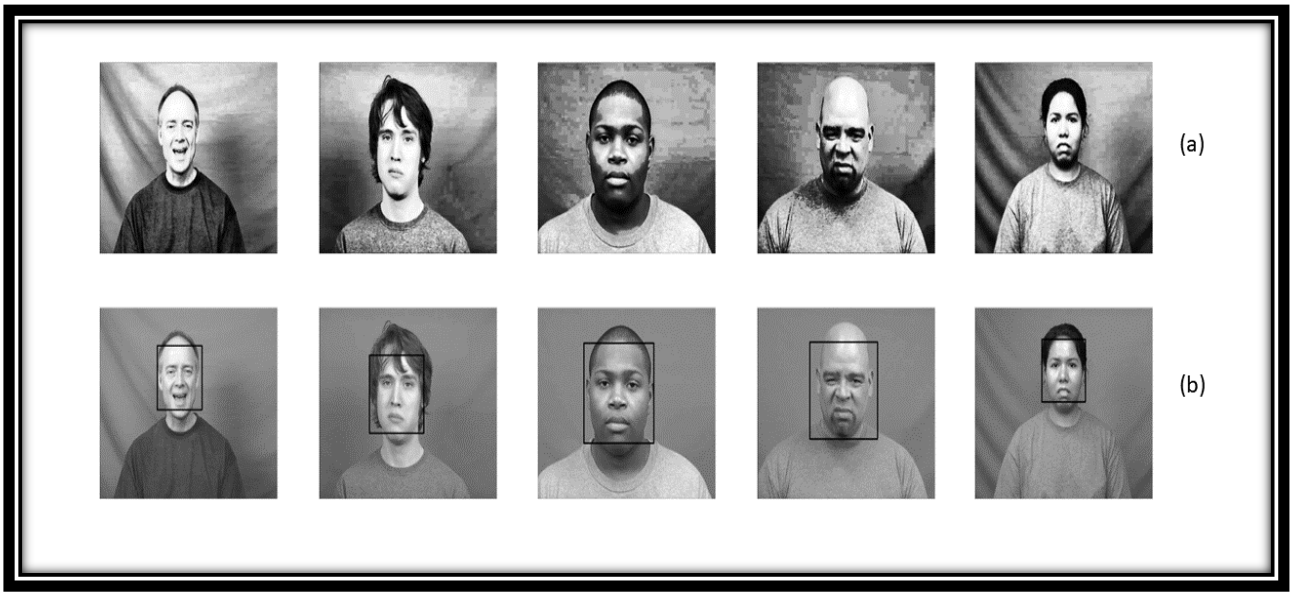


Figure 4.7: (a) Faces before applying (b) Face detection based on haar cascade(Viola Jones)

After defining the face, we cut the face area only by calculating the values (x, y, w, h). Figure (4.8) shows the stage of cutting the face only because it is a crucial part of classifying human feelings.

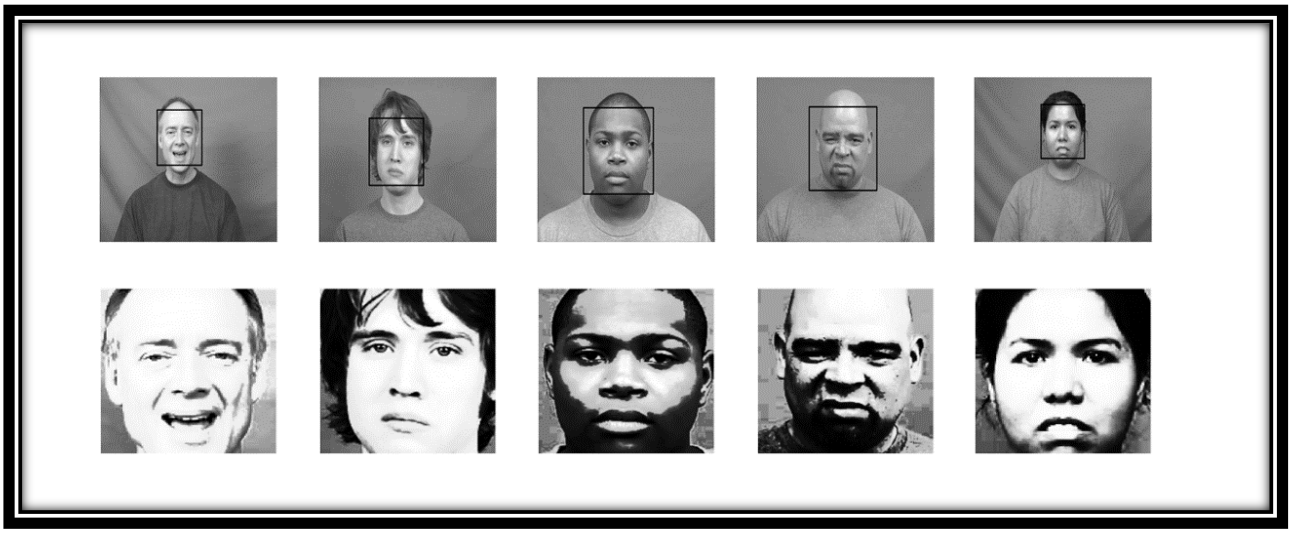


Figure 4.8: faces cropping after using haar cascade(Viola Jones).

4.4.1.5. Resize Images

After conducting face detection, the following stage is to accomplish the resizing process, during which the digital facial grey picture will attain a new dimensional value, during which it will become smaller. This will take place after the previous phase has been completed. All photographs were reshaped using a ratio of (50 * 50) for the distances. After the scaling procedure, the facial picture is shown in Figure (4.9).

Figure 4.9 : Applying resizes on frames.

4.4.2. Feature extraction

After processing, the work is divided into two parts, considered unified for both proposed models. The frames are entered directly into the CNN algorithm in the first part. As for the part in which one-dimensional deep learning is used, use another processing method to extract the characteristics before entering them into the model.

The HOG algorithm was applied to the resized frame (20 * 20) to extract the features for subsequent analysis and modeling. The following Figure (4.10) shows the characteristics extracted from the images after processing.

hog1	hog2	hog3	hog4	hog5	hog6	hog7	hog8	hog9	hog10	hog11	h
0.257544	0.28815	0.28815	0.25889	0.071459	0.030921	0.085091	0.047339	0.121069	0.182365	0.154349	
0.255102	0.259689	0.295002	0.288596	0.059021	0.030058	0.048188	0.063927	0.150431	0.191979	0.133228	
0.245567	0.256429	0.290775	0.282093	0.065146	0.04194	0.050489	0.061501	0.165844	0.197064	0.131763	
0.276328	0.290997	0.211092	0.290997	0.095592	0.032225	0.060239	0.050659	0.145707	0.242999	0.138372	
0.275301	0.282987	0.282987	0.282987	0.057873	0.027218	0.053276	0.047779	0.122155	0.195762	0.148147	
0.233937	0.286861	0.286861	0.135273	0.229142	0.049595	0.049114	0.029635	0.137948	0.161218	0.090291	
0.236006	0.283628	0.283628	0.283628	0.092719	0.034993	0.065929	0.013153	0.182647	0.171761	0.138391	
0.281298	0.281298	0.281298	0.281298	0.064239	0.01437	0.040051	0.057935	0.181054	0.181324	0.064524	
0.177352	0.2851	0.2851	0.137662	0.218295	0.029718	0.085038	0.033326	0.157844	0.166515	0.10818	
0.224224	0.281525	0.254017	0.25075	0.130609	0.063456	0.056422	0.057312	0.148413	0.22478	0.191229	
0.271015	0.282917	0.282917	0.104703	0.203952	0.043918	0.080855	0.03829	0.138468	0.282917	0.089936	
0.234735	0.280863	0.280863	0.045647	0.199811	0.059925	0.022557	0.100333	0.131979	0.221358	0.104851	
0.257507	0.257507	0.257507	0.257507	0.066334	0.02287	0.066727	0.083771	0.145391	0.257507	0.051563	
0.21486	0.284051	0.284051	0.032766	0.205691	0.050393	0.021613	0.122615	0.096728	0.246108	0.124042	
0.232028	0.268073	0.268073	0.047697	0.204892	0.045141	0.075616	0.068438	0.133961	0.268073	0.089756	
0.273845	0.275075	0.275075	0.181237	0.120377	0.004925	0.036897	0.012154	0.147945	0.220558	0.154049	
0.165282	0.290625	0.290625	0.117902	0.126958	0.010602	0.044149	0.098113	0.130524	0.13998	0.233729	
0.23375	0.28616	0.28616	0.043982	0.280173	0.014156	0.068618	0.033283	0.110426	0.179262	0.18176	

Figure 4.10: HOG feature extraction

After extracting the 576 characteristics utilizing HOG and storing them in a CSV file, the next step was to input them into the machine and Deep learning reduction (CNN 1D).

4.4.3. CNN 1D

This thesis applies deep learning, often known as CNN, which is one of the most popular and widely used deep learning classification techniques. Deep learning algorithms are believed to be flawless in classification, producing highly accurate results. This is especially true when working with large data sets; each workbook created using deep learning techniques consists of several layers.

The CNN 1D layers used for facial emotion detection proposed in this thesis can be found in Table (4.2), which provides comprehensive information on each layer that constitutes the CNN classification algorithm.

Table 4.2: CNN Layers Specific Details

Type	Filter	Parameter
Convolution 1D	16	64
Max pooling 1D	16	0
Convolution 1D	32	1568

Max pooling 1D	32	0
Convolution 1D	64	6208
Max pooling 1D	64	0
Convolution 1D	32	6176
Max pooling 1D	32	0
Convolution 1D	64	6208
Max pooling 1D	64	0
Convolution 1D	64	12352
Max pooling 1D	64	0
Convolution 1D	32	6176
Convolution 1D	32	3104
Max pooling 1D	32	0
Dense	128	4224
Dense	64	8256
Dense	64	4160
Dense	16	1040
Convolution 1D	55	2659
Flatten 1D	30800	0
Dense	6	184806
Total parameter		247,037

Table (4.2) describes information about each layer of the CNN1D that was implemented and shows the total number of transactions, which amounted to 248,687. In addition, the results show that the use of CNN1D with the HOG algorithm improved accuracy over what was extracted in the first model; The assessment and deployment results for the second model, CNN1D-HOG, for facial expression recognition are displayed in Tables 4.3 and 4.4. The performance of the model is

measured through assessment measures including precision, recall, f1-score, and accuracy.

The assessment scores for each group are shown in Tables 4.3. Precision, recall, and f1-score values range from 0.99 to 1.00 across the board for each emotion category, including ANG (Anger), DIS (Disgust), FEA (Fear), HAP (Happiness), SAD (Sadness), and NEU (Neutral). These results show that the CNN1D-HOG model performs very well in classifying facial expressions into their respective emotion categories.

Table 4.3: Deep CNN1D-HOG evaluation in every class

Classes	Pression	Recall	f1-score
ANG	0.99	0.99	0.99
DIS	0.99	0.99	0.99
FEA	0.99	0.99	0.99
HAP	0.99	0.99	0.99
SAD	1.00	0.99	0.99
NEU	0.99	0.99	0.99

Table 4.4: CNN-HOG implementation**.**

Accuracy	Pression	Recall	f1-score
0.99	0.99	0.99	0.99

The implementation results can now be analyzed.in Figure 4.11. All metrics, including accuracy, precision, recall, and f1, are at a perfect 0.99. That the CNN1D-HOG model is so successful across the board in terms of assessment criteria is clear evidence of its high degree of overall accuracy. With an accuracy of 0.99, the model has a high degree of confidence in its ability to properly categorize facial emotions.

Tables (4.3) and (4.4) show that the suggested CNN1D-HOG model for facial expression recognition is effective and reliable. High precision, recall, f1-score, and accuracy scores demonstrate the model's outstanding ability in correctly categorizing emotional states. These findings demonstrate that the CNN1D-HOG model can accurately capture and categorize the nuances between facial expressions that convey various emotions. The enhanced capacity to extract important features and predict outcomes more accurately than the prior model shows that HOG features added to the CNN1D architecture are responsible for the improvement.

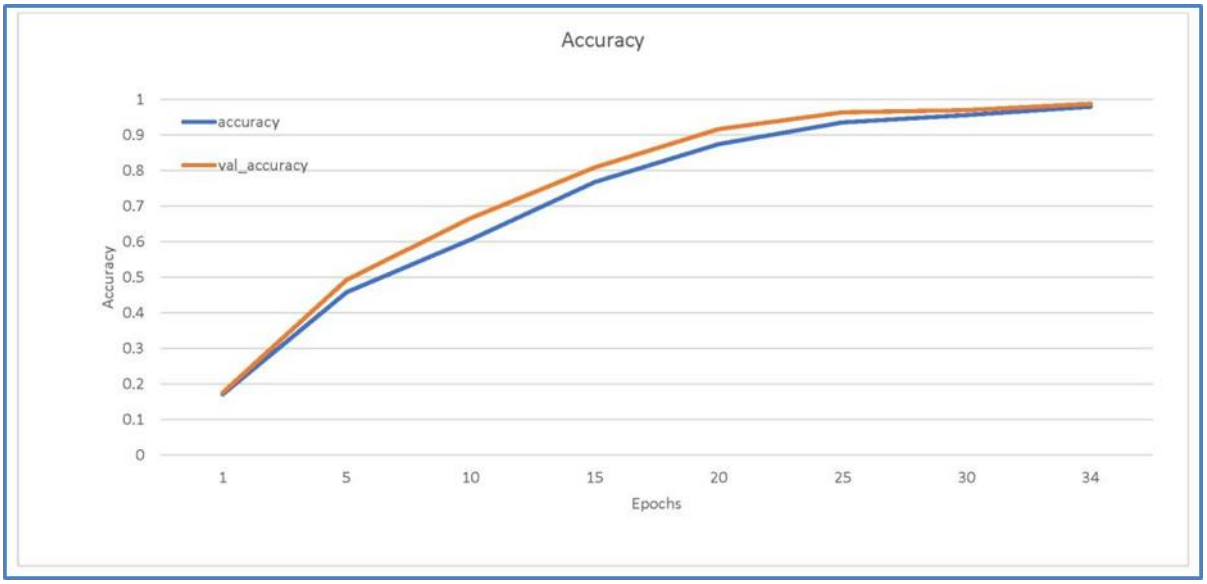


Figure 4.11: Accuracy and val- accuracy for CNN1D Classification.

Figure (4.11) shows the accuracy and validation accuracy (val-accuracy) trends for 1D CNN1D classification. The model's accuracy in categorizing emotions on the training dataset is called accuracy, while its validation dataset accuracy is called val-accuracy. The graph shows how model accuracy changes after training. Initial accuracy and val-accuracy values rise, indicating that the model is learning from the training data. As training progresses, accuracy stabilizes, suggesting that the model makes reliable predictions on the training dataset. Val-accuracy follows a similar trend, but can vary significantly owing to validation data.

It's important that accuracy and val-accuracy converge and stay nearby. A large gap between these two curves may suggest overfitting, where the model performs well on training data but struggles to generalize to new data. Figure (4.10) shows how the CNN1D model's accuracy develops throughout training, which is crucial to understanding its training dynamics and performance.

4.4.4. CNN 2D

To classify the video based on counting the frames within the video and extract the most accurate characteristics, one of the complex challenges we face is inserting a series of video frames into the CNN network. This is because we rely on the time sequence of frames to provide clear landmarks and impressions for each video.

In this case, a time-distributed layer that allows the CNN model was used to process a series of images, share weights over time steps, reduce parameters, and improve computational efficiency, allowing the model to learn and capture temporal patterns in the input sequence.

Table (4.5), which includes thorough information on each layer that creates the CNN classification algorithm, contains the CNN 2D layers that were used for facial emotion recognition in this thesis.

Table 4.5: CNN 2D Layers Specific Details

Type	Filter	Parameter
TimeDistributed(Conv2D)	64	640
TimeDistributed(Conv2D)	64	36928
MaxPooling2D	64	0
TimeDistributed(Conv2D)	128	73856
TimeDistributed(Conv2D)	128	147584
MaxPooling2D	128	0
Flatten 2D	768	0

Dense	1024	787456
Dense	512	524800
Dropout	512	0
Dense	6	3078
Total parameters		1,574,342
Trainable parameters		1,574,342
Non-trainable		0

The TimeDistributed layer in the given Table (4.5) is used to apply the same layer to each time step independently.

In the context of the Table above provided, the input shape of the model is (6, 50, 50, 1), which means we have a sequence of 6 grayscale images, each with dimensions 50x50. The TimeDistributed layer allows the processing of each image in the sequence individually while sharing the same weights and biases across all time steps.

Table (4.6) and Figure 4-11 show the outcomes of testing the proposed CNN 2D system on different types of emotions. Each class's performance in correctly identifying instances of that feeling is displayed in the table with metrics like Precision, Recall, and F1-score. Upon closer inspection, we find that results vary significantly between categories. For example, for classes like ANG, DIS, and HAP, the system demonstrates high Precision, Recall, and F1-score, indicating strong accuracy in identifying instances of these emotions. Comparatively lower Precision, Recall, and F1-score values are displayed by classes like FEA and SAD. The NEU group is in the middle, with acceptable efficiency indicators.

Table 4.6: CNN 2D evaluation for every class**.**

Classes	Pression	Recall	f1-score
ANG	0.87	0.74	0.80
DIS	0.91	0.82	0.86
FEA	0.76	0.82	0.79
HAP	0.91	0.92	0.92
SAD	0.68	0.81	0.74
NEU	0.85	0.79	0.82

Next, the overall accuracy of the proposed method in recognizing all six emotions is demonstrated in Table (4.7) and Figure (4.12). Metrics like Accuracy, Precision, Recall, and F1-score are included in the table to provide a holistic assessment of the system's efficacy.

A reasonable degree of classification accuracy was attained by the proposed method across all classes, as indicated by the accuracy score of 0.82. This indicates that the algorithm did a decent job of recognizing and differentiating between the six emotions.

Table 4.7: CNN 2D implementation.

Accuracy	Pression	Recall	f1-score
0.82	0.83	0.82	0.82

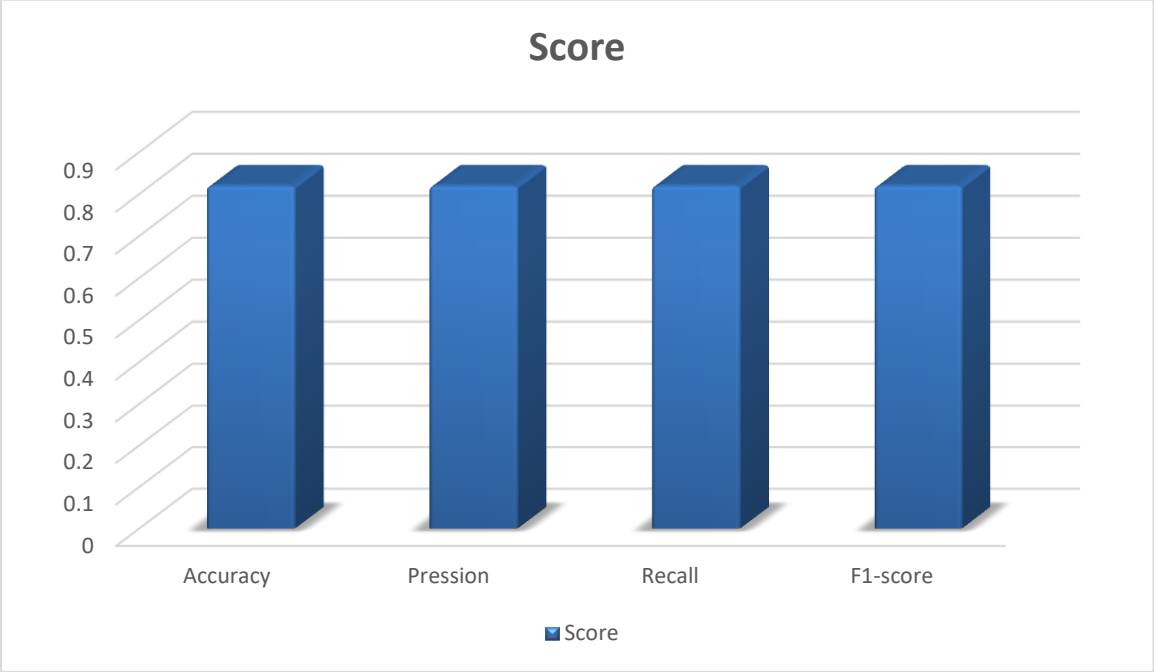


Figure 4.12: Proposed model.

The findings show that the suggested CNN 2D system is effective at recognizing facial expressions. In addition to adequate accuracy across all classes, the system demonstrates high performance for particular emotional states. The system's effectiveness in categorizing emotions with lower accuracy and recall values should be improved with more research and assessment to determine what variables contribute to the variances in performance among emotions.

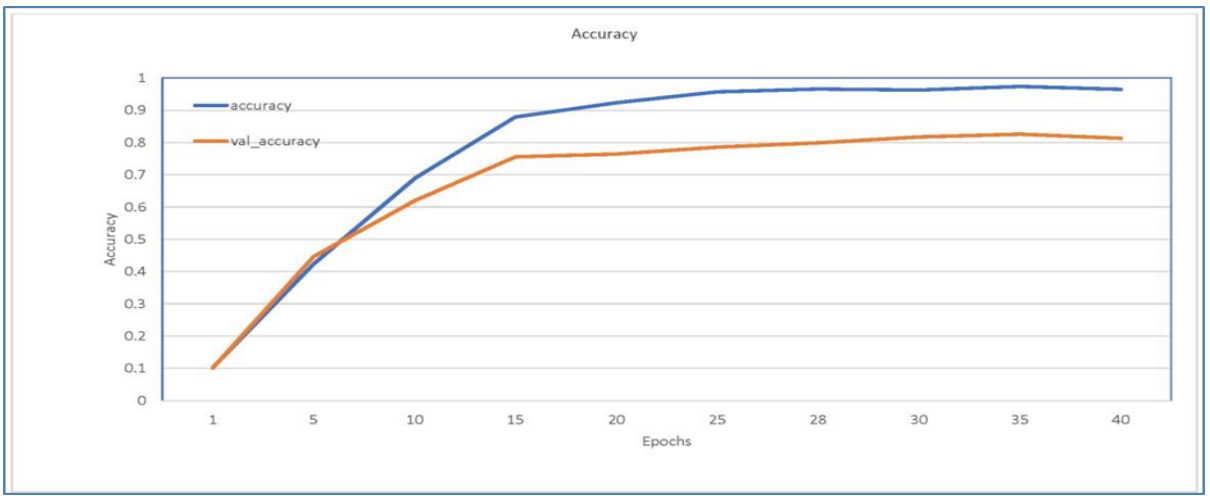


Figure 4.13: Accuracy and val- accuracy for CNN2D Classification.

Figure (4.13) shows 2D Convolutional Neural Network classification accuracy and validation accuracy (val-accuracy) trends. This graph shows model accuracy during training and validation. The model classifies emotions across the training dataset as shown by the accuracy curve. In contrast, the val-accuracy curve shows the model's validation dataset accuracy. This curve shows the model's learning progress and capacity to generalize to new data.

When training begins, the accuracy and val-accuracy curves climb, suggesting that the model is learning patterns and features from the training data. The model's consistent performance on the training dataset stabilizes the accuracy curve as training proceeds. Although validation data may cause modest oscillations, the val-accuracy curve follows a similar trend. Accuracy and val-accuracy curves should converge and stay close, indicating that the model is not overfitting and can generalize to new data. For CNN2D model learning dynamics, Figure (4.13) is useful. It helps academics and practitioners evaluate the model's training, data pattern capture, and performance.

4.5. Quranic verses based on Face Recognition

A recommendation algorithm with Quranic principles is easily integrated into the system so that it may deliver individualized content suggestions to the user. This algorithm also takes into consideration the user's emotional condition. The user is greeted by a clear interface upon initialization of the system, where they are given the opportunity to submit an image including a facial representation for the purpose of emotion categorization. Following the recognition and localization of faces, the software makes use of a deep learning model to assign each facial expression to one of the following six primary states of mind: anger, disgust, fear, happiness, neutrality, and sadness. After that, it displays the emotion label that was predicted in addition to a confidence score that indicates how certain the model is.

The suggestion system conducts an analysis of the identified emotions and links them with appropriate verses from the Quran. This process is informed by the

principles outlined in the Quran. These recommended verses provide users direction and a connection to their emotions depending on the feelings they have discovered for themselves. The user may easily switch between input alternatives, such as utilizing a camera or picking photos from a library, thanks to the interface's supplemental features, which include webcam support and the ability to select images from a library. Users also have the option to store the findings they acquire for future reference or for sharing with others.

4.6. Result Comparison with Other Studies

The tables below compare with other works using the evaluation methods mentioned above. In addition, it is comparable to several papers that used CREMA-D datasets. The findings depicted in Table (4.8) exhibit the enhanced efficacy of the suggested model in contrast to the CNN 1D model proposed by Srinivas and Mishra [19]. The model under consideration demonstrated a superior accuracy of 0.99, surpassing the accuracy of the reference model, which stood at 0.9788. This observation demonstrates the efficacy of the proposed model in accurately forecasting facial expressions, as indicated by its superior precision, recall, and F1-score metrics compared to the performance of the reference model.

Table 4.8: Comparison with CNN 1D model**.**

Reference	**Accuracy**	**Pression**	**Recall**	**F1-score**
Srinivas and Mishra [19]	0.9788	0.716	0.7	0.688
The proposed	0.99	0.99	0.99	0.99

Table (4.9) presents a comparative analysis of the proposed model with several 2D convolutional neural network (CNN) models sourced from different references. The model under consideration demonstrated a level of accuracy of 0.82, surpassing the accuracy values documented in the referenced studies. The accuracy values reported by Vijay and Yasutomo [18], Hans and Rao [15], Ryumina and Karpov [11], Ghaleb et al. [13], and Ristea et al. [10] were 0.7245, 0.7852, 0.791, 0.665, and 0.6248, respectively. The findings of this study indicate that the proposed model demonstrates superior accuracy in the recognition of facial expressions when compared to the reference models.

Table 4.9: Comparison with CNN 2D model

Reference	**Accuracy**
Vijay and Yasutomo [18]	0.7245
Hans and Rao [15]	0.7852
Ryumina and Karpov[11]	0.791
Ghaleb et al [13]	0.665
Ristea et al. [10]	0.6248
The proposed	0.82

Figure (4.14) presents a graphical depiction of the accuracy performance observed in different studies. The accuracy of the proposed model is illustrated, demonstrating its performance in comparison to the studies referenced. The graph presented in this study provides a comprehensive visual representation of the comparative accuracy results, shedding light on the efficacy of the proposed model in accurately discerning facial expressions.

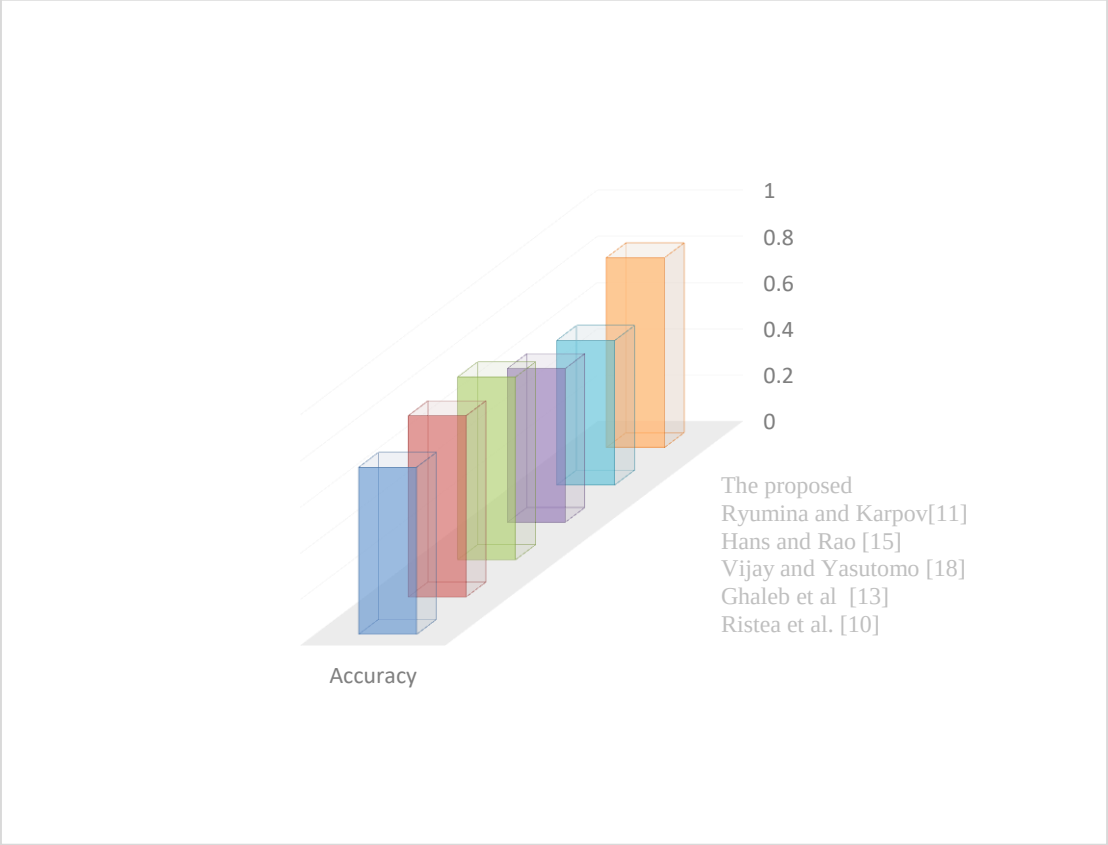


Figure 4.14: Accuracy Performance for Various Studies.

Figure 4.14 presents a comparative examination of accuracy performance across several investigations, encompassing the proposed models as well. The objective of these research was to investigate face expression recognition through the use of various approaches and procedures. Significantly, the proposed model demonstrated a noteworthy accuracy of 0.82, surpassing the performance of several prior investigations.

Upon examining the comparison between the two models, it becomes evident that the 1D CNN model exhibits exceptional accuracy, reaching an impressive rate of 99%. In contrast, the 2D CNN models demonstrate different degrees of achievement, all about at 82%. The enhanced efficacy of the one-dimensional convolutional neural network (1D CNN) model may be ascribed to a multitude of pivotal elements.

To begin with, it is worth noting that one-dimensional convolutional neural network (1D CNN) models exhibit remarkable proficiency in processing sequential input, rendering them very suitable for jobs that include time series or sensor/measurement data. The ability of the model to identify complex patterns within the dataset enables it to achieve a high level of precision in classification. Furthermore, the particular configuration of the one-dimensional convolutional neural network (1D CNN) architecture demonstrated greater compatibility with the dataset under consideration. Two-dimensional convolutional neural networks (CNNs) demonstrate superior performance in handling data organized in a grid-like structure. Conversely, one-dimensional CNNs exhibit exceptional capabilities in processing sequential data. The aforementioned versatility rendered the use of the 1D CNN a more pragmatic selection for this specific undertaking.

Nevertheless, it is important to acknowledge that attaining such impressive precision necessitated substantial hyperparameter adjustment and refining, a practice that is prevalent in contemporary scholarly investigations. The performance of a deep learning model is heavily influenced by the careful selection of hyperparameters such as kernel size, stride, and filter types. In addition, the significance of both the quality and amount of data cannot be overstated in the attainment of elevated levels of accuracy.

CHAPTER FIVE

CONCLUSIONS AND FUTUREWORKS

Chapter Five

Conclusions and Future Works

5.1 Conclusions

The findings of this thesis provide convincing evidence that important contributions were made toward answering the study's research questions and accomplishing its stated goals.

1. When entering frames into deep learning models, the time-distributed layer is used. The time-distributed layer aims to apply different layers or processes to each input sequence time step. This component helps the model efficiently learn sequence temporal patterns and dependencies.

2. The utilization of the Histogram of Oriented Gradients (HOG) approach is of utmost importance in the feature extraction process from video frames inside the suggested system. The Histogram of Oriented Gradients (HOG) approach is proficient at capturing substantial texture and shape information, hence enhancing the system's capability to properly interpret face emotions. The utilization of HOG-extracted features enhances the system's efficacy in obtaining pertinent data for later phases of emotion categorization.

3. The effectiveness of deep learning algorithms in accurately recognizing and classifying facial expressions has been extensively researched. Developing the layered structure in deep learning led to improvements in the model’s performance and predictive ability to detect emotions.

4. The purpose of histogram equalization is to make the most of the range of available intensities by redistributing the pixel intensities within an image. It aims to improve the overall contrast of the image and draw attention to the smaller features that help to more accurately identify faces and improve the effectiveness of the model.

5.2 Future Works:

Despite the impressive progress made in face expression detection and its incorporation with Quranic texts, there are still many questions to be answered and areas to be improved upon.

1. Using supplementary data sources, the accuracy of emotion identification might be improved by including additional modalities such as audio, physiological signs, or textual information.

2. Increasing the amount and variety of the training dataset through the use of cutting-edge data augmentation techniques can assist enhance the model's generalization and resilience.

3. Further studies may focus on the Using fuzzy to choose Quranic verses based on feelings by choosing the surah that is closest to expectation using the fuzzy membership function.

4. Using the optimization approach known as the Adaptive Gradient approach in deep learning to train models. During training, it is intended to adjust the ratesb at which various model parameters learn.

References

References

[1] D. Li, Z. Wang, Q. Gao, Y. Song, X. Yu, and C. Wang, “Facial expression recognition based on Electroencephalogram and facial landmark localization,” Technology and Health Care, vol. 27, no. 4, 2019, doi: 10.3233/THC-181538.

[2] S. Li and W. Deng, “Deep Facial Expression Recognition: A Survey,” IEEE Trans Affect Comput, vol. 13, no. 3, 2022, doi: 10.1109/TAFFC.2020.2981446.

[3] B. Li and D. Lima, “Facial expression recognition via ResNet-50,” International Journal of Cognitive Computing in Engineering, vol. 2, 2021, doi: 10.1016/j.ijcce.2021.02.002.

[4] A. Elmahmudi and H. Ugail, “Deep face recognition using imperfect facial data,” Future Generation Computer Systems, vol. 99, 2019, doi: 10.1016/j.future.2019.04.025.

[5] E. Goceri, “Deep learning based classification of facial dermatological disorders,” Comput Biol Med, vol. 128, 2021, doi: 10.1016/j.compbiomed.2020.104118.

[6] K. Chlasta, K. Wołk, and I. Krejtz, “Automated speech-based screening of depression using deep convolutional neural networks,” Procedia Computer Science, vol. 164, pp. 618–628, 2019, doi: 10.1016/j.procs.2019.12.228.

[7] W. H. Abdulsalam, R. S. Alhamdani, and M. N. Abdullah, “Facial emotion recognition from videos using deep convolutional neural networks,” Int J Mach Learn Comput, vol. 9, no. 1, pp. 14–19, 2019, doi: 10.18178/ijmlc.2019.9.1.759.

[8] D. A. S. Devi, C. Satyanarayana, and D. S. Rekha, “Facial Emotion Recognition Using Hybrid Approach for DCT and DBACNN,” in Lecture Notes in Electrical Engineering, Springer Science and Business Media Deutschland GmbH, 2022, pp. 411–423. doi: 10.1007/978-981-16-9885-

9_34.

[9] H. M. Shahzad, S. M. Bhatti, A. Jaffar, and M. Rashid, “A Multi-Modal Deep Learning Approach for Emotion Recognition,” Intelligent Automation and Soft Computing, vol. 36, no. 2, pp. 1561–1570, 2023, doi: 10.32604/iasc.2023.032525.

[10] N. C. Ristea, L. C. Dutu, and A. Radoi, “Emotion recognition system from speech and visual information based on convolutional neural networks,” in 2019 10th International Conference on Speech Technology and Human-Computer Dialogue, SpeD 2019, Institute of Electrical and Electronics Engineers Inc., Oct. 2019. doi: 10.1109/SPED.2019.8906538.

[11] E. Ryumina and A. Karpov, “Facial expression recognition using distance importance scores between facial landmarks,” in CEUR Workshop Proceedings, CEUR-WS, 2020. doi: 10.51130/graphicon-2020-2-3-32.

[12] A. Birhala, C. N. Ristea, A. Radoi, and L. C. Dutu, “Temporal aggregation of audio-visual modalities for emotion recognition,” in 2020 43rd International Conference on Telecommunications and Signal Processing, TSP 2020, Institute of Electrical and Electronics Engineers Inc., Jul. 2020, pp. 305–308. doi: 10.1109/TSP49548.2020.9163474.

[13] E. Ghaleb, M. Popa, and S. Asteriadis, “Metric Learning-Based Multimodal Audio-Visual Emotion Recognition,” IEEE Multimedia, vol. 27, no. 1, pp. 37–48, Jan. 2020, doi: 10.1109/MMUL.2019.2960219.

[14] J. Sujanaa and S. Palanivel, “Hog-Based Emotion Recognition Using One-Dimensional Convolutional Neural Network,” ICTACT J Image Video Process, vol. 11, no. 02. ictactjournals.in, pp. 2310–2315, 2020. [Online]. Available: http://ictactjournals.in/paper/IJIVP_Vol_11_Iss_2_Paper_5_2310_2315.pdf

[15] A. S. A. Hans and S. Rao, “A CNN-LSTM based deep neural networks for facial emotion detection in videos,” International Journal of Advances In Signal And Image Sciences, vol. 7, no. 1, pp. 11–20, Mar. 2021, doi:

10.29284/ijasis.7.1.2021.11-20.

[16] J. Sujanaa, S. Palanivel, and M. Balasubramanian, “Emotion recognition using support vector machine and one-dimensional convolutional neural network,” Multimed Tools Appl, vol. 80, no. 18, pp. 27171–27185, Jul. 2021, doi: 10.1007/s11042-021-11041-5.

[17] F. Zamani and R. Wulansari, “Emotion Classification using 1D-CNN and RNN based On DEAP Dataset,” Academy and Industry Research Collaboration Center (AIRCC), Dec. 2021, pp. 363–378. doi: 10.5121/csit.2021.112328.

[18] V. John and Y. Kawanishi, “Audio and Video-based Emotion Recognition using Multimodal Transformers,” in Proceedings - International Conference on Pattern Recognition, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 2582–2588. doi: 10.1109/ICPR56361.2022.9956730.

[19] P. V. V. S. Srinivas and P. Mishra, “Human Emotion Recognition by Integrating Facial and Speech Features: An Implementation of Multimodal Framework using CNN,” International Journal of Advanced Computer Science and Applications, vol. 13, no. 1, 2022, doi: 10.14569/ijacsa.2022.0130172.

[20] J. Latif, C. Xiao, A. Imran, and S. Tu, “Medical imaging using machine learning and deep learning algorithms: A review,” in 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies, iCoMET 2019, 2019. doi: 10.1109/ICOMET.2019.8673502.

[21] Z. Wang et al., “Automated Rest EEG-Based Diagnosis of Depression and Schizophrenia Using a Deep Convolutional Neural Network,” IEEE Access, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3197645.

[22] B. Victor, K. Bowyer, and S. Sarkar, “An evaluation of face and ear biometrics,” Proceedings - International Conference on Pattern Recognition, vol. 16, no. 1, 2002, doi: 10.1109/icpr.2002.1044746.

[23] Y. Ma, Z. Huang, X. Wang, and K. Huang, “An Overview of Multimodal Biometrics Using the Face and Ear,” Mathematical Problems in Engineering, vol. 2020. 2020. doi: 10.1155/2020/6802905.

[24] Bouman, C. A. “Chapter 13: Markov Chains and Hidden Markov Models,” Foundations of Computational Imaging: A Model-Based Approach, pp. 203–221, Jan. 2022, doi: 10.1137/1.9781611977134.ch13.

[25] B. Verma and A. Choudhary, “Grassmann manifold based dynamic hand gesture recognition using depth data,” Multimedia Tools and Applications, vol. 79, no. 3–4, pp. 2213–2237, Nov. 2019, doi: 10.1007/s11042-019-08266-w.

[26] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, , “Bosphorus database for 3D face analysis,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008. doi: 10.1007/978-3-540-89991-4_6.

[27] R. Zhi, M. Liu, and D. Zhang, “A comprehensive survey on automatic facial action unit analysis,” Visual Computer, vol. 36, no. 5, 2020, doi: 10.1007/s00371-019-01707-5.

[28] V. Lyashenko, “General Ideology of Analysis Digital Medical Images in RGB Format,” International Journal of Emerging Trends in Engineering Research, vol. 8, no. 5, pp. 1647–1655, May 2020, doi: 10.30534/ijeter/2020/25852020.

[29] H. Mutahira, B. Ahmad, M. S. Muhammad, and D. R. Shin, “Focus Measurement in Color Space for Shape from Focus Systems,” IEEE Access, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3098753.

[30] K. Kumar, R. K. Mishra, and D. Nandan, “Efficient Hardware of RGB to Gray Conversion Realized on FPGA and ASIC,” in Procedia Computer Science, 2020. doi: 10.1016/j.procs.2020.04.215.

[31] A. Durdu, “A new data hiding method with high capacity, low distortion, and reversible loss that hides 24-bit color image into 24-bit color image (YKKG),” Pamukkale University Journal of Engineering Sciences, vol. 27, no. 2, pp. 96–113, 2021, doi: 10.5505/pajes.2020.50215.

[32] S. Sood, H. Singh, and M. Malarvel, “Image quality enhancement for Wheat rust diseased images using Histogram equalization technique,” 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Apr. 2021, doi: 10.1109/iccmc51019.2021.9418023.

[33] H. Kh. Omar and N. E. Tawfiq, “Face Recognition Based on Histogram Equalization and LBP Algorithm,” Academic Journal of Nawroz University, vol. 8, no. 3, p. 33, Aug. 2019, doi: 10.25007/ajnu.v8n3a394.

[34] S. H. Gangolli, A. Johnson Luke Fonseca, and R. Sonkusare, “Image Enhancement using Various Histogram Equalization Techniques,” 2019 Global Conference for Advancement in Technology (GCAT), Oct. 2019, doi: 10.1109/gcat47503.2019.8978413.

[35] D. D. Olatinwo, A. Abu-Mahfouz, G. Hancke, and H. Myburgh, “IoT-Enabled WBAN and Machine Learning for Speech Emotion Recognition in Patients,” Sensors, vol. 23, no. 6, p. 2948, Mar. 2023, doi: 10.3390/s23062948.

[36] A. Jadhav, S. Lone, S. Matey, T. Madamwar, and S. Jakhete, “Survey on Face Detection Algorithms,” Int J Innov Sci Res Technol, vol. 6, no. 2, 2021.

[37] Z. Dahirou, M. Zheng, and M. Yuxin, “Face Detection with Viola Jones Algorithm,” 2020 7th International Conference on Information Science and Control Engineering (ICISCE), Dec. 2020, doi: 10.1109/icisce50968.2020.00130.

[38] P. Xayachack and J. Zhang, “Robust Face detection based on Viola-jones Algorithms,” IOP Conference Series: Materials Science and Engineering, vol. 768, no. 6, p. 062079, Mar. 2020, doi: 10.1088/1757-

899x/768/6/062079.

[39] J. Huang, Y. Shang, and H. Chen, “Improved Viola-Jones face detection algorithm based on HoloLens,” EURASIP J Image Video Process, vol. 2019, no. 1, 2019, doi: 10.1186/s13640-019-0435-6.

[40] “Comparative of Viola-Jones and YOLO v3 for Face Detection in Real time,” Iraqi Journal of Computer, Communication, Control and System Engineering, 2022, doi: 10.33103/uot.ijccce.22.2.6.

[41] A. Garg, and A. Negi, “A Survey on Content Aware Image Resizing Methods,” KSII Transactions on Internet and Information Systems, vol. 14, no. 7, Jul. 2020, doi: 10.3837/tiis.2020.07.015.

[42] V. Upadhyay and D. Kotak, “A Review on Different Facial Feature Extraction Methods for Face Emotions Recognition System,” 2020 Fourth International Conference on Inventive Systems and Control (ICISC), Jan. 2020, doi: 10.1109/icisc47916.2020.9171172.

[43] M. Moe Htay, “Feature extraction and classification methods of facial expression: a survey,” Computer Science and Information Technologies, vol. 2, no. 1, pp. 26–32, Mar. 2021, doi: 10.11591/csit.v2i1.p26-32.

[44] H. Benradi, A. Chater, and A. Lasfar, “A hybrid approach for face recognition using a convolutional neural network combined with feature extraction techniques,” IAES International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 627–640, Jun. 2023, doi: 10.11591/ijai.v12.i2.pp627-640.

[45] A. Reghunath, S. V. Nair, and J. Shah, “Deep learning based Customized Model for Features Extraction,” in Proceedings of the 4th International Conference on Communication and Electronics Systems, ICCES 2019, 2019. doi: 10.1109/ICCES45898.2019.9002299.

[46] P. N. Maraskolhe and A. S. Bhalchandra, “Analysis of Facial Expression Recognition using Histogram of Oriented Gradient (HOG),” 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Jun. 2019, doi: 10.1109/iceca.2019.8821814.

[47] R. C. Ng, K. M. Lim, C. P. Lee, and S. F. A. Razak, "Surveillance system with motion and face detection using histograms of oriented gradients," Indonesian Journal of Electrical Engineering and Computer Science, vol. 14, no. 2, 2019, doi: 10.11591/ijeecs.v14.i2.pp869-876.

[48] A. Nandi, P. Dutta, and Md Nasir, "Automatic Facial Expression Recognition Using Histogram Oriented Gradients (HoG) of Shape Information Matrix," Advances in Intelligent Systems and Computing, pp. 343–351, 2020, doi: 10.1007/978-981-15-1084-7_33.

[49] M. H. Mutar, E. H. Ahmed, M. R. M. ALsemawi, H. O. Hanoosh, and A. H. Abbas, "Ear recognition system using random forest and histograms of oriented gradients techniques," Indonesian Journal of Electrical Engineering and Computer Science, vol. 27, no. 1, 2022, doi: 10.11591/ijeecs.v27.i1.pp181-188.

[50] A. Decelle, "An Introduction to Machine Learning: a perspective from Statistical Physics," Physica A: Statistical Mechanics and its Applications, p. 128154, Sep. 2022, doi: 10.1016/j.physa.2022.128154.

[51] H. Khdair and N. M. Dasari, "Exploring Machine Learning Techniques for Coronary Heart Disease Prediction," International Journal of Advanced Computer Science and Applications, vol. 12, no. 5, 2021, doi: 10.14569/IJACSA.2021.0120505.

[52] A. Aldahiri, B. Alrashed, and W. Hussain, "Trends in Using IoT with Machine Learning in Health Prediction System," Forecasting, vol. 3, no. 1, 2021, doi: 10.3390/forecast3010012.

[53] D. Koblah et al., "A Survey and Perspective on Artificial Intelligence for Security-Aware Electronic Design Automation," ACM Transact Des Autom Electron Syst, vol. 28, no. 2, 2023, doi: 10.1145/3563391.

[54] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning-Based Text Classification," ACM Computing Surveys,

vol. 54, no. 3. 2021. doi: 10.1145/3439726.

[55] Y. Zhao, Q. Chen, W. Cao, W. Jiang, and G. Gui, “Deep Learning Based Couple-like Cooperative Computing Method for IoT-based Intelligent Surveillance Systems,” in IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, 2019. doi: 10.1109/PIMRC.2019.8904229.

[56] H. R. Baghaee, D. Mlakic, S. Nikolovski, and T. Dragicevic, “Support Vector Machine-Based Islanding and Grid Fault Detection in Active Distribution Networks,” IEEE J Emerg Sel Top Power Electron, vol. 8, no. 3, 2020, doi: 10.1109/JESTPE.2019.2916621.

[57] M. Shatnawi, N. Almenhali, M. Alhammadi, and K. Alhanaee, “Deep Learning Approach for Masked Face Identification,” International Journal of Advanced Computer Science and Applications, vol. 13, no. 6, 2022, doi: 10.14569/ijacsa.2022.0130637.

[58] F. J. Díaz-Pernas, M. Martínez-Zarzuela, D. González-Ortega, and M. Antón-Rodríguez, “A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network,” Healthcare (Switzerland), vol. 9, no. 2, Feb. 2021, doi: 10.3390/healthcare9020153.

[59] F. Alrasheedi, X. Zhong, and P. C. Huang, “Padding Module: Learning the Padding in Deep Neural Networks,” IEEE Access, vol. 11, pp. 7348–7357, 2023, doi: 10.1109/ACCESS.2023.3238315.

[60] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” IEEE Trans Neural Netw Learn Syst, pp. 1–21, Jun. 2021, doi: 10.1109/tnnls.2021.3084827.

[61] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D convolutional neural networks and applications: A survey,” Mech Syst Signal Process, vol. 151, 2021, doi: 10.1016/j.ymssp.2020.107398.

[62] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, “Evolving Deep Convolutional Neural Networks for Image Classification,” IEEE Transactions on Evolutionary Computation, vol. 24, no. 2, 2020, doi: 10.1109/TEVC.2019.2916183.

[63] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” Artif Intell Rev, vol. 53, no. 8, 2020, doi: 10.1007/s10462-020-09825-6.

[64] R. A. Pratiwi, S. Nurmaini, D. P. Rini, M. N. Rachmatullah, and A. Darmawahyuni, “Deep ensemble learning for skin lesions classification with convolutional neural network,” IAES International Journal of Artificial Intelligence, vol. 10, no. 3, 2021, doi: 10.11591/ijai.v10.i3.pp563-570.

[65] S. Sharma, S. Sharma, and A. Athaiya, “ACTIVATION FUNCTIONS IN NEURAL NETWORKS,” International Journal of Engineering Applied Sciences and Technology, vol. 04, no. 12, 2020, doi: 10.33564/ijeast.2020.v04i12.054.

[66] X. Zhang, Y. Zou, and W. Shi, “Dilated convolution neural network with LeakyReLU for environmental sound classification,” in International Conference on Digital Signal Processing, DSP, 2017. doi: 10.1109/ICDSP.2017.8096153.

[67] Y. Gao, W. Liu, and F. Lombardi, “Design and Implementation of an Approximate Softmax Layer for Deep Neural Networks,” 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Oct. 2020, doi: 10.1109/iscas45731.2020.9180870.

[68] D. Zhu, S. Lu, M. Wang, J. Lin, and Z. Wang, “Efficient Precision-Adjustable Architecture for Softmax Function in Deep Learning,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 12, 2020, doi: 10.1109/TCSII.2020.3002564.

[69] F. Es-Sabery, A. Hair, J. Qadir, B. Sainz-De-Abajo, B. Garcia-Zapirain, and I. Torre-DIez, "Sentence-Level Classification Using Parallel Fuzzy Deep Learning Classifier," IEEE Access, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3053917.

[70] M. A. Saleem, N. Senan, F. Wahid, M. Aamir, A. Samad, and M. Khan, "Comparative Analysis of Recent Architecture of Convolutional Neural Network," 2022, doi: 10.1155/2022/7313612.

[71] D. Lee and K. Myung, "ADAM: Method for Stochastic Optimization," 2017 IEEE International Conference on Consumer Electronics, ICCE 2017, 2017.

[72] D. Yi, J. Ahn, and S. Ji, "An effective optimization method for machine learning based on ADAM," Applied Sciences (Switzerland), vol. 10, no. 3, 2020, doi: 10.3390/app10031073.

[73] W. E. L. Ilboudo, T. Kobayashi, and K. Sugimoto, "Robust Stochastic Gradient Descent with Student-t Distribution Based First-Order Momentum," IEEE Trans Neural Netw Learn Syst, vol. 33, no. 3, 2022, doi: 10.1109/TNNLS.2020.3041755.

[74] R. Tiwari, "Stabilizing the training of deep neural networks using Adam optimization and gradient clipping," International Journal of Scientific Research In Engineering and Management, vol. 07, no. 01, 2023, doi: 10.55041/ijsrem17594.

[75] Y. Wang, Z. Xiao, and G. Cao, "A convolutional neural network method based on Adam optimizer with power-exponential learning rate for bearing fault diagnosis," Journal of Vibroengineering, vol. 24, no. 4, 2022, doi: 10.21595/jve.2022.22271.

[76] P. Mishra and K. Sarawadekar, "Polynomial Learning Rate Policy with Warm Restart for Deep Neural Network," in IEEE Region 10 Annual International Conference, Proceedings/TENCON, 2019. doi: 10.1109/TENCON.2019.8929465.

[77] K. Nakamura, S. Soatto, and B. W. Hong, “Stochastic batch size for adaptive regularization in deep network optimization,” Pattern Recognit, vol. 129, 2022, doi: 10.1016/j.patcog.2022.108776.

[78] I. Kandel and M. Castelli, “The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset,” ICT Express, vol. 6, no. 4, 2020, doi: 10.1016/j.icte.2020.04.010.

[79] X. Zhang, H. Tian, Z. Wang, and S. Nie, “Segmentation of brain glioma on MRI using multiple densely connected 2D-CNNs,” Guangxue Jishu/Optical Technique, vol. 46, no. 5, 2020.

[80] M. Nakano and D. Sugiyama, “Discriminating seismic events using 1D and 2D CNNs: applications to volcanic and tectonic datasets,” Earth, Planets and Space, vol. 74, no. 1, 2022, doi: 10.1186/s40623-022-01696-1.

[81] M. A. Gómez-Guzmán et al., “Classifying Brain Tumors on Magnetic Resonance Imaging by Using Convolutional Neural Networks,” Electronics (Switzerland), vol. 12, no. 4, 2023, doi: 10.3390/electronics12040955.

[82] V. H. Phung and E. J. Rhee, “A High-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets,” Applied Sciences (Switzerland), vol. 9, no. 21, 2019, doi: 10.3390/app9214500.

[83] P. Xu, Z. Guo, L. Liang, and X. Xu, “MSF-net: Multi-scale feature learning network for classification of surface defects of multifarious sizes,” Sensors, vol. 21, no. 15, 2021, doi: 10.3390/s21155125.

[84] V. B. Codreanu, S. Bv, S. Aigner, V. Weinberg, D. Podareanu, and G. Caspar Van Leeuwen, “Best Practice Guide - Deep Learning,” 2019, doi: 10.13140/RG.2.2.31564.05769.

[85] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, “Improved handwritten digit recognition using convolutional neural networks (Cnn),” Sensors (Switzerland), vol. 20, no. 12, pp. 1–18, Jun. 2020, doi:

10.3390/s20123344.

[86] R. D. Rakshit, D. R. Kisku, P. Gupta, and J. K. Sing, “Cross-resolution face identification using deep-convolutional neural network,” Multimed Tools Appl, vol. 80, no. 14, 2021, doi: 10.1007/s11042-021-10745-y.

[87] Y. S. Teo et al., “Benchmarking quantum tomography completeness and fidelity with machine learning,” New J Phys, vol. 23, no. 10, 2021, doi: 10.1088/1367-2630/ac1fcb.

[88] A. Kost, W. A. Altabey, M. Noori, and T. Awad, “Applying neural networks for tire pressure monitoring systems,” SDHM Structural Durability and Health Monitoring, vol. 13, no. 3, 2019, doi: 10.32604/sdhm.2019.07025.

[89] X. Zhang, Y. Wang, N. Zhang, D. Xu, and B. Chen, “Research on scene classification method of high-resolution remote sensing images based on RFPNet,” Applied Sciences (Switzerland), vol. 9, no. 10, 2019, doi: 10.3390/app9102028.

[90] S. Montaha, S. Azam, A. K. M. R. H. Rafid, M. Z. Hasan, A. Karim, and A. Islam, “TimeDistributed-CNN-LSTM: A Hybrid Approach Combining CNN and LSTM to Classify Brain Tumor on 3D MRI Scans Performing Ablation Study,” IEEE Access, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3179577.

[91] J. Trelinski and B. Kwolek, “Deep Embedding Features for Action Recognition on Raw Depth Maps,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2021. doi: 10.1007/978-3-030-77967-2_9.

[92] Y. Dai, Y. Wang, M. Leng, X. Yang, and Q. Zhou, “LOWESS smoothing and Random Forest based GRU model: A short-term photovoltaic power generation forecasting method,” Energy, vol. 256, 2022, doi: 10.1016/j.energy.2022.124661.

[93] A. Bibi, “Spam Mail Scanning Using Machine Learning Algorithm,” J Comput (Taipei), vol. 15, no. 2, 2020, doi: 10.17706/jcp.15.2.73-84.

[94] H. M. Fadhil, M. N. Abdullah, and M. I. Younis, “A Framework for Predicting Airfare Prices Using Machine Learning,” Iraqi Journal of Computers, vol. 22, no. 3, 2022, doi: 10.33103/uot.ijccce.22.3.8.

[95] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, “CREMA-D: Crowd-sourced emotional multimodal actors dataset,” IEEE Trans Affect Comput, vol. 5, no. 4, pp. 377–390, Oct. 2014, doi: 10.1109/TAFFC.2014.2336244.

Appendix A

The Published Paper

Impact of Features Extraction Technique on Emotion Recognition Using Deep Learning Model

Anwar Salah
College of Information Technology, Software department, University of Babylon, Babylon, Iraq, Iraq

Nashwan Hussein
College of Information Technology, Software department, University of Babylon, Babylon, Iraq, Iraq

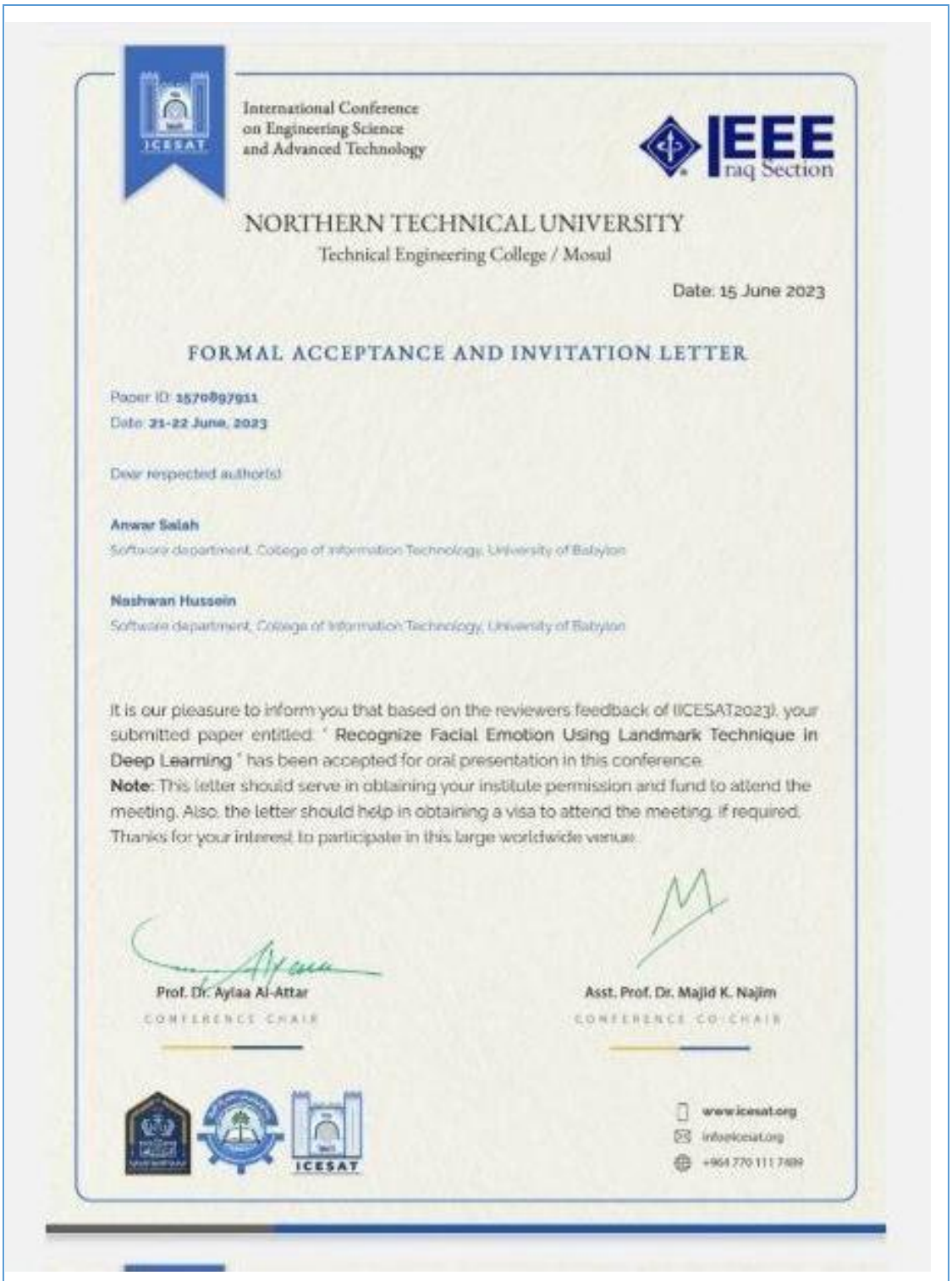
Keywords: Deep learning, Convolutional Neural Network (CNN), Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D), Histogram of Oriented Gradient (HOG).

ABSTRACT

A computer system would have a far harder time recognizing emotions from facial expressions than a human would. In a variety of settings, especially for human-computer interaction, the social signal processing sub-field of identifying emotions based on facial expressions is applied. Many studies have examined automatic emotion recognition, the majority of which make use of machine learning techniques. It remains a challenging issue in computer vision to recognize basic emotions including happiness, contempt, anger, fear, surprise, and sadness. Deep learning has received more attention recently as a possible option for a number of real-world problems, containing emotion recognition. In this paper, we proposed the usage of a 1-Diminsion Convolutional Neural Network (1D-CNN) to recognize some of the basic emotions and employed a different type of preprocessing and feature extraction ways to show how these methods impacted the performance of the proposed CNN model. The experiments on the Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D) revealed a high accuracy rate of 99.8%.

Appendix B

The Accepted Paper



International Conference
on Engineering Science
and Advanced Technology

IEEE Iraq Section

NORTHERN TECHNICAL UNIVERSITY
Technical Engineering College / Mosul

Date: 15 June 2023

FORMAL ACCEPTANCE AND INVITATION LETTER

Paper ID: 1570897911
Date: 21-22 June, 2023

Dear respected author(s)

Anwar Salah
Software department, College of Information Technology, University of Babylon

Nashwan Hussein
Software department, College of Information Technology, University of Babylon

It is our pleasure to inform you that based on the reviewers feedback of IICESAT2023), your submitted paper entitled: " Recognize Facial Emotion Using Landmark Technique in Deep Learning " has been accepted for oral presentation in this conference.
Note: This letter should serve in obtaining your institute permission and fund to attend the meeting. Also, the letter should help in obtaining a visa to attend the meeting, if required.
Thanks for your interest to participate in this large worldwide venue.

Prof. Dr. Aylaa Al-Attar
CONFERENCE CHAIR

Asst. Prof. Dr. Majid K. Najim
CONFERENCE CO-CHAIR

ICESAT

www.icesat.org
info@icesat.org
+964 770 111 7489

الخلاصة

يعد التعرف على تعبيرات الوجه ذا أهمية قصوى في فهم المشاعر الإنسانية وقد حظي باهتمام كبير في مجال رؤية الكمبيوتر والتعلم العميق. يعتبر اكتشاف تعابير الوجه في مقاطع الفيديو مهمة صعبة ومثيرة للاهتمام حيث أن الوجه هو وسيلة الاتصال الرئيسية والجزء الأكثر تواصلًا في الجسم لعرض المشاعر. تقدم الأطروحة الحالية بحثا مفصلا للكشف عن تعبيرات الوجه وتطبيقه على نظام قرآني مقترح يعتمد على العواطف. تهدف هذه الأطروحة إلى تطبيق أساليب التعلم العميق للتعرف على المشاعر التي تظهر في صور الوجه وتصنيفها بكفاءة. تستفيد هذه الأطروحة من قدرات الشبكات العصبية التلافيفية (CNN) (بعد واحد وبُعدين) والطبقات الموزعة زمنيًا من أجل التقاط التبعيات الزمنية بشكل فعال ضمن تسلسلات الفيديو في مجموعة بيانات CREMA-D. وقد خضع النظام لتدريب واختبار مكثف، مما أدى إلى مستوى عال من الدقة في تحديد ستة مشاعر أساسية: الغضب، والاشمئزاز، والخوف، والسعادة، والحياد، والحزن.

تعمل هذه الأطروحة على تحسين مجالها من خلال تطوير بنية جديدة للتعرف على المشاعر المعتمدة على الفيديو. تشتمل المعالجة المسبقة واستخراج الميزات والتصنيف باستخدام نماذج 1D-CNN و 2D-CNNعلى الإطار المعماري. يصنف نموذج 1D-CNNالميزات بعد استخلاصها بواسطة الرسم البياني للتدرجات الموجهة(HOG) ، بينما يتميز نموذج 2 D-CNN بالاستخراج والتصنيف في وقت واحد. تبلغ دقة نموذج 1D-CNN (0.99) مما يدل على أنه يحقق نتائج عالية. بالإضافة إلى ذلك، عملت شبكة CNN ثنائية الأبعاد بشكل رائع، حيث بلغت درجة دقتها (0.82). وتظهر هذه النتائج قدرة النظام على التعرف على تعابير الوجه واكتشاف الانفعالات، وتصميم أنظمة تعتمد على الانفعالات لاقتراح آية قرآنية.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات
قسم برمجيات

خوارزميات التعلم العميق للكشف عن الحالة المزاجية واقتراح الايات القرانية المناسبة

رسالة مقدمة
إلى مجلس كلية تكنولوجيا المعلومات في جامعة بابل والتي هي جزء من متطلبات الحصول على درجة الماجستير في تكنولوجيا المعلومات / البرمجيات

اعداد الطالبة
انوار صالح مشعان بريسم

اشراف
أ. م. د. نشوان جاسم حسين

1445 هـ **2023 م**