The University of Jordan

School of Engineering
Department of Computer Engineering

Real-Time Facial Emotion Recognition Using AI

Supervisor:
Prof. Gheith Abandah

Author(s):
Dana Nedal Ghazal 0183507
Rawan Rami Hamdan 0183609

June 18th, 2023

Submitted in partial fulfillment of the requirements of B.Sc. Degree in Computer Engineering

This page is intentionally left blank

ETHICAL STATEMENT

We, the undersigned students, certify and confirm that the work submitted in this project report is entirely our own and has not been copied from any other source. Any material that has been used from other sources has been properly cited and acknowledged in the report.

We are fully aware that any copying or improper citation of references / sources used in this report will be considered plagiarism, which is a clear violation of the Code of Ethics of the University of Jordan.

We further certify and confirm that we had no external help without the approval of our supervisor and proper acknowledgment when it is due. We certify and affirm that we never at any point commissioned a $3^{rd.}$ party to do the work or any part of it on our behalf regardless of the amount charged or lack thereof. We also acknowledge that if suspected and thereafter proven that we commissioned a $3^{rd.}$ party to do any part of this project that we risk failing the entire project.

We certify and confirm that all results presented in this project are true with no manipulation of data or fraud, that any statistics done, or surveys collected are conducted with the highest degree of scientific fidelity and integrity, and that if proven otherwise, we risk failing the entire project. We acknowledge that for any data collected, we have taken all the steps necessary in applying for proper authorizations if deemed necessary, and that all user data collected is subject to the utmost degrees of privacy and anonymity.

Dana Nedal Ghazal	Rawan Rami Hamdan
18/6/2023	18/6/2023
Signature:	Signature:

This page is intentionally left blank

SUPERVISOR CERTIFICATION

I hereby certify that the students in this project have **successfully finished** their senior year project and by submitting this report they have fulfilled in partial the requirements of B.Sc. Degree in Computer Engineering. ☐

I hereby certify that the students in this project have not completed their senior year graduation project and **I do not approve** that they proceed to the discussion. ☐

I suspect that the students have **violated** one or more of the clauses in the **ethical statement** and I suggest that an investigation committee look into the matter. ☐

Prof. / Dr. / Eng.

Signature:

This page is intentionally left blank

DEDICATION

We would like to dedicate our graduation report project to our families who have always been our biggest supporters. Their encouragement, love, and unwavering support have been instrumental in our academic success. We are grateful for their sacrifices, guidance, and the values they have instilled in us, which have helped shape us into the individuals we are today.

We would also like to express our appreciation to our professors and mentors who have challenged and inspired us throughout our academic journey. Their guidance, expertise, and encouragement have been invaluable in our development. We are grateful for the opportunities they have provided us, and we will always remember their contributions to our academic growth.

SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AdaBoost	Adaptive Boosting
Adagrad	Adaptive Gradient Descent
Adam	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
CK+	Extended Cohn-Kanade Dataset
CNN	Convolution Neural Network
Conv2D	2D Convolution Layer
DBN	Deep Belief Network
DL	Deep Learning
FER	Facial Emotion Recognition
FN	False Negative
FP	False Positive
GPU	Graphics Processing Units
HCI	Human-Computer Interaction
HR	Human Resources
IEFDB	Iranian Emotional Face Database
iOS	iPhone Operating System
JAFFE	Japanese Female Facial Expression dataset
JVM	Java Virtual Machine
LFW	Labelled Faces in the Wild
macOS	Mac Operating System
Maxpooling2D	Max Pooling Operation For 2D Spatial Data
MDD	Major Depressive Disorder
ML	Machine Learning
MLP	Multi-Layer Perceptron
NN	Neural Network
OpenCV	Open-Source Computer Vision
OS	Operating System
ReLU	Rectified Linear Unit
REST	Representational State Transfers
RMSprop	Root Mean Square Propagation
RNN	Recurrent Neural Network
SDK	Software Development Kit
TFLite	TensorFlow Lite
TL	Transfer Learning
TN	True Negative
TP	True Positive

TPU	Tensor Processing Units
TV	Television
UI	User Interface
XML	Extensible Markup Language

ABSTRACT

Facial emotion recognition (FER) is a technology that analyzes facial expressions from images or videos to identify an individual's emotional state. One of the most important challenges faced by current artificial intelligence (AI) based mobile applications in recognizing facial emotions is bias as there is a lack of representation of faces wearing hijabs and Middle Eastern faces in the databases of faces used in available applications, which leads to inaccurate results of these applications. The proposed project aims to build a real-time emotion analysis application focusing on Middle Eastern faces particularly women wearing hijabs to increase its usability in the Arab world.

The traditional method of emotion recognition may not meet the need of mobile application users for their emerging value-added services, so providing techniques for classifying facial expressions by taking advantage of the benefits of deep learning (DL) is important. We used DL model was trained using the FER2013 dataset and retrained it on a hybrid dataset that includes the Extended Cohn-Kanade (CK+) dataset, Japanese female facial expression (JAFFE) dataset, and Iranian emotional face database (IEFDB). We achieved a test accuracy of 88% on the hybrid test set and 90% test accuracy on the IEFDB test set.

Table of Contents

ETHICAL STATEMENT III
SUPERVISOR CERTIFICATION V
DEDICATION VII
SYMBOLS, ABBREVIATIONS, AND ACRONYMS VIII
ABSTRACT X
LIST OF FIGURES XIV
LIST OF TABLES XV
CHAPTER 1 INTRODUCTION 1
- **1.1. Problem Definition** 1
- **1.2. Project Importance** 2
- **1.3. Proposed Solution** 3
- **1.4. Project Deliverables** 4
- **1.5. Project Impact** 4
- **1.6. Report Outline** 5

CHAPTER 2 RELATED WORK 6
- **2.1. Facial Emotion Recognition Apps and Their Features** 6
  - **2.1.1. Cameralyze – Facial Emotion Recognition Solutions** 6
  - **2.1.2. Face Analyzer** 7
  - **2.1.3. CS230-FER** 8
  - **2.1.4. Emotimeter – Emotion Detector** 8
- **2.2. Facial Emotion Detection Models and APIs** 9
  - **2.2.1. CompreFace** 9
  - **2.2.2. FaceNet** 10
  - **2.2.3. FaceFirst** 10
  - **2.2.4. Kairos** 11
  - **2.2.5. Face++** 12
  - **2.2.6. BetaFace** 12
  - **2.2.7. Kaggle_Model** 13
- **2.3. Comparison Between the Facial Emotion Detection APIs** 13
- **2.4. Market Growth of Emotion Recognition Software** 14

CHAPTER 3 TECHNOLOGY BACKGROUND 15
- **3.1. Machine Learning in Facial Emotion Recognition** 15
- **3.2. Neural Network** 16

3.3. Deep Learning in Facial Expression Detection 17
3.3.1. Convolutional Neural Network 17
3.4. Transfer Learning 18
3.5. Hyperparameter Tuning 18
3.5.1. Learning Rate 18
3.5.2. Batch Size 18
3.5.3. Number of Epochs 18
3.5.4. Optimizer 19
3.5.5. Loss Function 19
3.6. Evaluation Metrics 19
3.6.1. Confusion Matrix 20
3.6.2. Accuracy 20
3.6.3. Precision 20
3.6.4. Recall 20
3.6.5. F1 Score 20
3.6.6. Loss 20
3.7. Operating Systems 21
3.8. Programming Languages 22
3.8.1. Python 22
3.8.2. Java 22
CHAPTER 4 PROGRESS AND IMPLEMENTATION 23
4.1. Preparation Process 23
4.2. Datasets 23
4.3. Datasets Preprocessing 25
4.4. Retraining and Testing the CNN Model 26
4.5. Mobile Application Development 26
4.5.1. App Design and How it Works 26
4.5.2. Convert Keras Model to TensorFlow Lite 27
4.5.3. Face Detection 28
CHAPTER 5 RESULTS AND DISCUSSION 29
5.1. Model Results 29
5.1.1. The Results of the Model Retrained on the IEFDB Dataset 29
5.1.2. The Results of the Model Retrained on the Hybrid Dataset 32
5.1.3. Discussion the Results 34
5.2. Application Results 35
CHAPTER 6 CONCLUSIONS AND FUTURE WORK 38

6.1. Conclusions 38
6.2. Future Work 38

LIST OF FIGURES

FIGURE 1 - FACIAL EXPRESSION CLASSIFICATION 3
FIGURE 2 - SCREENSHOT OF CAMERALYZE – FACIAL EMOTION RECOGNITION 6
FIGURE 3 - SCREENSHOTS OF FACE ANALYZER APPLICATION 7
FIGURE 4 - SCREENSHOTS OF CS230-FER MOBILE WEB APPLICATION 8
FIGURE 5 - SCREENSHOTS OF EMOTIMETER – EMOTION DETECTOR APPLICATION 8
FIGURE 6 - SCREENSHOT OF COMPREFACE SOFTWARE 9
FIGURE 7 - SCREENSHOTS OF FACENET SOFTWARE 10
FIGURE 8 - SCREENSHOT OF FACEFIRST SOFTWARE 11
FIGURE 9 - EMOTION DETECTION AND RECOGNITION MARKET 14
FIGURE 10 - ARTIFICIAL NEURAL NETWORK TOPOLOGY 16
FIGURE 11 - ARCHITECTURE OF BASIC CONVOLUTIONAL NEURAL NETWORK 17
FIGURE 12 - THE POWER OF USING ANDROID 21
FIGURE 13 - SAMPLE IMAGES FOR EACH EMOTION CLASS IN THE FER2013, CK+, JAFFE, AND IEFDB DATASETS 24
FIGURE 14 - THE DESIGN OF THE APPLICATION SCREENS 27
FIGURE 15 - THE STEPS AN IMAGE GOES THROUGH TO DETECT FACIAL EMOTIONS 28
FIGURE 16 - MODEL ACCURACY AND LOSS FOR TRAINING AND VALIDATION SETS FOR IEFDB DATASET 30
FIGURE 17 - CONFUSION MATRIX OF THE MODEL FOR IEFDB TEST SET 31
FIGURE 18 - MODEL ACCURACY AND LOSS FOR TRAINING AND VALIDATION SETS FOR THE HYBRID DATASET 33
FIGURE 19 - CONFUSION MATRIX OF THE MODEL FOR HYBRID TEST SET 33
FIGURE 20 - THE RESULTS OF EMOTION RECOGNITION IN TWO CASES OF TAKE PHOTO 35
FIGURE 21 - THE RESULTS OF REAL-TIME EMOTION RECOGNITION 36
FIGURE 22 - SELECT DELAY TIME AND SEND NOTIFICATION 37

LIST OF TABLES

TABLE 1 - COMPARISON BETWEEN THE FACIAL EMOTION DETECTION APIS 13
TABLE 2 - COMPARISON BETWEEN THE FACIAL EMOTION RECOGNITION DATASETS USED IN THIS PROJECT 24
TABLE 3 - COMPARISON BETWEEN THE NUMBER OF IMAGES FOR EACH EMOTION IN THE DATASETS 24
TABLE 4 - THE SPLIT VALUE AND NUMBER OF IMAGES FOR TRAIN, TEST, AND VALID SET OF IEFDB 25
TABLE 5 - THE SPLIT VALUE AND NUMBER OF IMAGES FOR TRAIN, TEST, AND VALID SET OF HYBRID DATASET 26
TABLE 6 - MODEL HYPERPARAMETERS RETRAINED ON THE IEFDB TRAIN SET 29
TABLE 7 - ACCURACIES OF THE MODEL RETRAINED ON THE IEFDB TRAIN SET 30
TABLE 8 - CLASSIFICATION RESULTS OF THE MODEL ON THE IEFDB TEST SET 31
TABLE 9 - MODEL HYPERPARAMETERS RETRAINED ON THE HYBRID TRAIN SET 32
TABLE 10 - ACCURACIES OF THE MODEL RETRAINED ON THE HYBRID TRAIN SET 32
TABLE 11 - CLASSIFICATION RESULTS OF THE MODEL ON THE HYBRID TEST SET 34

This page is intentionally left blank

CHAPTER 1
INTRODUCTION

The rapid growth of artificial intelligence (AI) has contributed a lot to the world of technology. Since traditional algorithms fail to meet human needs in real-time, machine learning (ML) and deep learning (DL) algorithms have gained great success in various applications such as classification systems, recommendation systems, pattern recognition, *etc*. [1].

Facial emotion recognition (FER) is a technology that analyzes facial expressions from images or videos in order to reveal an individual's emotional state. Emotion plays a vital role in determining a person's thoughts, behavior, and feeling. In order to maximize user satisfaction, recognizing user emotion is the most important. Recognizing emotions has become an essential part of providing emotional care for people. Providing emotional care can greatly improve users' experience to improve quality of life. The traditional method of emotion recognition may not meet the need of mobile application users for their emerging value-added services, so providing techniques for classifying facial expressions by taking advantage of the benefits of DL is important [2, 3, 4].

Automatic analysis of facial expressions has attracted the interest of many AI researchers because these systems will have many uses in the fields of healthcare, gaming, and human-computer interaction (HCI) [5]. But this type of application has spread quite a bit in the Middle East and this is what led us to the idea of our project, where we wanted to build a real-time emotion analysis application focusing on Middle Eastern faces where women wear hijab to increase its usability in the Arab world, especially in Jordan.

1.1. Problem Definition

Facial expressions are the most commonly used form of nonverbal communication. As a result, the inability to perceive these expressions leads to a loss of communication effectiveness. Identifying facial expressions as representative of particular emotions can be challenging even for humans. Studies show that different people recognize different emotions in the same facial expressions. There are a lot of factors that make emotional recognition difficult such as psychological, there are people who have poor perception of facial expressions, and this deficit is common in patients with major depressive disorder (MDD), schizophrenia, autism, and alexithymia, which leads to poor communication [6, 7, 8].

The recent pandemic that affected the entire world has transformed teaching and learning environments from physical contact to virtual reality. This made a challenge in how to be able to understand whether students are able to absorb the content in an online classroom scenario. Most of the applications used such as Zoom and others have the challenges of capturing students affect, which is vital information in achieving class dynamics and effective teaching. This entails finding a solution to meet these needs by recognizing facial emotions by applying DL algorithms [9].

This is where AI comes in. AI can detect emotions by learning the meaning of each facial expression and applying that knowledge to new information presented to it. Emotional AI is a technology capable of reading, imitating, interpreting and responding to human facial expressions and emotions. DL is an AI facial expression recognition function that works like the human brain by processing data and developing patterns used to discover objects and even make decision [10].

Several AI-based mobile applications have been made available for facial emotion recognition, but it turns out that they have some issues that we will overcome in our project, most notably: bias and inaccuracy. Bias is based on race, age group, gender, education, or some other demographic. The faces of many ethnicities are still underrepresented in the databases of faces used in current applications, as they do not contain images of Middle Eastern faces and images of women wearing hijabs, which may affect the accuracy of use these applications in Arab and Islamic countries [11, 12].

1.2. Project Importance

Facial expression recognition is an important topic, with results from Mehran's study demonstrated that 7% of information is conveyed between people via writing, 38% through conversation, and 55% through facial expression. The first indication a person gives of their emotional state being in a state is facial expressions, so many researchers are very interested in automatic facial expression recognition in recent years [13].

Facial expressions are perhaps the most important signal because they tell us about people's personalities, emotions, motivations, or intentions. They are not just signs of people's inner states; They are also signals to others to behave in certain ways, and to provide messages of social coordination. They are also important signals for socialization and education. When we learn about someone's facial expressions, we know the direction of their thoughts, how their minds handle something, and what their bodies are prepared to do before they do [14].

Thus, all this information that we can read from facial expressions of emotions allows us to gain insight into the minds of the people we interact with, even when they are not talking. It allows us to draw reasonable conclusions and testable hypotheses about people's personalities, motives, and intentions, which are very important for many professionals [14].

Facial expressions help us make sense of movies or television (TV) shows. We can interpret the context of the movie by observing the facial expressions and gestures of the characters. Facial expressions also have a long-lasting effect on memory and cognition. People remember facial expressions even after several days [6, 15].

Facial emotion detection and recognition technology is important in a wide range of applications, as it can be used to test video games. Video games evoke different emotions in players, and the reaction is usually difficult to assemble and pinpoint accurately, for testing and optimization, thus facial emotion detection can help analyze player's feelings in real-time while they are playing [12].

FER also applies to some psychological fields such as; neuropsychiatric disorders, rehabilitation, depression and autism. As the available methods used by doctors in these fields are qualitative manual methods, which are more subjective and human intensive. This challenge requires FER technology that is likely to reduce extensive human efforts and provide a qualitative result, by incorporating FER into homes to monitor people's psychological problems, for example, it can detect the first signs of sadness or depression, which are a major health concern for older people living alone [9, 17].

Emotion detection technology enhances the user's emotional awareness, as this can help people determine what they are feeling, how they regulate their behavior, and what they should do about those feelings. So, a lack of this quality can prove to be quite debilitating to the social and emotional health of a person [17].

1.3. Proposed Solution

In this project, we aim to build an easy-to-use Android app that that detects emotions by taking a photo, selecting one from the gallery or in real-time. The app works to provide the user's emotional state and some recommendations and suggestions to the user, such as: a suggestion to take a picture in a state of happiness and a suggestion to exercise in a state of anger and others services that increase user awareness of how to act positively while feeling. In addition to providing a link to a video that the user can watch to help him improve his mood and overall well-being.

In this project, we will focus on analyzing the expression of faces in real-time, especially Middle Eastern faces, and the faces of women who wear the hijab, we will use AI and DL algorithm to recognize human emotions and classify them into sadness, happiness, anger, fear, disgust, surprise, and neutral. As we can see in Fig. 1 [18] through the movement of facial muscles we can recognize and differentiate emotions based on that movement. We will develop our AI model using the convolution neural network (CNN) algorithm. We will use Android to develop our mobile application because it is one of the most popular operating system (OS) in general, especially in the Middle East region, which most of the people can use easily.

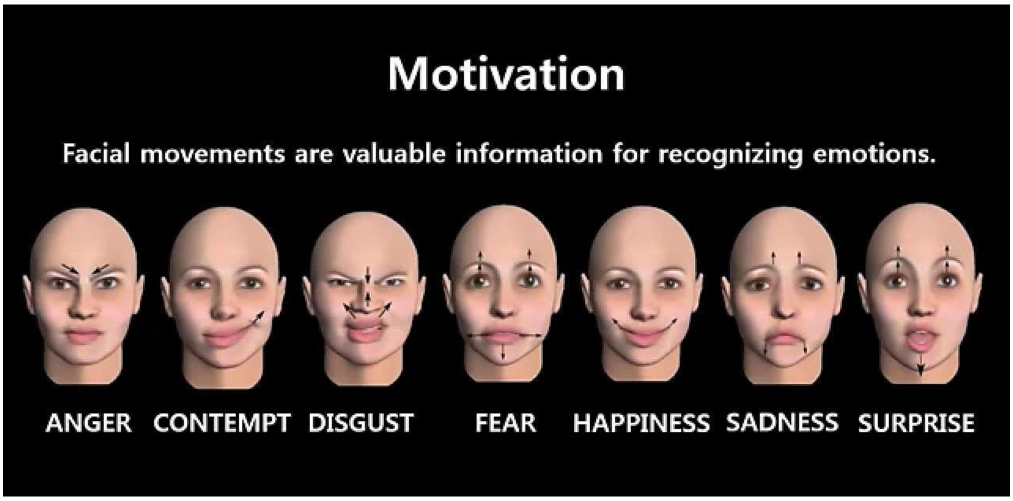


Figure 1 - Facial Expression Classification

1.4. Project Deliverables

At the end of the real-time facial emotion recognition project, we will hand over a mobile application to the project committee. The software will be implemented using the latest ML technology and will be able to accurately detect and recognize a range of emotions in real-time. The program will include a user-friendly interface that will allow users to easily interact with the application and display the detected emotions by taking a photo, choosing one from the gallery or in real-time, the photo will be processed then the app will show the user emotion and best advice for him, also a video that the user can watch to help him enhance his feeling. The app will be designed to run efficiently on Android mobile devices. In addition, the application will be scalable, allowing additional features and functionality to be added over time. We hope in the future that our application will be used in the Arab countries in several potential fields such as psychology, marketing, entertainment and education.

1.5. Project Impact

Emotions are a window to the human soul, and they cannot be hidden, no matter how hard we try. It is important to know how to express and understand feelings correctly. Therefore, FER plays an essential role in human-to-human communication and human-machine interaction. Based on the analysis of facial expressions, the machine can understand the emotional state of a person and take appropriate actions.

FER systems are important due to their impact on the field of learning, employment, marketing, automatic cars, *etc*. They are used to help develop the economy, in enhancing social communication, and in caring for people. As in healthcare, FER systems are used to monitor patients' facial expressions and detect their feelings, and based on the results they are given appropriate treatment. Therefore, care robots equipped with the feature of recognizing facial expressions take care of patients in hospitals and at home, which helps reduce the burden on doctors. As biometric technology becomes more affordable, its adoption in the healthcare sector is sure to increase [19].

In the automotive field, auto manufacturers focus on making cars that are safe to drive by using FER systems to understand human emotions. This ensures the safety of the driver by analyzing his facial expressions to detect fatigue or drowsiness and send alerts such as: The system prompts the driver to stop for a while, or take a coffee break, or play music [19].

In market research with the arrival of FER systems powered by AI, it has become easier for market research companies to automate video analysis and detect the facial expressions of users when interacting with a specific product. FER systems help market research companies expand their data collection efforts instead of using paper questionnaires to understand customer requirements and needs and know their satisfaction with a particular product. This saves time and labor, reduces cost and increases sales as well [19].

In job interviews, an interview is an important way to interact with candidates and understand whether they are suitable for the position. However, it is not always possible to analyze the candidate's personality in such a short period, so sentiment detection can assess and measure the candidate's feelings through facial expressions. It helps interviewers understand the candidate's temperament and personality traits and accordingly human resources (HR) can take advantage of this technology to create recruitment strategies and design policies for optimizing employee performance [19].

In social life, the recognition of facial expressions plays a major role in social communication with others. There are patients with impairment in recognizing facial expressions because they suffer from MDD, which leads to poor communication and a change in their adaptive behaviors, which affects their lives negatively. FER systems help these patients to understand the emotions of others and facilitate communication with them [7].

In the education sector, FER technology is used by monitoring students' attention by capturing their facial expressions and analyzing them to ensure that they are paying attention and can understand the content of a lecture or have difficulty understanding and solving a specific problem, and to assess student interests by detecting their expressions when reading specific content. As the education system is rapidly moving from the physical to the virtual environment, so at that time facial expression and emotion detection can play a vital role in terms of better understanding of a student [20].

1.6. Report Outline

In this report, after we have introduced the concept of our project, we will move on to discuss some of the popular applications, models, and application programming interface (API) related to our project in Chapter 2. Next, in Chapter 3 we will provide a background information on the technologies we used in our project. Subsequently, we will go over the processes we undertook and explain the design of our application in Chapter 4. Then we will present and discuss the results of the ML model in Chapter 5. Finally, in Chapter 6 we will summarize the project and its results and we will suggest ideas for future development of the project.

CHAPTER 2
RELATED WORK

In this chapter, we will review related work in four sections. In section 2.1, we will show examples of applications built for facial emotion recognition, their main features, and their datasets. Then in sections 2.2 and 2.3, respectively. We will explore some of emotion recognition models and APIs, and compare the accuracy of each. Finally, we will end this chapter with section 2.4 by showing the growth of the market in emotion recognition software.

2.1. Facial Emotion Recognition Apps and Their Features

Many applications currently exist for the process of facial emotion recognition. In this section, we will show some of them with their features and screenshots for each application.

2.1.1. Cameralyze – Facial Emotion Recognition Solutions

Cameralyze offers the most accurate and automated system for recognizing human emotions from facial expressions, many specific characteristics in facial photos, videos, or live broadcasts, including universal expressions: happy, sad, angry, surprised, afraid, calm, and confused using AI. It is a fast, flexible, accurate and easy-to-use platform. With its drag-and-drop technology, it can instantly analyze your data, saving valuable time [21].

Cameralyze offers higher performance, and the response time is only a few milliseconds. When comparing the platform's results to facial expressions manually recorded by professional commentators, Cameralyze scores an accuracy score of up to 99%, depending on the emotion being measured. It is also easily scalable as it provides automatic load balancing and synchronization. It processes thousands of videos simultaneously and detects facial emotions in less than a second, and allows you to monitor metrics in real-time. It also provides high privacy by encrypting data, so no one can access your data [21]. Fig. 2 [21] shows how Cameralyze detects facial emotions.

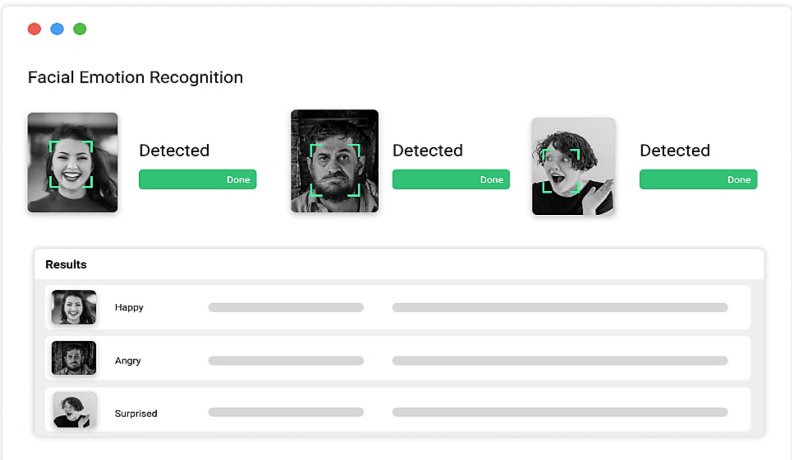


Figure 2 - Screenshot of Cameralyze – Facial Emotion Recognition

2.1.2. Face Analyzer

Face Analyzer is Android app that use the Microsoft Face API to not only detect individual faces in an image, but also provide information about facial attributes for each face such as emotions, estimated age, and gender. Possible applications for using this app are in amusement parks, classrooms, and residential homes [22].

Amusement parks can use the app to gather data about the type of crowds they collect based on age and other attributes as well as analyze people's feelings before and after the rides. Moreover, the app can be used in classrooms to analyze students' faces while teaching. The teacher can then review the data on emotions to see if students are able to understand, enjoy, or dislike the lesson. Finally, another application of the app is in residential homes where caretakers can regularly use the app to identify patients' feelings and store them in a database for later analysis [22].

The application is simple enough to use as it consists of two screens as shown in Fig. 3 [23]. The first screen has two buttons, one to take the photo and the other to process the photo. Hence, the app requires the camera permission. Once the photo is taken, you can hit the process button and the app will use AsyncTask and the Microsoft Face API to detect faces in the photo and get information about facial attributes such as age, gender, and emotions. Once the image has been processed, it takes you to a second screen, where it creates a thumbnail of the individual, and displays the analyzed information relative to each person's face next to it [22].

The disadvantage of the Face Analyzer app is that it is not 100% accurate and may misidentify age, gender and emotion. The accuracy can be affected by various factors, such as the quality of the input image, lighting conditions, and facial expression. Therefore, the results of the analysis not always be accurate, and the app does not detect emotions in real-time. Also, the app requires users to upload images of themselves, which can raise privacy concerns. These images could be stored and used for other purposes without the user's knowledge. The model used in the app may be biased towards certain facial emotions or misidentify emotions for people of certain ethnic or gender groups, this could lead to unfair results.

Figure 3 - Screenshots of Face Analyzer Application

2.1.3. CS230-FER

The CS230-FER is a mobile web app that uses FER model to analyze emotions and is designed to be user-friendly and easy to navigate. Hosted on Firebase that uses TensorFlow.js, React.js, and face-api.js to detect, crop, and resize a user's face. It also has low memory requirements and high prediction speeds. The five-layer CNN model trained on the FER2013 dataset, Extended Cohn-Kanade (CK+) dataset, and Japanese female facial expression (JAFFE) dataset, which achieved 75.8% test accuracy, outperforming the highest reported 75.2% test accuracy in published works. Additionally, the web app test set achieved an accuracy of 69.8%, with a recognition speed of 40ms [24]. See Fig. 4 [24] which displays the web app screen that has a single button, when you take picture, it shows your emotion.

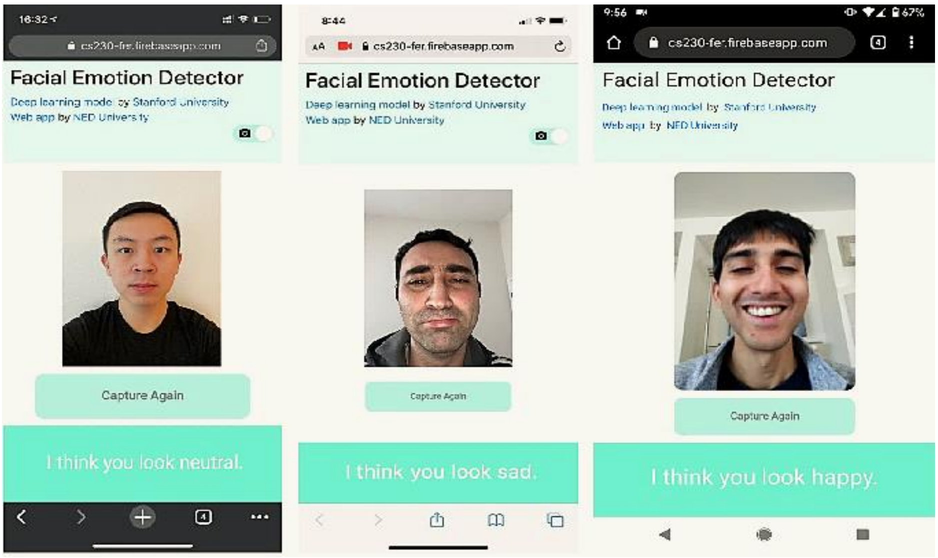


Figure 4 - Screenshots of CS230-FER Mobile Web Application

2.1.4. Emotimeter – Emotion Detector

Emotimeter is a free Android app based on FER. It can detect emotions from facial expressions using cutting edge ML technologies. It can detect emotions in real-time from the images obtained by the camera, or analyze images or videos from the gallery to detect the emotions of all the people in it [25]. Fig. 5 [25] shows how the app detects user emotions.

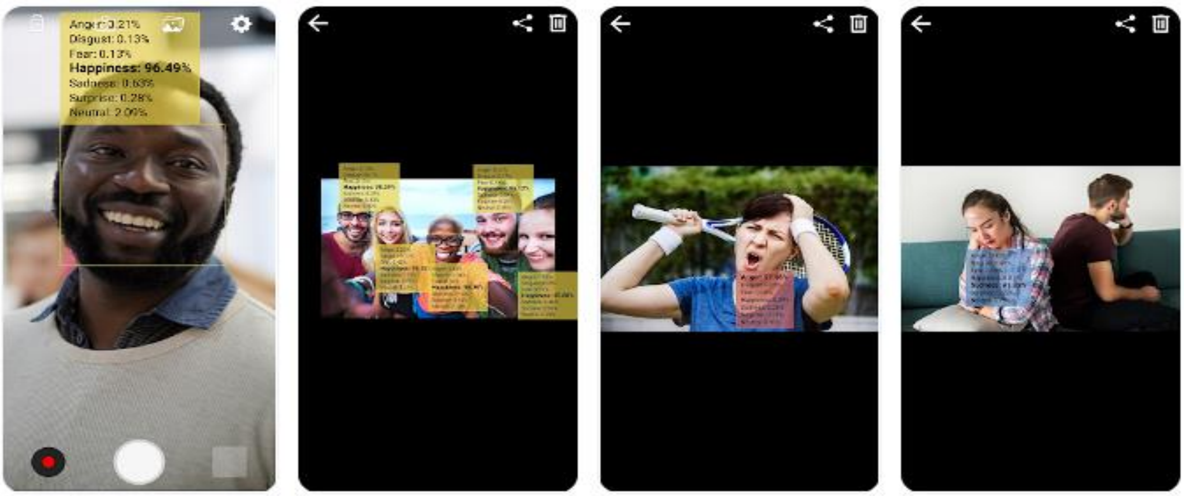


Figure 5 - Screenshots of Emotimeter – Emotion Detector Application

2.2. Facial Emotion Detection Models and APIs

There is a lot of models and APIs available online that parallel the human ability to discern emotional gestures. Let's explore some of emotion recognition models and APIs that can be used to interpret a user’s emotion from photos and videos, especially the kaggle_model that we used in this project [26].

2.2.1. CompreFace

CompreFace is one of the few self-hosted representational state transfers (REST) API face recognition solutions on this list. The solution is scalable, so you can simultaneously recognize faces on several video streams. CompreFace also has a straightforward user interface (UI) for managing user roles and face collections. CompreFace shows state-of-the-art accuracy on the labelled faces in the wild (LFW) dataset 99.83% [27, 28].

CompreFace is designed to be able to recognize both photographs and videos of individuals. This software is very scalable. The software is able to determine the gender or ethnicity of an individual. CompreFace was created with the intention of allowing consumers to make more informed decisions about their purchases. This program more accurate than other similar software on the market at this time [27, 28].

CompreFace API there are also some potential disadvantages to consider, including privacy concerns, can be used to collect and analyze data about individuals without their knowledge or consent, which can be a violation of their privacy rights. Also, accuracy for CompreFace API can struggle to recognize faces in low-light environments, when the face is partially obscured, or when the individual is wearing a mask. This can lead to false positives or false negatives, which can have serious consequences. Finally, CompreFace API is a powerful tool, but integrating it into existing systems can be complex and time-consuming. It may require changes to existing workflows and infrastructure, which can be a significant investment of time and resources. See Fig. 6 [29] that show screenshot of CompreFace software.

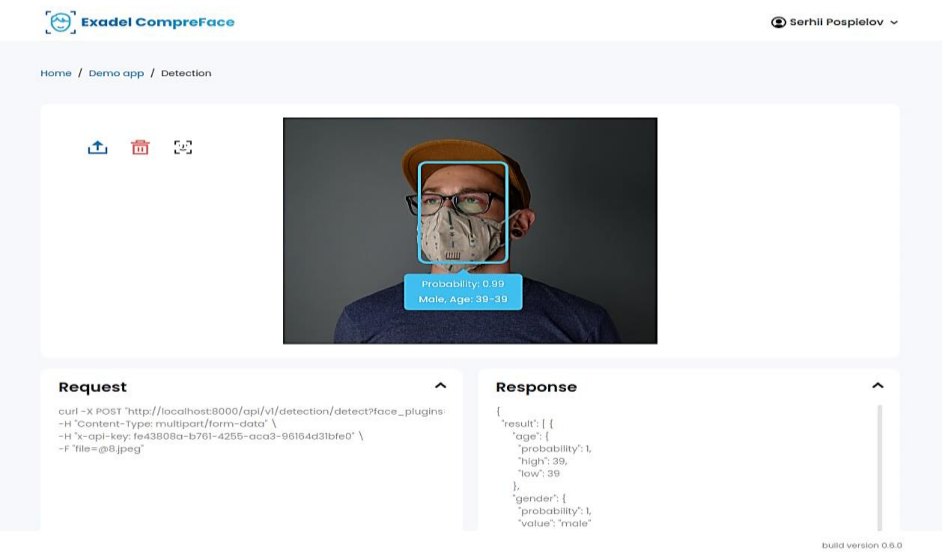


Figure 6 - Screenshot of CompreFace Software

2.2.2. FaceNet

FaceNet is a face recognition method created by google researchers and the open-source python library that implements it. Some researchers use it most recently for face recognition in masks. The accuracy of this method is quite high 99.65% on the LFW dataset [27].

One disadvantage of the FaceNet API is that it requires a significant number of computational resources to operate efficiently. FaceNet uses deep neural networks, which are computationally intensive and require powerful hardware such as graphics processing units (GPU) or tensor processing units (TPU) to achieve real-time performance. This can limit the scalability of the system, particularly when processing large volumes of data or operating in resource-constrained environments. Additionally, the accuracy of FaceNet API can be affected by factors such as lighting conditions, pose variations, and occlusions, which can lead to false positives or false negatives in face recognition. Therefore, additional preprocessing and tuning may be required to ensure reliable performance in different environments. Fig. 7 [30] shows screenshots of FaceNet software.

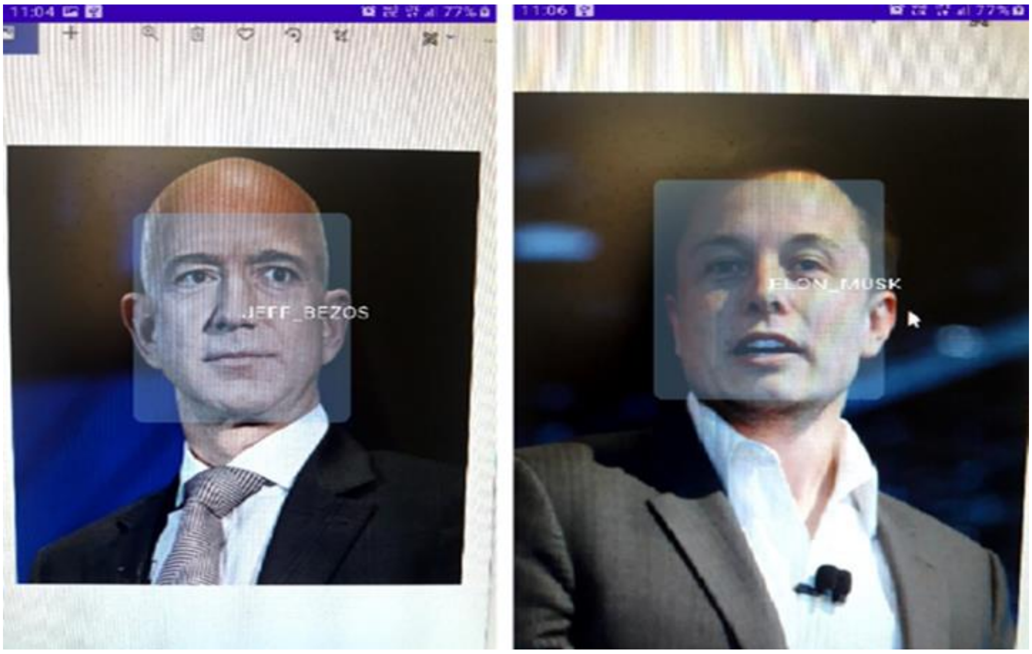


Figure 7 - Screenshots of FaceNet Software

2.2.3. FaceFirst

FaceFirst is highly accurate, scalable, secure and private, offers a very wide range of products, such as smart motion detectors that will track faces, smart cameras with high resolution and face recognition capabilities, and other security products [28].

Some potential disadvantages of using the FaceFirst API include privacy concerns, facial recognition technology can be considered an invasion of privacy, and some people may feel uncomfortable having their photos captured and stored in a database. Another disadvantages accuracy issues there are still some concerns about accuracy, especially when it comes to identifying people with darker skin or facial hair.

Also, security risks If the FaceFirst API is not properly secured, there is a risk that malicious actors can gain access to the system and use it for nefarious purposes. Finally, cost FaceFirst API can be costly for some companies or organizations, especially small businesses that may not have the budget to invest in this type of technology. See Fig. 8 [28] that show screenshot of FaceFirst software.

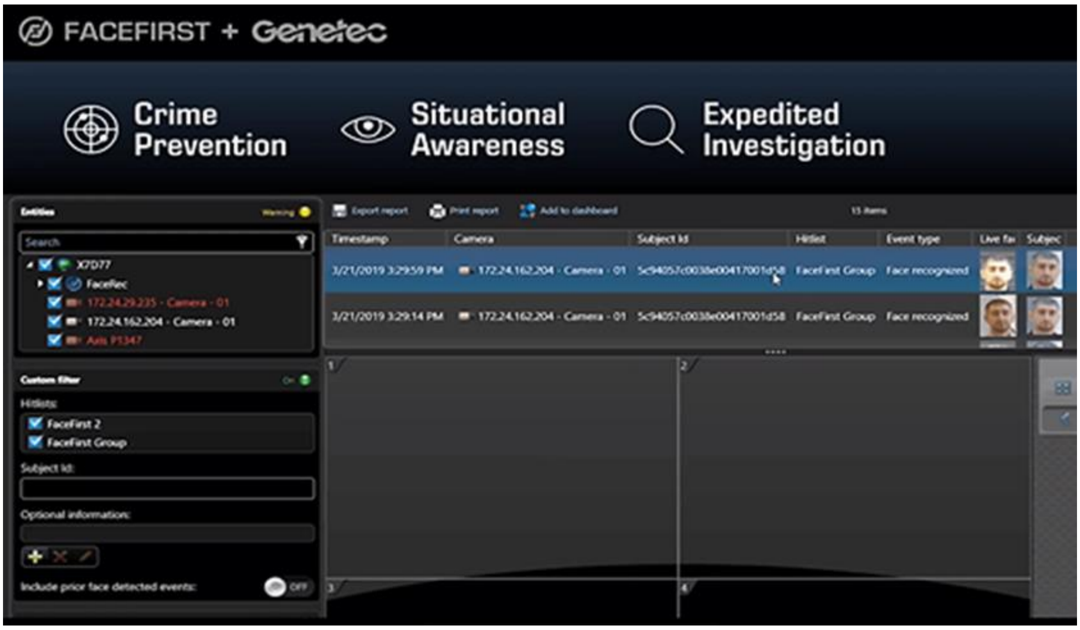


Figure 8 - Screenshot of FaceFirst Software

2.2.4. Kairos

Kairos API can recognize a human's face and do many different things like blink, smile, and move their head to make the interaction between the two easier. Another key point in facial recognition software is the ability of it to interact with a user's natural expressions. Currently, there are a couple of different technologies in place that are used to enable the user to tell when they are happy, sad, angry, or anything else. You will have the added benefit of being able to set your mood and have the computer constantly monitor how you feel so that you can better control your emotions [31].

Kairos has many advantages, there are also some disadvantages to consider like privacy concerns and accuracy, while Kairos API is known for its accuracy, facial recognition technology is not 100% accurate. The accuracy of the API can be affected by factors such as lighting, angle, and the quality of the image being analyzed.

Another disadvantage is cost for Kairos API with more advanced features available at a higher cost. For some developers, the cost of using the API may be a disadvantage. Also, integration is disadvantage while Kairos API is designed to be easy to integrate into applications, it may still require a significant amount of development work to incorporate the API into an application effectively. This could be a disadvantage for developers who do not have the necessary skills or resources to complete this work.

2.2.5. Face++

Face++ face detection API from Megvii detects and analyzes people using DL models. These models are trained on millions of images to ensure maximum face recognition accuracy with 99%. The main features of Face++ include facial recognition models, comparison, emotion recognition, and high-accuracy body recognition. It can determine age, race, gender and determine whether a person is smiling or not, by analyzing a facial image [32].

One disadvantage of using the Face++ API is that it may not always be accurate in identifying faces. The accuracy of facial recognition technology depends on a few factors, such as lighting, pose, facial expressions, and image quality. In some cases, the API may misidentify a person or fail to identify a face at all, which can lead to errors or inaccuracies in the data being analyzed.

Another disadvantage is that the Face++ API may raise privacy concerns. Facial recognition technology has the potential to be used for surveillance purposes, and there have been concerns about how this technology may be used to monitor individuals without their consent. Additionally, the use of facial recognition technology raises questions about data security and the potential for the misuse of personal information.

Finally, the Face++ API may be limited by its dependence on visual data. While facial recognition technology can be useful in certain contexts, it may not be the best option for tasks that require other forms of data, such as voice or biometric data. In these cases, other types of technology may be more effective.

2.2.6. BetaFace

BetaFace mainly focuses on image and video analysis for facial recognition and compare faces. It offers three kinds of services which are facial recognition software development kit (SDK), customer software development services, and hosted web services. Services can be simple face detection or complex facial recognition, identification, and verification. It uses biometric measurements for facial feature tracking on both images and videos. The tool can detect gender, age, facial expression, ethnicity. It can also track skin, hair, facial features, and hairstyle [32, 33].

BetaFace API has some disadvantages. Here are some potential disadvantages like accuracy in some cases, it may fail to recognize a face or may misidentify a person. This could be a significant problem if the system is being used for security or surveillance purposes.

Another disadvantages, privacy concerns If BetaFace API is being used to collect and store images of people's faces, there is a risk that this information could be misused or hacked. Also, cost for BetaFace API is not a free service, and the cost of using the API may be prohibitive for some individuals or organizations.

Finally, dependence on internet connection BetaFace API is a cloud-based service, it requires a reliable internet connection to work. If the internet connection is slow or unstable, the system may not function properly.

2.2.7. Kaggle_Model

The “kaggle_model.h5” is a pre-trained ML model for facial emotion recognition, this model that we used in our project that available on Kaggle website [34] is a CNN based model trained using the Keras deep learning library. The model has been trained on the FER2013 train set, which consists of 28,709 labeled images of faces with seven different emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral.

The “kaggle_model.h5” model has a test accuracy of 67.05% on the FER2013 test set which consists of 7,178 images belonging to seven classes. This accuracy can be affected by various factors, such as the quality of the input images and the specific use case [35].

The CNN model used in "kaggle_model.h5" is a deep neural network architecture, which uses a combination of several different functions such as sequential, 2D convolution layer (Conv2D), batch normalization, max pooling operation for 2D spatial data (maxpooling2D), rectified linear unit (ReLU), flatten, dropout, dense, and SoftMax function [35].

2.3. Comparison Between the Facial Emotion Detection APIs

This section explains the comparison of the previous facial emotion recognition APIs in terms of where they are best used and the accuracy of each, see Table 1 [32] that show the comparison between the facial emotion detection APIs depending on accuracy.

Table 1 - Comparison Between the Facial Emotion Detection APIs

API	% Accuracy	Best for
Kairos	62%	Finding faces and detecting features
Face++	99%	Extraction of facial information from images
BetaFace	81%	Facial recognition and transforming

2.4. Market Growth of Emotion Recognition Software

FER based technologies form an important part of the emotion recognition market. The facial emotion detection market is expanding tremendously. Recently, it has been estimated that the global advanced facial expression recognition market is expected to reach $56 billion by 2024. Emotion recognition takes a step further, and its use cases are nearly endless [36]. See Fig. 9 [36] that show the emotion detection and recognition market in North America, Europe, APAC, and RoW countries over the years, so the more years go by, the greater market demand for it.

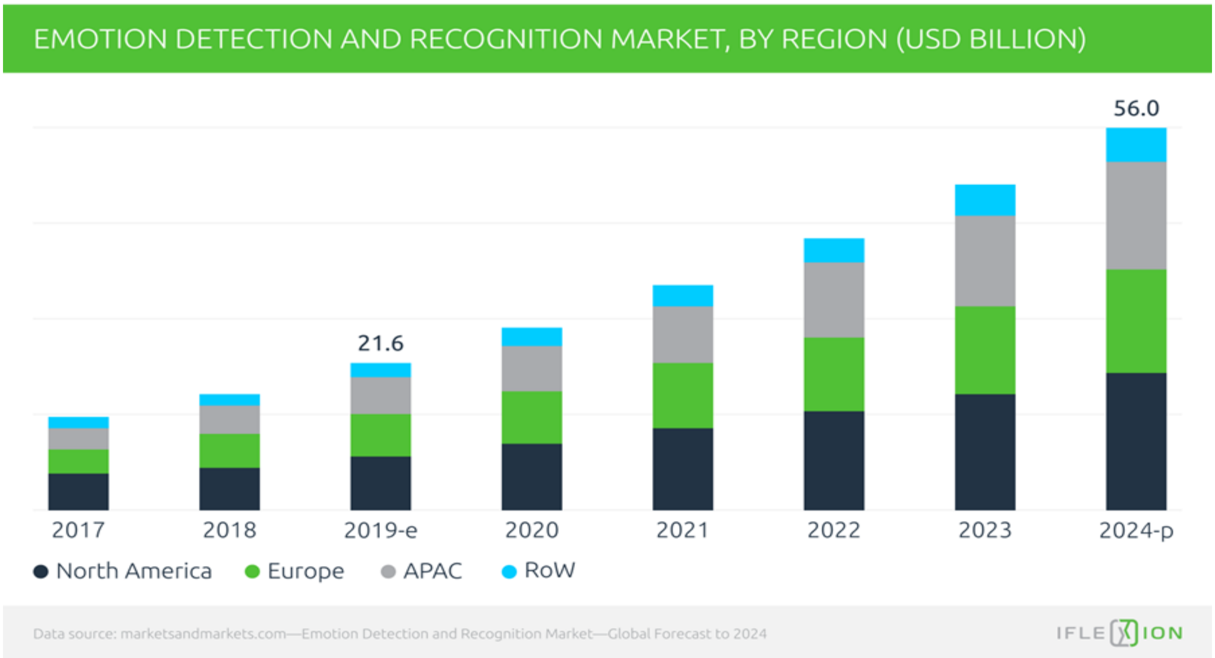


Figure 9 - Emotion Detection and Recognition Market

CHAPTER 3
TECHNOLOGY BACKGROUND

In this chapter, we will review how ML can be used in FER as well as some of the problems that may arise from using existing models. Then we move on to the definition of neural network (NN). Then we will review the use of DL in detecting facial expressions, learn about CNN and transfer learning (TL), and why it is effective in recognizing facial emotions. Then we will show hyperparameters and their effect. In addition, we will review the evaluation metric used in the ML algorithms to measure the performance of the model, operating systems, and programming languages used in this project.

3.1. Machine Learning in Facial Emotion Recognition

ML is the study of algorithms that can improve performance, data prediction, data learning and accuracy automatically through experience and by the use of data. ML is very useful in detecting and recognizing features accurately with the help of algorithms. ML has wide range of applications in medicine, computer vision, speech recognition and image recognition. One of the best applications of ML is facial emotion recognition. ML algorithms are helpful to develop emotion recognition models which can be trained to recognize human emotions accurately [37].

Facial expression recognition technology is everywhere these days, even if people are not that aware of it. Many people use facial expression recognition technology in a wide range of applications, like helping to early detection of depression and anxiety, computer interactive graphical games, also in social robots. It would be very beneficial if the machines are able to understand human emotions, emotion detection is necessary for machines to better serve their purpose. For example, the use of robots in areas like elderly care or as porters in hospitals demand a deep understanding of the environment. If a machine is able to obtain a sequence of facial images, then the use of ML techniques would help machines to be aware of their interlocutor’s mood. In this context, machine learning has the potential to become a key factor to build better interaction between humans and machines, while providing machines with a kind of self-awareness about its human peers, and how to improve its communication with humans [38].

Machine learning's growth has greatly improved the accuracy of facial expression identification tasks, recently many models have been built to overcome problems with emotion recognition from facial expressions. Nowadays different algorithms had applied for facial expression recognition models like multi-Layer perceptron (MLP), deep belief network (DBN), and recurrent neural network (RNN). But the most common type of ML algorithms that used for build facial expression analysis and emotion recognition model is CNN. CNN is a type of artificial neural network (ANN) that is well-suited for image classification tasks. CNN algorithm is used to analyze images for emotion detection and is in charge of feature extraction and classification of images based on facial features to assist in detecting human emotions [37, 39, 40, 41].

Feature extraction approaches based on neural networks have attracted a lot of attention recently, and this has helped to improve facial emotion detection performance. However, although the facial expression recognition system has a wide range of applications and a bright future, many technical problems still need to be solved. Face expression recognition is prone to complex issues such as difficulties extracting facial characteristics, face frame extraction failure, slow recognition speed or poor recognition accuracy [39].

Poor representation of people of different skin colors and ages in training data can lead to performance problems and biases in detecting facial expressions. These performance problems and biases are directly correlated with the problem of class imbalance in the datasets used to train these ML algorithms. Often, learned emotion models are trained on data sets that may not sufficiently represent a target population of interest. For example, many of these on-line services have focused on training and testing using a majority representation of adults and thus are tuned to the dynamics of mature faces. Similar challenges with biases in performance arise in other situations where datasets in these large-scale on-line services have a non-representative ratio of the desired class of interest. Although the derivations of these problems are different, these situations can bias the classification results and reduce accuracy, when designing ML algorithms for emotion recognition in intelligent systems [42].

3.2. Neural Network

Neural network also known as artificial neural network is a collection of interconnected layers that mimics the functioning of NNs in the human brain in certain aspects, is widely used in facial emotion recognition for classification of emotion features [37].

ANNs consists of an input layer, hidden layers, and an output layer. The input layer collects the data and passes it to the hidden layer where the data is processed to produce the results. Because NNs can adapt to changing input, they can produce the best possible result without requiring the output criteria to be redesigned [38]. In Fig. 10 [38], a typical NN topology is presented.

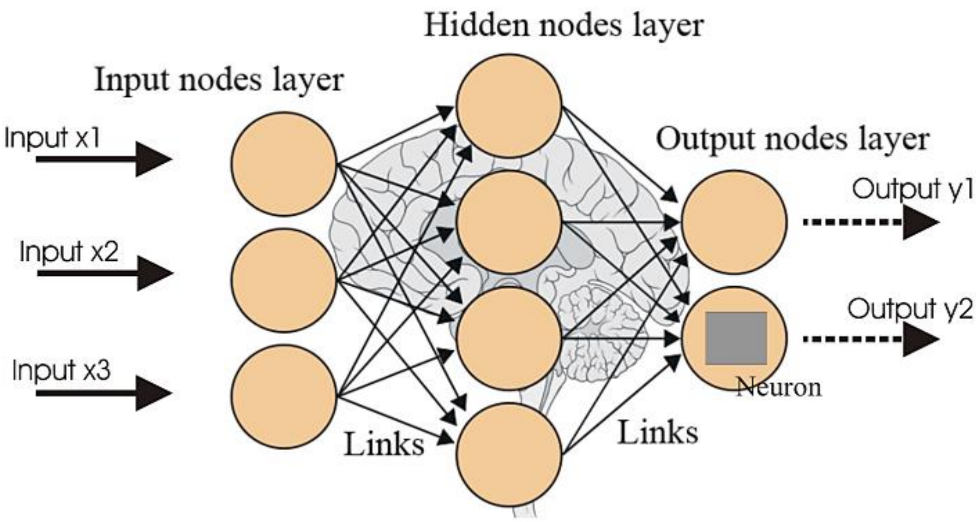


Figure 10 - Artificial Neural Network Topology

3.3. Deep Learning in Facial Expression Detection

DL is one of the branches of ML which completely makes use of artificial neural networks which helps to implement particular technical thoughts of the human brain into a machine. DL is a deep neural network which is defined as a network that is made up of many layers, allows computational models that are composed of multiple processing layers to learn and train machines to understand huge data and their representations with multiple levels of abstraction. DL uses data in various forms including image, video, text, speeches and conversations. DL algorithms make use of raw data to extract features, group objects and different patterns of meaningful data [37].

DL technology can be used to analyze facial expressions and recognize emotions, to help us extract more effective and flourishing expression characteristics, and to analyze and interpret various facial features and gestures of a person, such as mouth shape, eye movement and eyebrow position. Also used to detect human faces by recognizing human facial features from images or videos, which enhances the accuracy of facial expression identification and classification results [39, 43]. There are many types of deep neural networks such as CNN.

3.3.1. Convolutional Neural Network

CNN is a type of deep neural network which is mainly used for image classification and recognition tasks, including face emotion recognition. The three important layers of CNN as illustrated in Fig. 11 [37] below are the input layer, hidden layers and an output layer, the hidden layers consist of the convolutional layer, pooling layer, and fully connected layer. The figure shows how the input given to the model is processed by the CNN to produce evaluated outputs after the model is trained [37].

CNNs are a powerful tool for facial emotion recognition, since they are effective in feature extraction. CNNs can learn to extract facial features such as eye shape, lip curvature, and eyebrow position, which are important signals for emotion identification. CNNs are also relatively easy to implement. It also provides high accuracy and robustness to changes in facial expression, lighting, and orientation. This is important for FER because emotions can be expressed differently by different people and in different lighting conditions [37].

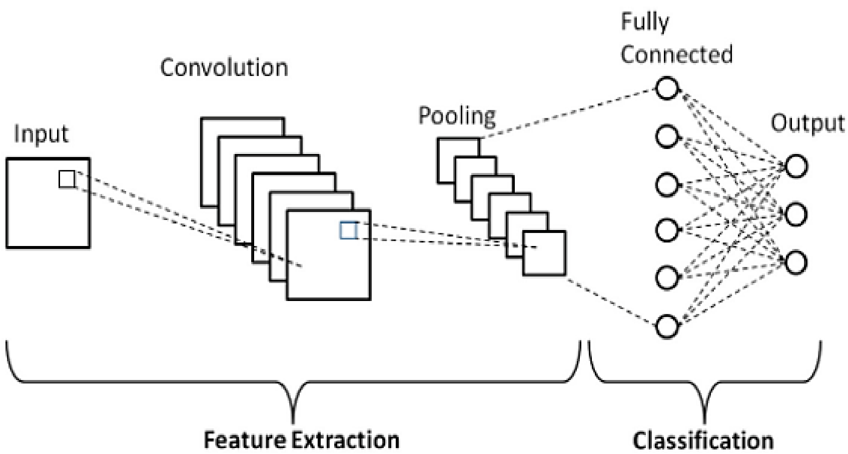


Figure 11 - Architecture of Basic Convolutional Neural Network

3.4. Transfer Learning

TL is a ML technique that involves leveraging the knowledge gained while training a ML model on one task, and applying it to a different but related task, rather than training a model from scratch for a new task. TL allows us to use a pre-trained model that is usually a deep neural network, trained on a large and diverse dataset [44].

TL can be particularly effective for face emotion recognition, as pre-trained models can be used to leverage knowledge gained from large datasets of facial images to improve the efficiency and accuracy of emotion recognition on smaller datasets. This is especially useful in cases where the dataset for emotion recognition is small, as transfer learning allows us to take advantage of the knowledge learned from larger datasets. In addition to reducing the time and resources required to train a model from scratch [45].

3.5. Hyperparameter Tuning

Hyperparameters are the parameters that are set before training a ML model and are not learned from the data during training. These parameters control the behavior of the algorithm and can significantly impact the performance of the model. Examples of hyperparameters include learning rate, batch size, and number of epochs. Choosing the right hyperparameters is critical for achieving good performance and generalization of the model, and hyperparameter tuning is an essential step in the ML pipeline [46].

3.5.1. Learning Rate

Learning rate is a key hyperparameter in ML algorithms that determines the step size taken during the optimization process. This hyperparameter defines how much freshly collected data will overrule previously accessible data. If the learning rate is too high, the model may overshoot the optimal solution and fail to converge. On the other side, if the learning rate is too low, the model may take a long time to converge or get stuck in a suboptimal solution [37].

3.5.2. Batch Size

Batch size is another important hyperparameter in ML that determines the number of samples used in each iteration of the training process. A larger batch size can lead to faster convergence and more stable gradients, but may require more memory and computational resources. A smaller batch size, on the other hand, can improve the generalization ability of the model and reduce overfitting, but may require more training time and produce more noisy gradients [37].

3.5.3. Number of Epochs

The number of epochs is key hyperparameter in DL models. It refers to the number of times the entire training dataset is processed during the training process. If the number of epochs is too low, the model may underfit the data and fail to capture the underlying patterns. On the other hand, if the number of epochs is too high, the model may overfit the data and memorize the training set, leading to poor generalization on new data [37].

3.5.4. Optimizer

The optimizer that we used in this project is adaptive moment estimation (Adam), which is a popular gradient descent optimization algorithm that is used in ML to update the parameters of a model to minimize the loss function. It is particularly useful in DL, where we have many parameters that need to be updated efficiently. It combines the advantages of two other optimization algorithms, adaptive gradient descent (Adagrad) and root mean square propagation (RMSprop). The algorithm uses adaptive learning rates for each parameter, based on the estimated first and second moments of the gradients, which allows it to converge faster and more reliably than other optimization algorithms [47].

Therefore, Adagrad adapts the learning rate for each parameter based on the historical gradient information. It uses a separate learning rate for each parameter, which allows it to give more weight to the parameters that have a larger gradient magnitude. Adagrad is well-suited for sparse data and has shown to work well in natural language processing tasks [47].

As RMSprop, it also adapts the learning rate for each parameter based on the historical gradient information. However, instead of storing all historical gradients as in Adagrad, RMSprop only keeps a moving average of the squared gradients, which reduces the oscillations in the gradient updates. RMSprop is known to work well in DL applications with non-stationary or noisy gradients [47].

In addition, Adam optimizer includes bias correction to compensate for the fact that the estimated moments may be biased towards zero at the beginning of training, and it also performs well in noisy and non-stationary environments [47].

3.5.5. Loss Function

The loss function that we used in this project is categorical cross-entropy, which is particularly used in classification tasks with multiple classes where the output of the model is a probability distribution over several classes. It measures the difference between the predicted probability distribution and the true distribution of the target class labels [37].

The formula for categorical cross-entropy involves taking the logarithm of the predicted probabilities and multiplying them by the true probabilities, then summing over all the classes. The resulting value indicates how well the model is predicting the correct class probabilities, with lower values indicating better performance [37].

3.6. Evaluation Metrics

Evaluation metrics are an important aspect of any analytical process. They are used to quantify the effectiveness of a particular model or algorithm, providing insights into how well it performs in solving a specific problem. Common evaluation metrics include confusion matrix, accuracy, precision, recall, F1 score, and loss [48].

3.6.1. Confusion Matrix

A confusion matrix is a useful tool for evaluating the performance of a ML model, particularly in classification problems. Each row of the matrix contains real class occurrences, whereas each column represents expected class instances. The diagonal elements of the matrix represent the number of correct predictions, while the off-diagonal elements represent the number of incorrect predictions. The confusion matrix can be used to calculate accuracy, precision, recall, and F1 score [48].

The matrix consists of four entries: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The TP entry represents the number of samples that were correctly classified as positive, while the FP entry represents the number of samples that were incorrectly classified as positive. The TN entry represents the number of samples that were correctly classified as negative, while the FN entry represents the number of samples that were incorrectly classified as negative [48].

3.6.2. Accuracy

Accuracy is perhaps the most basic evaluation metric, measuring the percentage of correct predictions made by the model. It is defined as the ratio of the true predicted values to total predicted values. The higher the accuracy, the better is the performance of the model [48].

3.6.3. Precision

Precision is a commonly used evaluation metric in ML algorithms, particularly in classification problems, which measures the proportion of true positive predictions out of the total number of positive predictions made by the model. It is a measure of the model's ability to correctly identify positive samples, and it is defined as the ratio of TP to the sum of TP and FP [48].

3.6.4. Recall

Recall is another evaluation metric that particularly used in classification problems, which measures the proportion of true positive predictions out of the total number of actual positive samples in the dataset. It is a measure of the model's ability to correctly identify all positive samples, and it is defined as the ratio of TP to the sum of TP and FN [48].

3.6.5. F1 Score

The F1 score is an evaluation metric that used particularly in classification problems, which combines both precision and recall into a single score. It is defined as the harmonic mean of precision and recall [48].

3.6.6. Loss

The loss occurred in the training process, also called as training loss, it indicates the error on the training set of data. It is the total number of errors occurred for training set while training the model [37].

3.7. Operating Systems

When it comes to developing mobile applications, the choice between Android and iPhone operating system (iOS) depends on a variety of factors such as target audience, business requirements, and development resources. In our project, we adopted Android for several reasons, including a larger user base as Android has a larger market share than iOS, which means developing an app for Android can reach a larger audience because the platform is open source, which means that developers can access the source code and modify it to suit their needs. This makes it easy to customize the application and integrate it with other systems [49].

The most important reason for choosing android is that it has more flexibility when it comes to developing applications, allowing developers to create more complex and feature-rich applications as they can be downloaded on the Google Play Store which has more applications than the Apple App Store, which means that users are more likely to discover your app so Android app development is generally easier and requires less initial investment than iOS app development [50].

Although iOS is a popular and powerful mobile app development platform, there are also some drawbacks that made us avoid using this platform. It is a closed ecosystem, which means that app development is subject to stricter guidelines and regulations. This can limit the flexibility of app development and can make it more difficult to get apps approved for the App Store, and higher development costs as iOS app development is more expensive than Android app development due to the need for specialized development tools, hardware, and programming languages. iOS devices are limited to iPhones and iPads, which may limit the potential audience for your app as they have limited hardware options. iOS also offers limited customization options compared to Android, which may not be suitable for apps that require more flexibility or customization [51].

In general, Android and iOS are two of the most popular mobile OS in the world, and they both have their advantages and disadvantages. The choice between them depends on the specific needs and goals of the project, as each platform has its own strengths and weaknesses. Therefore, the android system is adopted in our project based on the above points [49]. The following Fig. 12 [49] shows the power of using the android system.

Figure 12 - The Power of Using Android

3.8. Programming Languages

3.8.1. Python

Python is a popular programming language used for a wide variety of projects, from web development to scientific computing to AI. There are many reasons why Python is a great choice for this project. It is easy to learn and use, so the Python language has a simple and intuitive syntax, which makes it easy to learn and use, even for beginners. It's also an active community of developers who contribute to a vast ecosystem of libraries and frameworks, making it easy to find solutions to common problems, and versatile it can be used for a variety of tasks, from data analysis and visualization to web development and ML [52].

It also has cross-platform compatibility through which it runs on multiple platforms, including Windows, Mac operating system (macOS), and Linux, which makes it easy to deploy applications on different systems, and it has scalability as it is a scalable language, which means that it can handle large and complex projects with ease. Therefore, Python is a versatile, easy-to-use language that is well-suited to a wide variety of projects, making it a popular choice among developers [52].

3.8.2. Java

Java is a popular programming language used in a wide variety of projects, including web development, mobile app development, enterprise software development, and more. There are several reasons why Java is a great choice for this project. It is a platform-independent language, which means it can run on any platform that has the Java virtual machine (JVM) installed, including Windows, Linux, and macOS. This makes it easy to deploy applications on different systems. It's also an object-oriented language, which means it helps build scalable and modular applications. It's also within a large, active community of developers who contribute to a vast ecosystem of libraries and frameworks, making it easy to find solutions to common problems [53].

Also, one of the most important features is security. Java contains a built-in security mechanism that protects the system from viruses and other malicious attacks, through which it can expand and develop, which means that it can handle large and complex projects easily. Therefore, Java has been widely used in the enterprise for decades, and as such, it has developed a wide range of tools, frameworks, and technologies to support enterprise-wide applications. Therefore, Java is a powerful and versatile language that is well suited to a wide range of projects, especially those that require scalability, security, and enterprise support [53].

CHAPTER 4
PROGRESS AND IMPLEMENTATION

This chapter outlines our progress so far from the moment we decided to create an easy-to-use Android app that detects emotions by taking a photo, selecting one from the gallery or in real-time. It works on processing the image, then shows the user's emotion and best advice for him, also suggesting a video for the user to watch to help him enhance his mood and overall well-being.

We will go over what we did to get ready to start the implementation process by studying, reading, and preparing the datasets. Then we will review our experiments with preprocessing datasets, retrain and test the CNN model. Finally, we will end the chapter by explain how we developed our mobile app including the design of the app and how it works.

4.1. Preparation Process

After we have searched for an idea for our project and it was approved, we read and studied several papers so that we could write the second chapter of the documentation, which reviews related works and papers that discussed facial emotion recognition using ML and mentions several applications used to facial emotion recognition.

Then we started to review what we learned in the AI course, also we learned more about ML and DL, and we looked for good resources to learn Android application development. Then we searched for datasets to work on and studied them. Eventually, we determined that FER2013 dataset from Kaggle website [54] is the best choice, which contains 35,887 images of men's faces and women's faces who do not wear hijab, and seven categories.

But since this dataset does not contain images of Middle Eastern faces and also does not contain images of women wearing hijab, we decided to looked for another dataset to work on, and we found the Iranian emotional face database (IEFDB) from OSF website [55] that contains the images of Middle Eastern faces of men and women wearing hijab, this dataset contains 248 face images and seven categories.

4.2. Datasets

FER is a well-studied field, and there are several popular datasets available on the web that one can use to learn about facial expressions. Table 2 shows a comparison of the facial emotion recognition datasets we used in this project that include FER2013 dataset, CK+ dataset, JAFFE dataset, IEFDB dataset, and hybrid dataset depending on what includes, total number of images, number of classes, resolution, emotion classes, and source. Also, Table 3 provides the label and number of images for each emotion in the datasets [55, 56, 57, 58]. The image representation of the datasets is shown in Fig. 13 such that each row represents an individual dataset and each column represents a different emotion [11, 54, 59].

Table 2 - Comparison Between the Facial Emotion Recognition Datasets Used in This Project

Dataset	Includes	Total Number of Images	Number of Classes	Resolution	Emotion Classes	Source
FER2013	_	35,887	7	48x48	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	Kaggle
CK+	_	902	7	640x490 or 640x480	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	University of California, Berkeley
JAFFE	_	213	7	256x256	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	Kyushu University, Japan
IEFDB	_	248	7	5184x3456	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	Tehran University of Medical Sciences, Iran
Hybrid	CK+, JAFFE, and IEFDB Datasets	1363	7	48x48	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	We Created It

Table 3 - Comparison Between the Number of Images for Each Emotion in the Datasets

Label	Emotion	Number of Images for FER2013	Number of Images for CK+	Number of Images for JAFFE	Number of Images for IEFDB	Number of Images for Hybrid
0	Angry	4,953	45	30	39	114
1	Disgust	547	59	29	40	128
2	Fear	5,121	25	32	13	70
3	Happy	8,989	69	31	40	140
4	Sad	6,077	28	31	38	97
5	Surprise	4,002	83	30	38	151
6	Neutral	6,198	593	30	40	663

Figure 13 - Sample Images for Each Emotion Class in the FER2013, CK+, JAFFE, and IEFDB Datasets

4.3. Datasets Preprocessing

To prepare ourselves for the implementation process, we started studying codes we found on GitHub and Kaggle, along with researching a pre-trained facial expression recognition model using CNN, trained on the FER2013 dataset, we also looked at how integrate the model into our application.

Initially, we tried to understand how to preprocess the images of IEFDB to be applied to the ML model, to improve model performance and reduce the risk of overfitting. Therefore, we preprocessed the dataset images into the FER2013 dataset images format by resized the images to a size of 48x48 pixels, converted the color space of the images to grayscale, and normalized the pixel values in the images to a range between 0 and 1 by dividing each value by 255, then we split it into 80% for the training set and 20% for the test set, then we split again the train set into 80% for the training set and 20% for the validation set. The following Table 4 shows the IEFDB splitting of the train, test, and validation dataset, depending on the split value and the number of images.

Table 4 - The Split Value and Number of Images for Train, Test, and Valid Set of IEFDB

Dataset	Split Value	Number of Images
Train	60%	158
Test	20%	50
Validation	20%	40

Then to obtain a good accuracy score and to improve the performance and generalization ability of ML model, and reduce the impact of overfitting, also to increase diversity by covering a broader range of scenarios and conditions, we tried to increase the size of the training dataset by adding the CK+ dataset, and the JAFFE dataset. The CK+ dataset contains 902 face images of female and male between the ages of 18 and 50 with a variety of heritages, and have seven categories. As for the JAFFE dataset contains 213 images of Japanese female faces and seven categories, and we download them from GitHub website [60].

We started by preprocessing the images of these datasets into the FER2013 dataset images format by resizing them to 48x48 pixels, converted the color space of the images to grayscale, and normalized the pixel values in the images to a range between 0 and 1. Then we added them to the IEFDB, thus we have a hybrid dataset containing 1363 images from IEFDB, JAFFE, and CK+ datasets. We split it into 80% for the training set and 20% for the test set, then we split again the train set into 80% for the training set and 20% for the validation set. The following Table 5 shows the splitting of the hybrid dataset for the train, test, and validation dataset, depending on the split value and the number of images.

Table 5 - The Split Value and Number of Images for Train, Test, and Valid Set of Hybrid Dataset

Dataset	Split Value	Number of Images
Train	60%	872
Test	20%	273
Validation	20%	218

4.4. Retraining and Testing the CNN Model

Using the Python programming language, we retrain and test the ML model with different sizes of the training set and test set. The package used for split was imported from the Scikit-learn library. Using the TensorFlow library, we loaded a pre-trained kaggle_model which were mentioned in subsection 2.2.7. That we found on Kaggle website [34] and was trained on the FER2013 train set, then we started retraining this model on two different sizes of training sets. The accuracy of the model was calculated and analyzed for each training set. The model is then tested on the test set and the obtained results have been analyzed and discussed in section 5.1.

4.5. Mobile Application Development

We developed the facial emotion recognition mobile application named Emotional ID using Android Studio, chose Java programming languages to develop the application, and used extensible markup language (XML) code to design the UI.

4.5.1. App Design and How it Works

Our Android app is simple enough to use as it will consist of five screens. In the first screen, the user will see the name of the app and the logo as shown in Fig. 14(a), then it will move to the second screen that contains two buttons, the first to take the photo with the camera, and the other to detect the user's emotions in real-time as shown in Fig. 14(b).

When press the first button, a screen will appear with three buttons, one to capture an image, the other to choose an image from gallery, and the last to process the image as shown in Fig. 14(c). Once the image has been processed, you will proceed to the next screen, which displays the image captured, the user's emotion, and tips on how to act positively while experiencing that feeling. In addition, there is a button that, when clicked, will open a video for the user to watch to help him improve his emotional intelligence and motivate him to improve his mood.

When the second button is pressed, a camera screen will appear for the user, which will detect his emotions in real-time. At the top of the camera there will be a button to flip the camera and next to it there is a button to set the delay time, and based on the time that the user sets, a notification will be sent to the user containing the user's emotion and some tips, and when clicking on the notification, a video will open for the user to watch to give him a visual reference to understand how to manage his feeling.

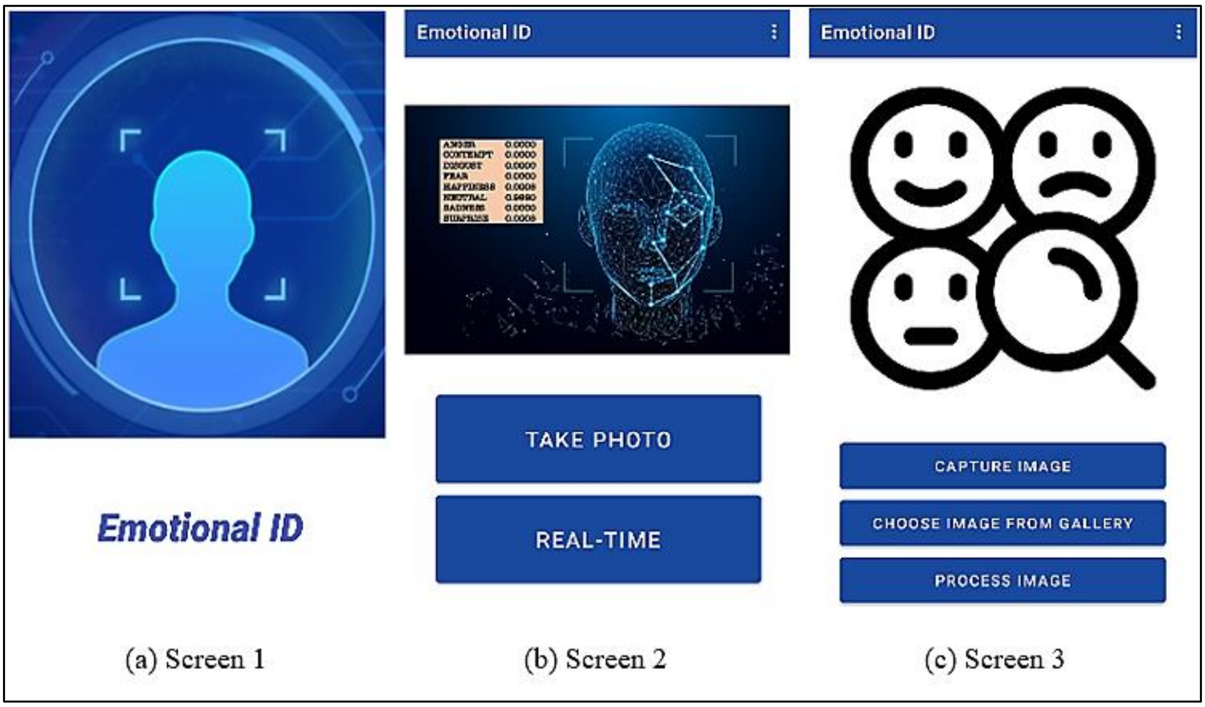


Figure 14 - The Design of the Application Screens

4.5.2. Convert Keras Model to TensorFlow Lite

To integrate our Keras model into our Android app, we used TensorFlow Lite (TFLite) which can be a powerful image recognition and analysis tool. By leveraging the power of ML on mobile devices, developers can create powerful and innovative applications that can help users solve complex problems and make better decisions [61].

TensorFlow Lite is a lightweight version of the popular TensorFlow library designed specifically for mobile and embedded devices. By converting a pre-trained Keras model from .h5 file extension into .tflite file extension, developers can run the model on mobile devices in real-time without requiring internet connectivity or a remote server [62, 63].

To convert a Keras model to TFLite format for use in an Android application, we used the TFLiteConverter module provided by TensorFlow. The TFLiteConverter module provides a simple and straightforward API for converting Keras models to the TFLite format. First, we install the TensorFlow Lite converter and import it into our Python code. Next, we define the model that needs to be converted to TFLite format, and then use the TFLiteConverter module to convert the Keras model to TFLite format [62, 63].

Once the model is in the TensorFlow Lite format, it can be added to the Android application as a model file. The application can then use the TensorFlow Lite interpreter to load the model and make predictions on new data. The TensorFlow Lite interpreter can then be used to run inference on the model and generate predictions in real-time [62].

4.5.3. Face Detection

We used the haarcascade_frontalface_alt.xml file available on GitHub website [64] in our application for face detection, which is a pre-trained model classifier trained on a large dataset of labeled face images for detecting frontal faces in images and videos. The model uses adaptive boosting (AdaBoost) algorithm in order to yield better results and accuracy. It is part of the open-source computer vision (OpenCV) library and ML library. This classifier can be used in various computer vision applications, such as security systems, facial recognition, augmented reality, and facial expression recognition. The file can be used in various programming languages that support OpenCV, including Python, and Java [65].

To use this file for face detection in our app, we followed several steps. First, we download the haarcascade_frontalface_alt.xml file from GitHub and include it in the Android project directory. Next, we initialize the cascade classifier object in the application code and load the haarcascade_frontalface_alt.xml file into it. Next, we read the image in which faces will be detected and converted it to grayscale. Finally, we used the DetectMultiScale method of the cascading classifier object to detect faces in grayscale images. The DetectMultiScale method returns the position and size of the detected object as a bounding box or a set of rectangles. These rectangles represent the regions in the image where the object is likely to be present [66].

Fig. 15 illustrates the process that the image goes through. First, the image is fed into the haar cascade model, which detects the face in the image, then fed into the CNN model, which predicts the emotion of the input image.

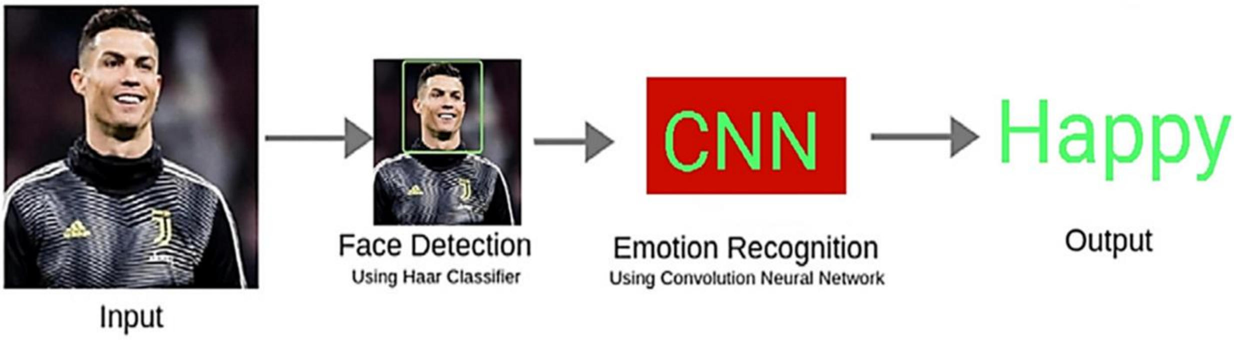


Figure 15 - The Steps an Image Goes Through to Detect Facial Emotions

CHAPTER 5
RESULTS AND DISCUSSION

This chapter presents the results that we have reached in this project in terms of results for the retrained model on the IEFDB dataset and the retrained model on the hybrid dataset. In addition to the results of the application, and it will include a graphs and tables that showing the accuracy results that we have reached, with a discussion of these results.

5.1. Model Results

5.1.1. The Results of the Model Retrained on the IEFDB Dataset

After we loaded the pre-trained kaggle_model, we selected the batch size 40 and epochs 400. To prevent overfitting and improve the generalization ability of the model during training, we used two callback functions: Early Stopping and ReduceLROnPlateau. These functions helped us determine the optimal stopping point, and adjust the learning rate when progress stagnated. For compiling we used Adam as the optimizer with a learning rate of 0.01, and the categorical cross-entropy as the loss function. The model hyperparameters are shown in Table 6.

Table 6 - Model Hyperparameters Retrained on the IEFDB Train Set

Model Hyperparameters	Values
Learning Rate	0.01
Number of Epochs	400
Batch Size	40
Optimizer	Adam
Loss Function	Categorical Cross-entropy

We first tested it on the test set of the IEFDB dataset and we achieved 28% test accuracy, also we tested it on the test set of the hybrid dataset and we achieved 16% test accuracy, then we retrained the model on the training set of the IEFDB dataset and evaluated the model, we achieved 92% accuracy on the IEFDB training set, 74% accuracy on the IEFDB test set, and 58% accuracy on the IEFDB valid set.

Also, after retraining the model, we retested it on the test set of the hybrid dataset and we achieved 33% test accuracy, in order to ensure the accuracy of the model's recognition of faces for all genders without exception, including images of women wearing the hijab. The briefly accuracy of the model on training, test, and validation sets show in Table 7.

Based on the results presented in the table, we note the improvement of the accuracy values of the model retrained on the IEFDB dataset, also we note the improvement in the test accuracy of the hybrid dataset that consisting of IEFDB, CK+, and JAFFE datasets after retraining the model compared to before.

Table 7 - Accuracies of the Model Retrained on the IEFDB Train Set

Dataset	IEFDB Dataset Accuracy	Hybrid Dataset Accuracy
Train	92%	_
Test Before Retraining the Model	28%	16%
Test After Retraining the Model	74%	33%
Validation	58%	_

Fig. 16 shows the accuracy and loss for the model that retrained on the training set of the IEFDB dataset the figure explains the increase in both the accuracy of training and validation as the model progressed through the training epochs, and the figure illustrates a decrease in loss for each of the training and validation, this indicates the model's ability to minimize prediction errors. The figure also highlights the effect of the Early Stopping callback function, which halted the training process at epoch 71.

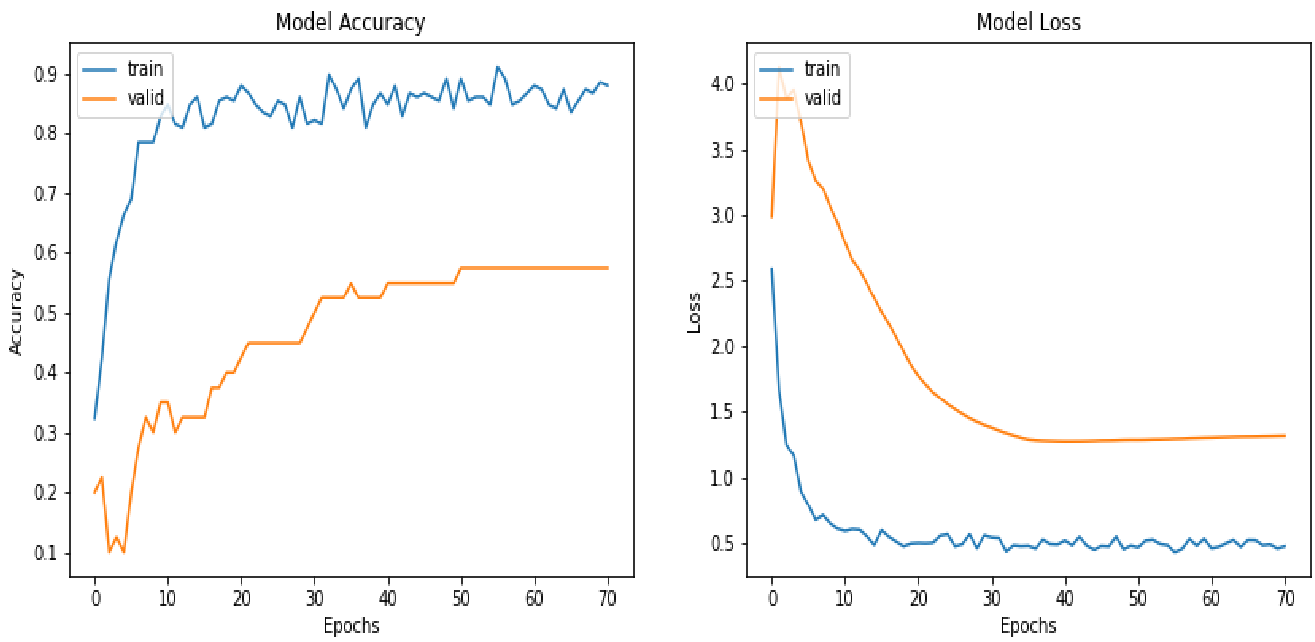


Figure 16 - Model Accuracy and Loss for Training and Validation Sets for IEFDB Dataset

Fig. 17 also displays the confusion matrix representing the number of TP, TN, FP, and FN, where the values on the diagonal represent the correct predictions made by the model. As we can see most of the false predictions are in surprise emotion where surprise is mostly predicted as neutral or sad. The second largest false predictions are in neutral emotion as neutral is sometimes predicted to be sad. We also note that the model did not predict any correct predictions of the emotion of fear, due to the small number of images of the fear emotion.

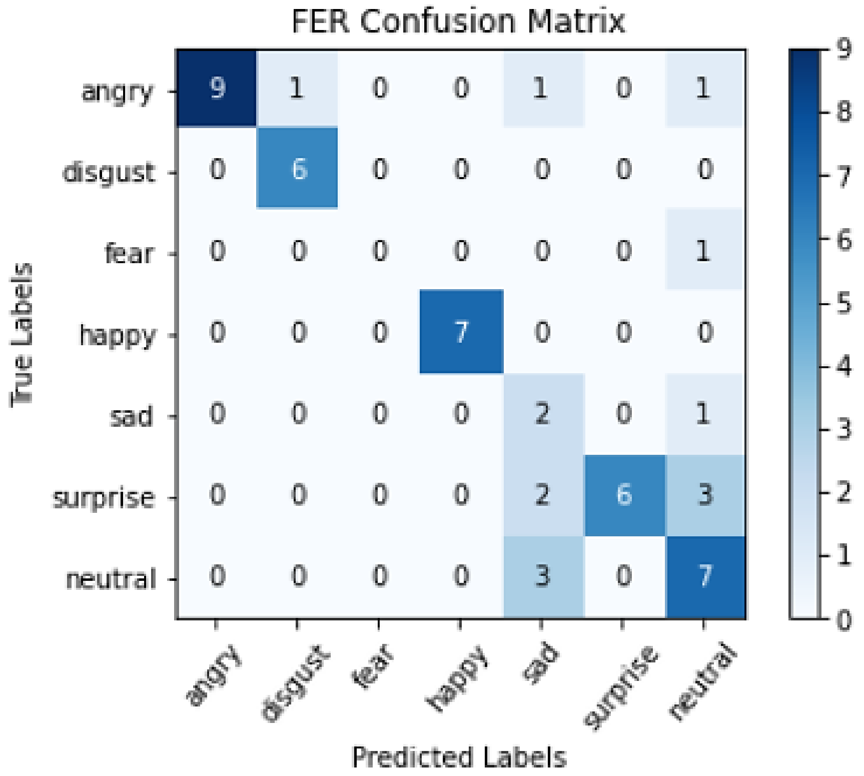


Figure 17 - Confusion Matrix of the Model for IEFDB Test Set

The classification accuracy and average results on the IEFDB test set for the model retrained on the IEFDB train set based on precision, recall, and F1 score are provided in Table 8. The table shows that emotions like angry, disgust, happy, and surprise have achieved high precision, recall, and F1 scores, indicating that the model accurately identified instances of these emotions. On the other hand, the model struggles with emotions like fear and sad, where the model's performance is considerably lower.

Table 8 - Classification Results of the Model on the IEFDB Test Set

Emotion	Precision	Recall	F1 Score
Angry	100%	75%	86%
Disgust	86%	100%	92%
Fear	0%	0%	0%
Happy	100%	100%	100%
Sad	25%	67%	36%
Surprise	100%	55%	71%
Neutral	54%	70%	61%
Average	66%	67%	64%

5.1.2. The Results of the Model Retrained on the Hybrid Dataset

After we loaded the pre-trained kaggle_model, we compiled the model using Adam's optimizer with a learning rate of 0.001, and the categorical cross-entropy as the loss function. We chose the number of epochs 300 and batch size 70. To prevent overfitting and improve model performance, we used three callback functions: Early Stopping, ReduceLROnPlateau, and Model Checkpoint. These functions helped us determine the optimal stopping point, adjust the learning rate when progress stagnated, and save the best model checkpoint, respectively. Table 9 shows the model hyperparameters.

Table 9 - Model Hyperparameters Retrained on the Hybrid Train Set

Model Hyperparameters	Values
Learning Rate	0.001
Number of Epochs	300
Batch Size	70
Optimizer	Adam
Loss Function	Categorical Cross-entropy

Then we tested it on the test set of the hybrid dataset and achieved 19% test accuracy, and tested it on the test set of the IEFDB so we achieved 34% test accuracy, these result before training the kaggle_model. Then we retrained the model on the training set of the hybrid dataset and evaluated the model, we achieved 88% test accuracy on the hybrid test set, 99% train accuracy on the hybrid train set, and 87% valid accuracy on the hybrid valid set.

Also, after retraining the model, we retested it on the test set of the IEFDB and we achieved 90% test accuracy, in order to ensure the accuracy of the model's recognition of Middle Eastern faces, and the faces of women who wear the hijab. Table 10 demonstrates our accuracy gains and the comparison of train accuracy, validation accuracy, and test accuracy before and after training kaggle_model on the hybrid dataset. The results, show that the test accuracy after retraining the model is higher than the test accuracy before retraining the model, and the model demonstrated its ability to recognize emotions in Middle Eastern faces and women wearing the hijab, as indicated by the high-test accuracy on the IEFDB test set after retraining the model.

Table 10 - Accuracies of the Model Retrained on the Hybrid Train Set

Dataset	IEFDB Dataset Accuracy	Hybrid Dataset Accuracy
Train	_	99%
Test Before Retraining the Model	34%	19%
Test After Retraining the Model	90%	88%
Validation	_	87%

Fig. 18 shows the accuracy and loss for the model that retrained on the training set of the hybrid dataset the figure explains the increase in both the accuracy of training and validation as the model progressed through the training epochs, and also shows a decrease in loss for each of the training and validation, also the Early Stopping made it stop training at epoch 118. Also, the confusion matrix of the model is shown in Fig. 19 that displays the number of TP, FP, TN, and FN for each class. As we can see most of the false predictions are in neutral emotion where neutral is mostly predicted as sad or disgust.

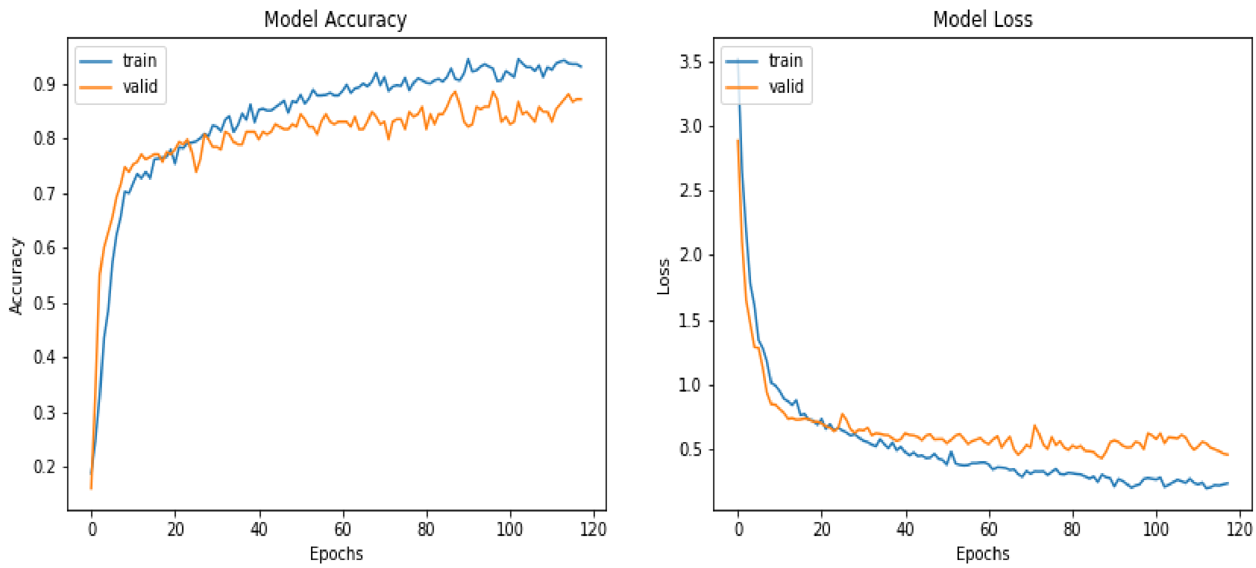


Figure 18 - Model Accuracy and Loss for Training and Validation Sets for the Hybrid Dataset

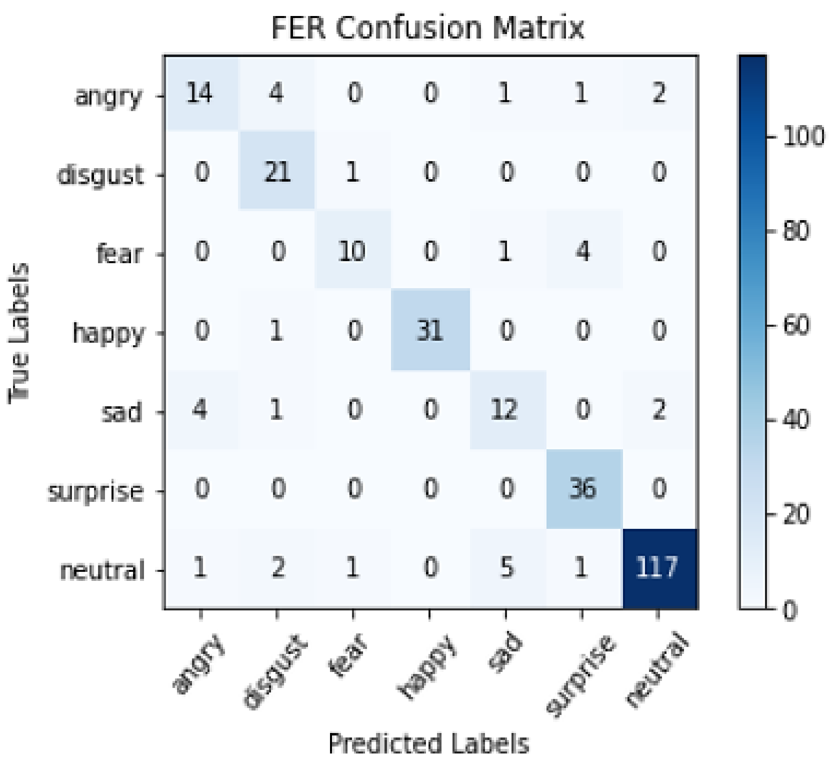


Figure 19 - Confusion Matrix of the Model for Hybrid Test Set

The classification accuracy and average results on the hybrid test set for the model retrained on the hybrid train set based on precision, recall, and F1 score are provided in Table 11. The table indicates that the model performs well in detecting happy and neutral emotions. However, it struggles a bit to detect emotions like angry and sad.

Table 11 - Classification Results of the Model on the Hybrid Test Set

Emotion	Precision	Recall	F1 Score
Angry	74%	64%	68%
Disgust	72%	95%	82%
Fear	83%	67%	74%
Happy	100%	97%	98%
Sad	63%	63%	63%
Surprise	86%	100%	92%
Neutral	97%	92%	94%
Average	82%	83%	82%

5.1.3. Discussion the Results

In this subsection, we will discuss the most important results that we reached based on the accuracy values that we obtained. On the basis of the above graphs and tables, which showed the improvement in model predictions after increasing the size of the dataset and adding two JAFFE and CK+ datasets with the IEFDB dataset, and we called it a hybrid dataset, then we divided it into a training and test sets, in order to evaluate the ML model performance on the independent test set that it has not seen before. We also divided the training set into training and validation sets for helps to ensure that the model is not overfitting to the training data and that it will perform well on new, unseen data.

The first model, retrained only on the IEFDB dataset with a learning rate of 0.01 and batch size of 40 and epochs 400, may converge faster and overfit more easily to the specific features and patterns of the IEFDB dataset. However, it may not generalize to new datasets or different images of the same emotion due to the limited training data and hyperparameters.

The second model, retrained on the hybrid dataset which is larger and more diverse dataset consisting of the IEFDB, CK+, and JAFFE datasets with a smaller learning rate of 0.001 and a larger batch size of 70 and epochs 300, may take longer to converge but may also lead to better generalization to new datasets. The larger dataset helps the model learn more robust and representative features and patterns of different emotions, while the smaller learning rate and larger batch size can help prevent overfitting and improve the model's stability.

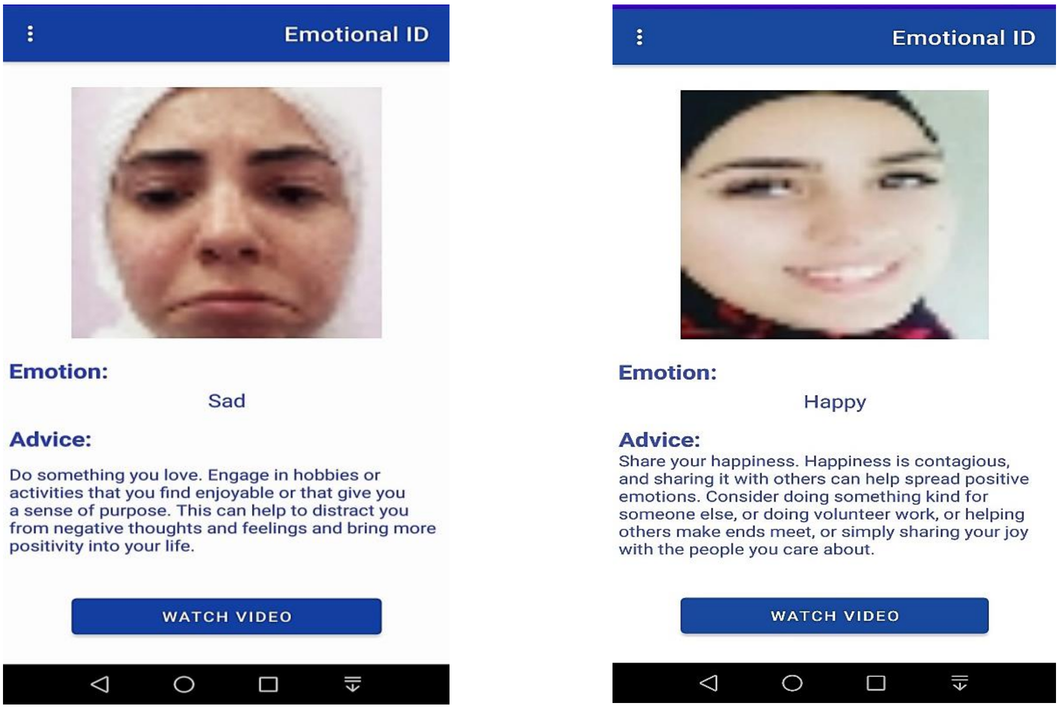
In summary, while both models may have different strengths and weaknesses depending on the specific datasets and hyperparameters, the second model retrained on a larger and more diverse dataset with a smaller learning rate and larger batch size have better generalization and performance on new datasets and images of different emotions.

Finally, after reviewing the accuracy results that came out with us, which for the first model is 74% test accuracy on the IEFDB test set, 33% test accuracy on the hybrid test set. For the second model is 90% test accuracy on the IEFDB test set, and a test accuracy of 88% on the hybrid test set. We therefore decided to use the second model that was retrained on the hybrid dataset, containing IEFDB, JAFFE, and CK+ datasets in our app because it showed higher accuracy, better performance, and less overfitting than the first model that retrained on the IEFDB dataset. The retrained model on the hybrid dataset can also recognize the feelings of most faces of different genders and races, including the faces of women wearing hijab and Middle Eastern faces that our project is focusing on because it was retrained on a large dataset that contained diversity in face images unlike the model retrained on IEFDB only.

Comparing the result that we reached with the previous works, we find that the highest accuracy that was reached was for the previous work CS230-FER that was mentioned in subsection 2.1.3. In which they trained the model on FER2013, CK+, and JAFFE datasets and achieved a test accuracy of 75.8%, while we used the pre-trained kaggle_model on the FER2013 dataset and its test accuracy is 67.05%, and we retrained it on CK+, IEFDB and JAFFE datasets, and we achieved 88% test accuracy, and thus we obtained a higher accuracy than the previous works, which is the highest to our knowledge.

5.2. Application Results

Our Emotional ID mobile application identifying the user's feelings can provide valuable feedback on the effectiveness and accuracy of the technology. In the first scenario, where the user can take a photo from the camera or choose photo from gallery, then the app will show the image, displaying the detected emotion below it, while showing advice for the user on how to act positively while experiencing that feeling. In addition, at the bottom of the screen there is a button that when the user clicks on it opens a video to help the user improve their emotional intelligence and mood. Fig. 20(a) shows the result of the sad face image that taken by camera, also Fig. 20(b) shows the result of the happy face image that chosen from gallery, and how the advice and the watch video button appear in the screen.

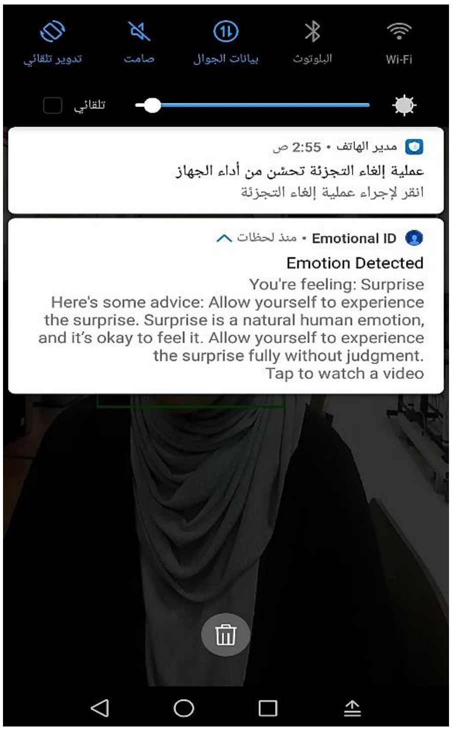


(a) Image That Take by Camera (b) Image That Choose from Gallery

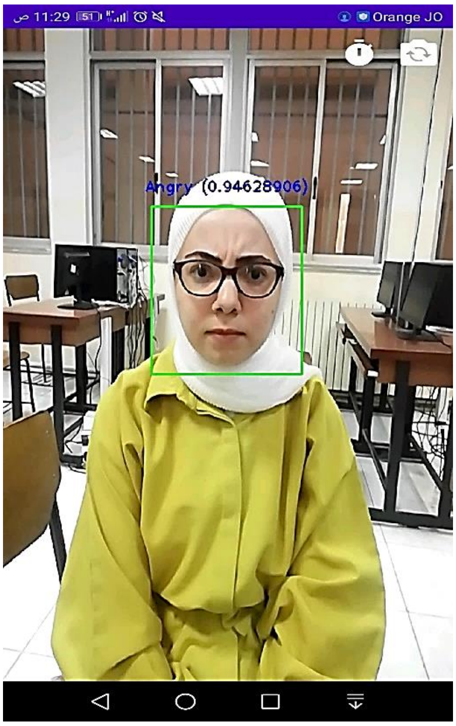
Figure 20 - The Results of Emotion Recognition in Two Cases of Take Photo

In the second scenario, where the app detects the user's emotions in real-time, a camera screen appears, with the flip camera button and delay time to send notification setting option located at the top, and based on the delay time set by the user, the app sends a notification containing the user's emotion and tips for managing it, which can be clicked to watch a helpful video. Fig. 21(a) and Fig. 21(c) show the result for the real-time emotion detection also Fig. 21(b) and Fig. 21(d) show how the notification appears on the screen.

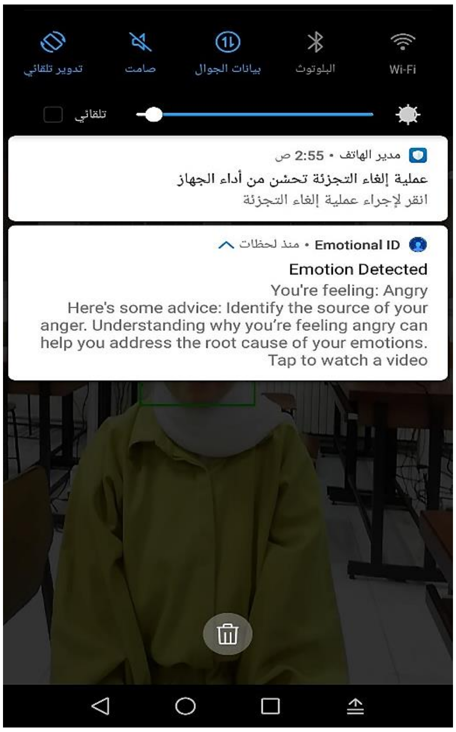
(a) Surprise Emotion in Real-Time



(b) Surprise Notification



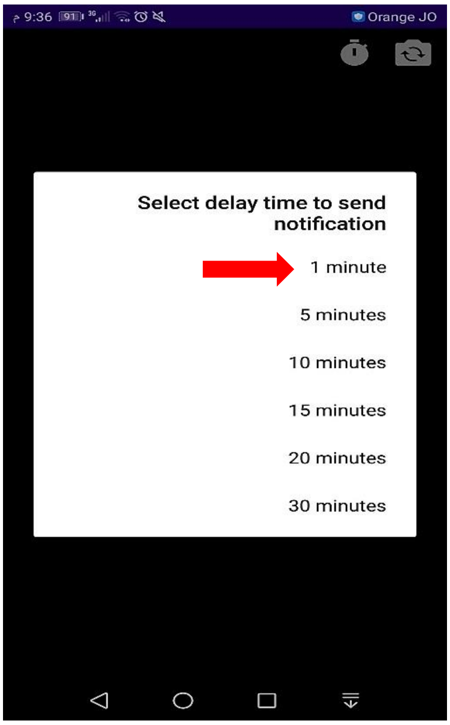
(c) Angry Emotion in Real-Time



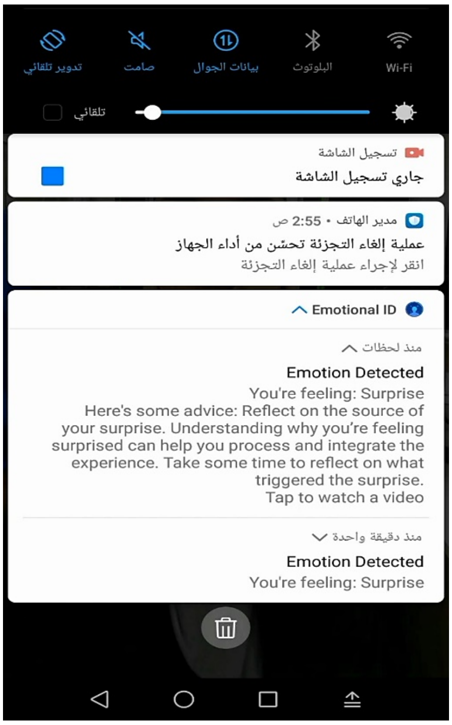
(d) Angry Notification

Figure 21 - The Results of Real-Time Emotion Recognition

Fig. 22(a) show the times to delay sending the notification so that the user can choose one of these times. Here we chose one minute, and Fig. 22(b) shows that the notification was sent after one minute.



(a) Choose a Delay of One Minute



(b) The Notification Appears After One Minute

Figure 22 - Select Delay Time and Send Notification

However, there are many issues our app may have in recognizing emotions as it is not 100% accurate. One major problem is that emotions are complex and subtle, and people express them differently, which can lead to difficulties in accurately interpreting and predicting emotions. In addition, the accuracy of the emotion detection feature in the application can also be affected by factors such as lighting, camera quality, image quality, and the user's facial expressions, which must be taken into account when evaluating the results of the emotion detection feature in the application, which can lead to incorrect results.

Our application is distinguished from previous applications in that it takes cultural differences into account so that there is no bias in the training data used. Our application can accurately recognize the feelings of East Asian faces and faces that have European features, and focus on Middle Eastern faces and faces that wear the hijab, which are faces that do not other applications recognize her feelings correctly. Where our application is characterized by providing advice and giving useful videos to improve the user's feeling, as in the case of detecting emotions in real-time, it sends a notification that contains the user's feeling and gives him the appropriate advice, and these features are absent in previous works.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

6.1. Conclusions

In conclusion, our project addressed the challenge of identifying facial expressions by developing a real-time facial emotion recognition application that is focused on Middle Eastern faces, especially for women wearing hijab, to overcome the bias and inaccuracy issues that current AI-based applications have. The project's importance lies in the fact that facial expressions are a key part of nonverbal communication, and the inability to perceive these expressions leads to a loss of communication effectiveness. Our project utilized the efficient CNN algorithm to analyze facial expressions and identify the emotional state of the person in real-time. This system has potential applications in various fields, including healthcare, gaming, and marketing. Additionally, it can also be useful in online education, where capturing students' affect can contribute to the effectiveness of teaching. By recognizing facial emotions, we can gain insight into the minds of people we interact with, even when they are not talking, and this information is important for many professionals.

We explored and retrained the model on a different dataset, and it turned out that the retrained model on a large dataset containing a variety of facial images is the most accurate, as the results of the project showed that the model that was retrained on the hybrid dataset achieved a test accuracy of 88% on the hybrid test set and 90% test accuracy on the IEFDB test set, which is the highest based on the research we have read. In addition, we demonstrated that FER models can be applied in the Arab world by developing an Android application with real-time recognition speeds that recognizes the emotions of different faces including Middle Eastern faces and the faces of women wearing hijab. The successful implementation of our project provides a foundation for future research in this field, and we hope that our work will be beneficial for both academic and industrial purposes.

6.2. Future Work

Our project has focused on developing a real-time emotion analysis application focusing on Middle Eastern faces where women wear the hijab to increase its usability in the Arab world. However, there are several ways in which our solution can be further improved to enhance its accuracy, usability, and effectiveness. One direction for future work is to expand the dataset used for training the DL model. Although we collected a large number of images for our dataset, it is still limited in terms of diversity.

Another direction for future work is to develop a system for continuous monitoring of emotions. Our current application provides real-time emotion analysis, but it is not linked to cloud services to store the user's emotions and keep track of their mood flow weekly or monthly and give them an arithmetic average of the emotions they felt over that time period. Furthermore, the application can be integrated with wearable devices, such as smartwatches or fitness trackers, to provide continuous monitoring of emotions throughout the day, such as monitoring the emotions of students during an online class or detecting the emotional state of patients in a hospital.

Finally, integration of acoustic information since emotions can be conveyed through voice, so we can improve the accuracy of our emotion recognition system by integrating vocal information with facial expression recognition. This can be done by using verbal emotion recognition techniques that can help provide additional information to the model. Body language can also be used to convey feelings and combine them with facial expressions. Thus, the application can be developed to make it discover the user's feeling from his voice or from his body language. We believe that implementing these suggestions will make a significant contribution to the field of facial emotion recognition, especially in the Middle Eastern region.

REFERENCES

[1] E. Pranav, S. Kamal, C. S. Chandran, and M. H. Supriya, "Facial emotion recognition using deep convolutional neural network," In 2020 6th International conference on advanced computing and communication Systems (ICACCS), pp. 317-320, IEEE, 2020.

[2] EDPS, K. Vemou, and A. Horvath, "Facial Emotion Recognition," 2021. [Online]. Available: https://edps.europa.eu/data-protection/our-work/publications/techdispatch/techdispatch-12021-facial-emotion-recognition_en. [Accessed 11 November 2022].

[3] I. K. Choi, H. E. Ahn, and J. Yoo, "Facial expression classification using deep convolutional neural network," Journal of Electrical Engineering and Technology, 13(1), pp. 485-492, 2018.

[4] M. Shamim Hossain, and G. Muhammad, "An emotion recognition system for mobile applications," IEEE Access 5, pp. 2281-2287, 2017.

[5] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," In 2005 IEEE international conference on multimedia and Expo, pp. 5-pp, IEEE, 2005.

[6] Communication Theory, S. Zhang, and colleagues, "Importance of Facial Expressions in Communication," [Online]. Available: https://www.communicationtheory.org/importance-of-facial-expressions-in-communication. [Accessed 11 November 2022].

[7] F. Mo, J. Gu, K. Zhao, and X. Fu, "Confusion effects of facial expression recognition in patients with major depressive disorder and healthy controls," Frontiers in psychology, 12, 2021.

[8] Apriorit, A. Beliba, "Challenges of Emotion Recognition in Images and Video," [Online]. Available: https://www.apriorit.com/dev-blog/642-ai-emotion-recognition. [Accessed 11 November 2022].

[9] O. Ekundayo, and S. Viriri, "Facial expression recognition: A review of trends and techniques," IEEE Access, 9, pp. 136944-136973, 2021.

[10] S. Deshmukh, and S. Yadav, "Facial Expression Recognition with Deep Learning," 2021.

[11] F. Heydari, S. Sheybani, and A. Yoonessi, "Iranian Emotional Face Database: Acquisition and Validation of a Stimulus Set of Basic Facial Expressions," Behavior Research Methods, 2022.

[12] Indiaai, P. Saxena, “Real-time emotion recognition: Potential use cases and challenges,” 8 Jun 2021. [Online]. Available: https://indiaai.gov.in/article/real-time-emotion-recognition-potential-use-cases-and-challenges. [Accessed 13 November 2022].

[13] A. Pise, M. Alqahtani, P. Verma, K. Purushothama, D. Karras, S. Prathibha, and A. Halifa, "Methods for Facial Expression Recognition with Applications in Challenging Situations," Computational Intelligence and Neuroscience, 2022.

[14] Humintell, D. Matsumoto, “Benefits of Reading Facial Expressions of Emotion,” 6 April 2021. [Online]. Available: https://www.humintell.com/2021/04/benefits-of-reading-facial-expressions-of-emotion/. [Accessed 13 November 2022].

[15] Researchomatic, “Impact of Facial Expressions,” [Online]. Available: https://www.researchomatic.com/Impact-Of-Facial-Expressions-90827.html. [Accessed 18 November 2022].

[16] Itv, A. Jha, "How facial recognition could be key to maintaining independence of elderly," 17 Feb 2017. [Online]. Available: https://www.itv.com/news/2017-02-17/technology-could-be-key-to-maintaining-independence-of-elderly. [Accessed 14 November 2022].

[17] A. Bandyopadhyay, S. Sarkar, A. Mukherjee, and S. Basu, "Identifying emotional Facial expressions in Practice: A Study on Medical Students," Indian Journal of Psychological Medicine, 43(1), pp. 51-57, 2021.

[18] Mayankbimbra.medium, “Real Time Facial Expressions/Emotions Recognition on a Web Interface using Python,” 7 Jul 2020. [Online]. Available: https://mayankbimbra.medium.com/real-time-facial-expressions-emotions-recognition-on-a-web-interface-using-python-b42f58a25780. [Accessed 14 November 2022].

[19] Softwebsolutions, A. Modawal, “Know the benefits of facial recognition and emotion detection tools,” 13 May 2022. [Online]. Available: https://www.softwebsolutions.com/resources/benefits-of-facial-recognition.html. [Accessed 18 November 2022].

[20] Venturebeat, "Chinese school installs facial recognition cameras to monitor students," 17 May 2018. [Online]. Available: https://venturebeat.com/ai/chinese-school-installs-facial-recognition-cameras-to-monitor-students/. [Accessed 19 November 2022].

[21] Cameralyze, “Facial Emotion Recognition,” [Online]. Available: https://www.cameralyze.co/facial-emotion-recognition-with-artificial-intelligence. [Accessed 4 January 2023].

[22] Github, “Face Analyzer,” [Online]. Available: https://github.com/ishaanjav/Face_Analyzer. [Accessed 4 January 2023].

[23] Apkgk, “Face Analyzer,” [Online]. Available: https://apkgk.com/app.anany.faceanalyzer. [Accessed 4 January 2023].

[24] A. Khanzada, C. Bai, and F. Celepcikay, "Facial expression recognition with deep learning," arXiv preprint arXiv:2004.11823, 2020.

[25] Apkcombo, “Emotimeter - Emotion detector,” [Online]. Available: https://apkcombo.com/emotimeter-emotion-detector/com.reaimagine.josem.emotimeter_facialemotionrecognizer/. [Accessed 4 January 2023].

[26] Nordicapis, “20+ Emotion Recognition APIs That Will Leave You Impressed, and Concerned,” 31 Dec 2015. [Online]. Available: https://nordicapis.com/20-emotion-recognition-apis-that-will-leave-you-impressed-and-concerned/. [Accessed 5 January 2023].

[27] Towardsdatascience, “What is the Best Facial Recognition Software to Use in 2022?,” 11 Mar 2021. [Online]. Available: https://towardsdatascience.com/what-is-the-best-facial-recognition-software-to-use-in-2021-10f0fac51409. [Accessed 5 January 2023].

[28] Fixthephoto, “6 best facial recognition software in 2023,” [Online]. Available: https://fixthephoto.com/best-facial-recognition-software.html. [Accessed 5 January 2023].

[29] GitHub, “CompreFace,” [Online]. Available: https://github.com/exadel-inc/CompreFace. [Accessed 6 January 2023].

[30] Androidexample365, “Face Recognition using FaceNet and Firebase MLKit on Android,” 24 Sep 2020. [Online]. Available: https://androidexample365.com/face-recognition-using-facenet-and-firebase-mlkit-on-android/. [Accessed 6 January 2023].

[31] Kairos, “FAQs,” [Online]. Available: https://www.kairos.com/faq. [Accessed 12 January 2023].

[32] Recfaces, “15 of the Best Face Recognition APIs in 2021: RecFaces’ Overview,” 19 Feb 2021. [Online]. Available: https://recfaces.com/articles/face-recognition-apis#5. [Accessed 12 January 2023].

[33] Spiceworks, “Top 11 Facial Recognition Software in 2021,” 2 Sep 2021. [Online]. Available: https://www.spiceworks.com/it-security/identity-access-management/articles/facial-recognition-software/amp/. [Accessed 12 January 2023].

[34] Kaggle, [Online]. Available: https://www.kaggle.com/code/aroop123/face-emotion-recoginition/data. [Accessed 27 February 2023].

[35] Kaggle, [Online]. Available: https://www.kaggle.com/code/aroop123/face-emotion-recoginition/notebook. [Accessed 27 February 2023].

[36] Iflexion, “The Future of Emotion Recognition in Machine Learning,” 5 July 2022. [Online]. Available: https://www.iflexion.com/blog/emotion-recognition-software. [Accessed 12 January 2023].

[37] L. Bejjagam, and R. Chakradhara, "Facial Emotion Recognition using Convolutional Neural Network with Multiclass Classification and Bayesian Optimization for Hyper Parameter Tuning," 2022.

[38] D. Spiers, "Facial emotion detection using deep learning," 2016.

[39] M. Bie, H. Xu, Y. Gao, and X. Che, "Facial Expression Recognition from a Single Face Image Based on Deep Learning and Broad Learning," Wireless Communications and Mobile Computing, 2022.

[40] Pxl-vision, “Machine learning and face recognition,” [Online]. Available: https://www.pxl-vision.com/en/blog/machine-learning-and-how-it-applies-to-facial-recognition-technology. [Accessed 25 December 2022].

[41] M. Pourebadi, and M. Pourebadi, "MLP neural network based approach for facial expression analysis," Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016.

[42] A. Howard, C. Zhang, and E. Horvitz, "Addressing bias in machine learning algorithms: A pilot study on emotion recognition for intelligent systems," IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), IEEE, 2017.

[43] Width.ai, “State of the Art for Facial Expression Recognition from Image Data | Techniques, Architectures and Marketing Use Case,” Available: https://www.width.ai/post/facial-expression-recognition. [Accessed 28 December 2022].

[44] Seldon, “Transfer Learning for Machine Learning,” [Online]. Available: https://www.seldon.io/transfer-learning. [Accessed 28 March 2023].

[45] Builtin, "What Is Transfer Learning? Exploring the Popular Deep Learning Approach," [Online]. Available: https://builtin.com/data-science/transfer-learning. [Accessed 28 March 2023].

[46] Towardsdatascience, "Parameters and Hyperparameters in Machine Learning and Deep Learning," [Online]. Available: https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac. [Accessed 29 March 2023].

[47] Towardsdatascience, "Adam — latest trends in deep learning optimization," [Online]. Available: https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c. [Accessed 29 March 2023].

[48] Analyticsvidhya, "Metrics to Evaluate your Classification Model to take the right decisions," [Online]. Available: https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/. [Accessed 30 March 2023].

[49] Rishabhsoft, "Top Benefits of Developing Android App to SkyRocket Your Business Ideas," [Online]. Available: https://www.rishabhsoft.com/blog/top-advantages-of-developing-android-app-for-your-business. [Accessed 30 March 2023].

[50] Scaler, "iOS Operating System," [Online]. Available: https://www.scaler.com/topics/ios-operating-system/. [Accessed 30 March 2023].

[51] Digitalaptech, "Advantages and Disadvantages of iOS," [Online]. Available: https://www.digitalaptech.com/advantages-and-disadvantages-of-ios/. [Accessed 30 March 2023].

[52] Techvidvan, "Python Advantages and Disadvantages – Step in the right direction," [Online]. Available: https://techvidvan.com/tutorials/python-advantages-and-disadvantages/. [Accessed 30 March 2023].

[53] Ibm, "Advantages of Java," [Online]. Available: https://www.ibm.com/docs/en/aix/7.1?topic=monitoring-advantages-java. [Accessed 30 March 2023].

[54] Kaggle, [Online]. Available: https://www.kaggle.com/datasets/msambare/fer2013. [Accessed 7 February 2023].

[55] OSF, "Iranian Emotional Face Database," [Online]. Available: https://osf.io/a6e2u/. [Accessed 30 January 2023].

[56] Kaggle, [Online]. Available: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data. [Accessed 30 January 2023].

[57] Paperswithcode, “CK+ (Extended Cohn-Kanade dataset),” [Online]. Available: https://paperswithcode.com/dataset/ck. [Accessed 30 January 2023].

[58] Zenodo, “The Japanese Female Facial Expression (JAFFE) Dataset,” [Online]. Available: https://zenodo.org/record/3451524#.Y9eyBHZBxPY. [Accessed 30 January 2023].

[59] I. Dagher, E. Dahdah, and M. Al Shakik, "Facial expression recognition using three-stage support vector machines," Visual Computing for Industry, Biomedicine, and Art 2(1), 1-9, 2019.

[60] GitHub, [Online]. Available: https://github.com/spenceryee/CS229. [Accessed 27 February 2023].

[61] Seeedstudio, “Getting Started with TensorFlow Lite on reTerminal,” [Online]. Available: https://wiki.seeedstudio.com/reTerminal_ML_TFLite/. [Accessed 28 March 2023].

[62] Studymachinelearning, “Introduction to TensorFlow Lite,” [Online]. Available: https://studymachinelearning.com/introduction-to-tensorflow-lite/. [Accessed 28 March 2023].

[63] Tensorflow, “Convert TensorFlow models,” [Online]. Available: https://www.tensorflow.org/lite/models/convert/convert_models. [Accessed 28 March 2023].

[64] GitHub, [Online]. Available: https://github.com/opencv/opencv/tree/master/data/haarcascades. [Accessed 28 February 2023].

[65] Ersanpreet, S. Singh, “Face and eyes detection with Viola Jones along with python code,” 10 March 2020. [Online]. Available: https://ersanpreet.wordpress.com/tag/haarcascade_frontalface_default-xml/. [Accessed 28 February 2023].

[66] Towardsdatascience, G. Behera, “Face Detection with Haar Cascade,” 24 Dec 2020. [Online]. Available: https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08. [Accessed 28 February 2023].

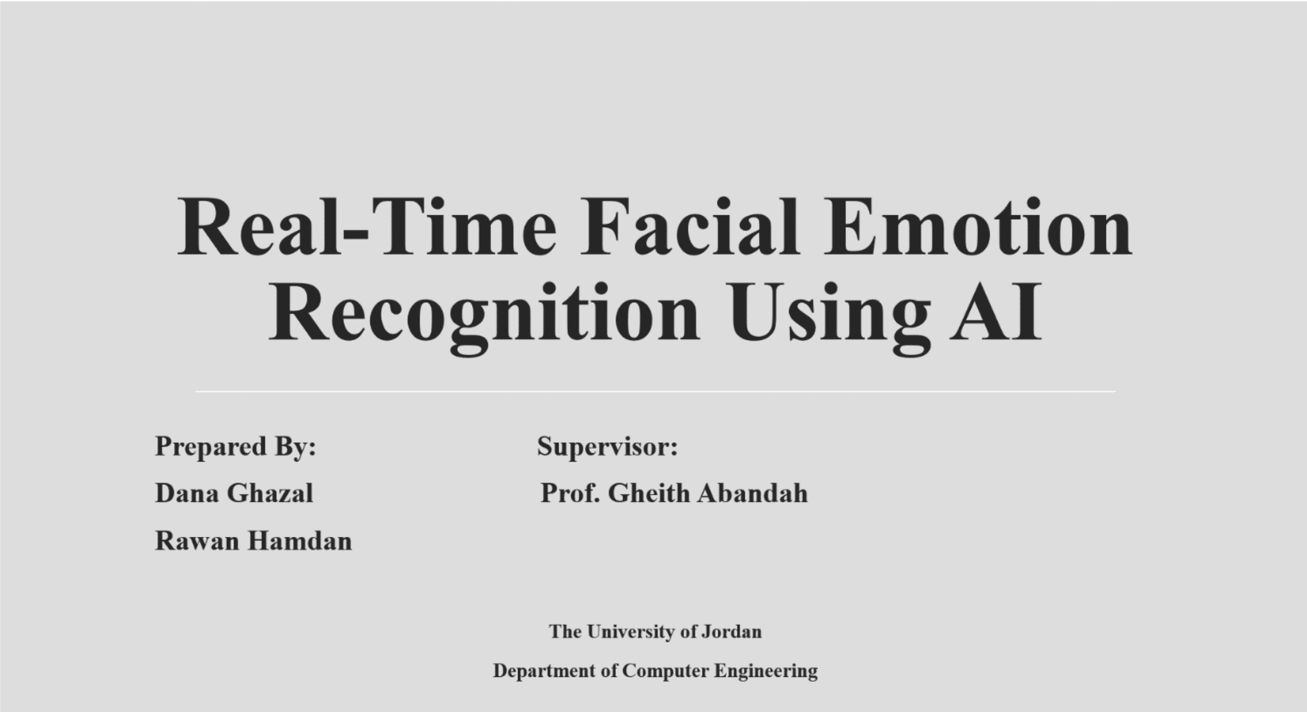
APPENDICES

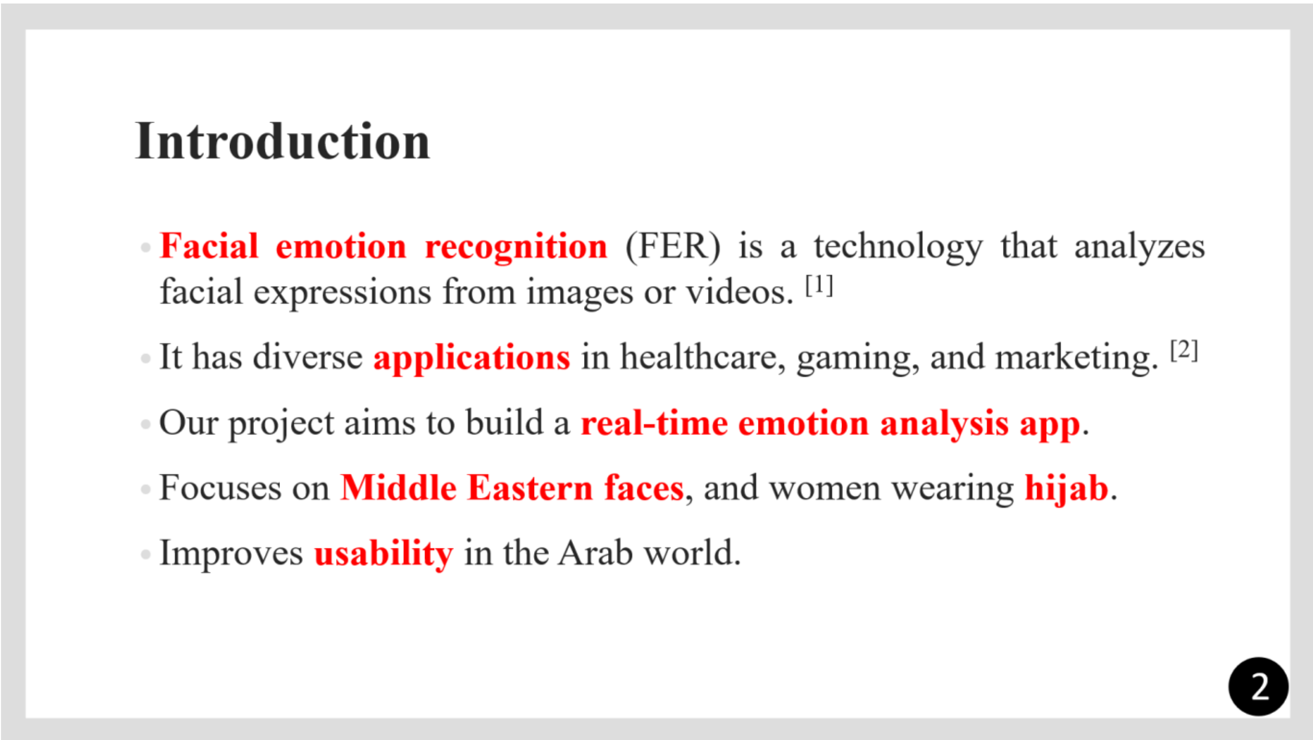
- **Project Time Chart**

TASK	START	END	NAME
Research			
Research for datasets	9/10/22	20/10/22	Both
Research for papers	21/10/22	21/11/22	Both
Research for impact the face emotions	21/10/22	21/11/22	Dana
Research for related work that use FER	22/11/22	22/12/22	Rawan
Research for applications and services	22/11/22	22/12/22	Both
Learning technologies			
Learning Python language	21/10/22	21/11/22	Both
Learning JAVA language	21/10/22	21/11/22	Both
Learning knowledge for AI	22/11/22	22/12/22	Both
Learning Android	23/12/22	23/2/23	Both
Implementation			
Preview the available datasets and models	23/12/22	23/1/23	Both
Implement the AI engine	24/1/23	1/3/23	Both
Extend the IEFDB with CK+ and JAFFE datasets	25/1/23	27/1/23	Rawan
Preprocessing the datasets	28/1/23	31/1/23	Dana
Retraining the model	1/2/23	15/3/23	Both
Testing the accuracy and the output of the model	10/2/23	15/3/23	Both

Building mobile application			
Application planning	16/3/23	21/2/23	Both
Building the user interface design	22/3/23	2/4/23	Rawan
Application development	3/4/23	30/4/23	Both
Interface the mobile app with the AI engine	15/4/23	30/4/23	Dana
Test the mobile application			
Testing the application using the emulator	15/4/23	15/5/23	Both
Testing the application using the real device	15/4/23	15/5/23	Both
The end information and updated			
Writing project documentation report	26/2/23	17/5/23	Both
Prepare the demo	17/5/23	20/5/23	Both
Prepare the power point slide show	18/5/23	29/5/23	Both

- **Presentation Slides**





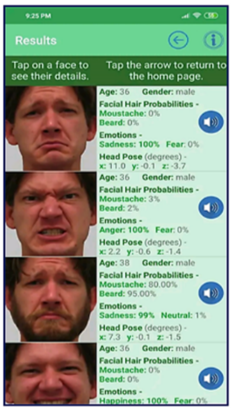
Project Impact

- FER systems are important due to their **impact** on many field.
- In **healthcare**, used to monitor the facial expressions of patients. [3]
- In **social life**, it can help in communicate with others. [4]
- In **education**, used to monitoring students' attention. [5]

3

Related Work

- **Face Analyzer** is Android app that detect faces and provides facial attributes. [6]
  - The **disadvantage** is that results are not always accurate, and it does not work in real-time.
- Where our app stands out is the **lack of bias**.
  - Provide **advice** and helpful **video**.
  - Detect emotions in **real-time**.
  - Sends a **notification** about the user's feeling.



Face Analyzer App [7]

4

Datasets

Dataset	Includes	Total Number of Images	Number of Classes	Resolution	Emotion Classes	Source
FER2013	_	35,887	7	48x48	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	Kaggle
Extended Cohn-Kanade (CK+)	_	902	7	640x490 or 640x480	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	University of California, Berkeley
Japanese Female Facial Expression (JAFFE)	_	213	7	256x256	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	Kyushu University, Japan
Iranian Emotional Faces Database (IEFDB)	_	248	7	5184x3456	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	Tehran University of Medical Sciences, Iran
Hybrid	CK+, JAFFE, and IEFDB Datasets	1363	7	48x48	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	We Created It

5

Datasets Preprocessing

- Preprocessed the datasets images into the **FER2013** dataset images format.
  - **Resized** the images to a size of **48x48** pixels.
  - Converted the **color space** of the images to **grayscale**.
  - **Normalized** the pixel values to a range between 0 and 1.
  - **Split** them as show in the table below.

Dataset	Split Value	Number of Images for IEFDB Dataset	Number of Images for Hybrid Dataset
Train	60%	158	872
Test	20%	50	273
Validation	20%	40	218

6

Retraining and Testing the CNN Model

- We used a **CNN** algorithm that can learn to extract facial features.
- It provides high accuracy to changes in facial expression and lighting. [8]
- Using **Python**, we have retrained and tested the ML model.
- Using the **Scikit-learn** library to train-test split.
- Using the **TensorFlow** library to load a pre-trained kaggle_model file.

7

The First Results

- We first **retrained** the model on the training set of the **IEFDB**.
- Using specific **hyperparameters** values.

Model Hyperparameters Retrained on the IEFDB Train Set

Model Hyperparameters	Values
Learning Rate	0.01
Number of Epochs	400
Batch Size	40
Optimizer	Adam
Loss Function	Categorical Cross-entropy

Accuracies of the Model Retrained on the IEFDB Train Set

Dataset	IEFDB Dataset Accuracy	Hybrid Dataset Accuracy
Train	92%	–
Test Before Retraining the Model	28%	16%
Test After Retraining the Model	74%	33%
Validation	58%	–

8

The Final Results

- To obtain good accuracy, and improve the performance of the ML model.
- We tried to increase the size of the training dataset using the hybrid dataset.
- We **retrained** the model on the training set of the **hybrid dataset**.

Model Hyperparameters Retrained on the Hybrid Train Set

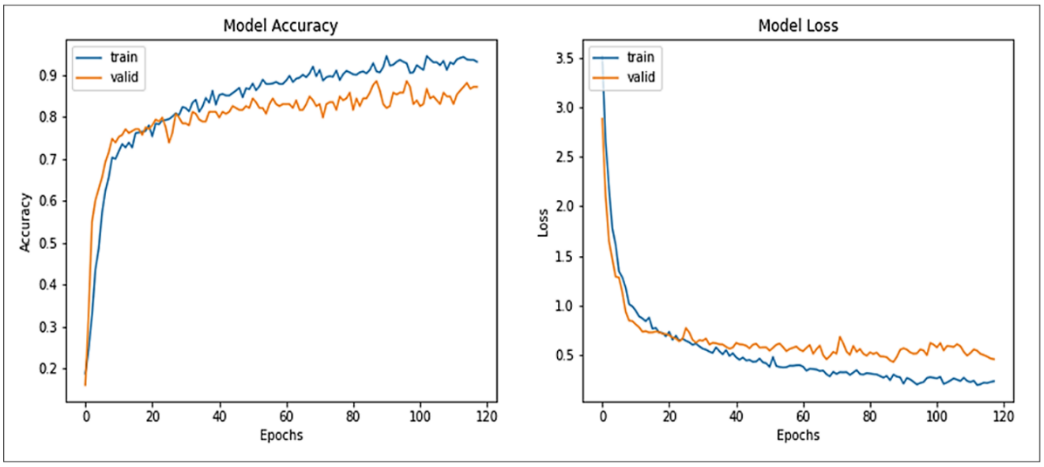
Model Hyperparameters	Values
Learning Rate	0.001
Number of Epochs	300
Batch Size	70
Optimizer	Adam
Loss Function	Categorical Cross-entropy

Accuracies of the Model Retrained on the Hybrid Train Set

Dataset	IEFDB Dataset Accuracy	Hybrid Dataset Accuracy
Train	_	99%
Test Before Retraining the Model	34%	19%
Test After Retraining the Model	90%	88%
Validation	_	87%

9

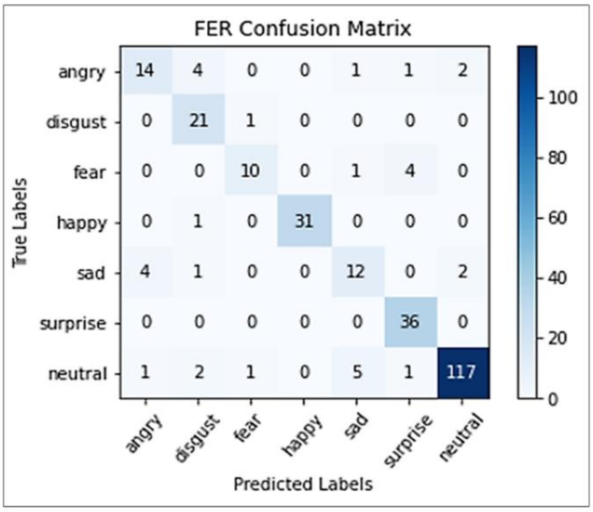
Accuracy and Loss Model Graphs for Training and Validation Sets in the Hybrid Dataset



10

Confusion Matrix for Hybrid Test Set

- The most of the false predictions are in **neutral** emotion.



11

Convert Keras Model to TensorFlow Lite

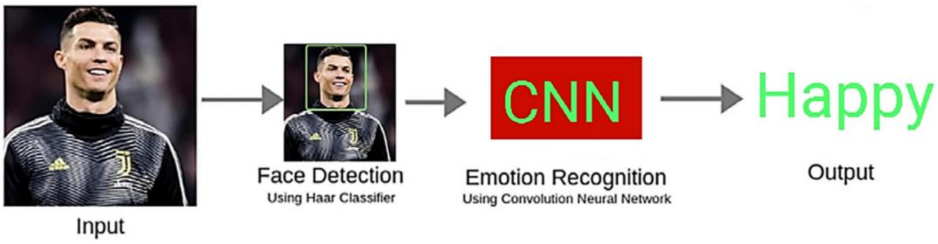
We converted our Keras model to **TFLite** format for use in our Android app based on several steps:

1. Import the TensorFlow Lite converter.
2. Used the **TFLiteConverter** module to convert our Keras model into TFLite format.
3. We add it to our Android app as a model file.
4. In our app, we used the **TensorFlow Lite interpreter** to load the model.

12

Face Detection

- We used the **haarcascade_frontalface_alt.xml** file in our app for face detection.
  - Which is a pre-trained model classifier trained on a large dataset of face images.
  - It is part of the **OpenCV** library.
- This classifier can be used in various computer vision **applications**, such as security systems, and facial recognition. [9]



Face Detection and Emotion Recognition

13

App Design and How it Works

- We developed our app using **Android Studio**, **Java** language, and **XML** code to design the UI.

Screen 1

Screen 2

Screen 3

14

Take a Photo from the Camera

Steps:

- Press the button TAKE PHOTO.
- Press CAPTURE IMAGE button to take a photo.
- To view the results press PROCESS IMAGE button.
- Then, the app displays the captured image, emotion, and advice.
- Press WATCH VIDEO button to opens a video.

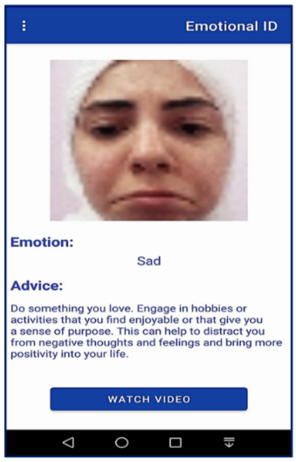


Image That Take by Camera

15

Choose Photo from Gallery

Steps:

- Press the button TAKE PHOTO.
- Press CHOOSE IMAGE FROM GALLERY button to choose a photo.
- To view the results press PROCESS IMAGE button.
- Then, the app displays the chosen image, emotion, and advice.
- Press WATCH VIDEO button to opens a video.

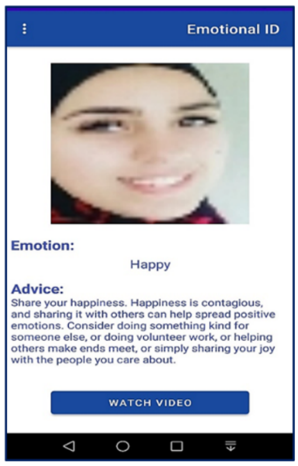


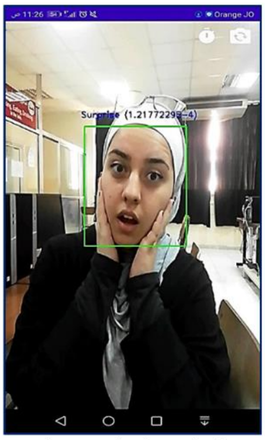
Image That Choose from Gallery

16

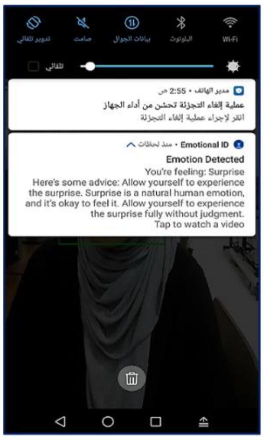
Detects Emotions in Real-Time

Steps:

- Press the button REAL-TIME.
- The camera screen will open.
- Press at the top to flip the camera.
- Press at the top to set the delay time to send notification.
- You will receive a notification containing your emotions and tips.
- Clicking on the notification will open a video for you to watch.



Surprise Emotion in Real-Time

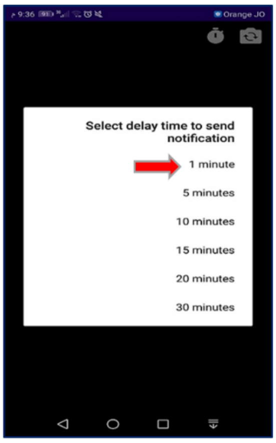


Surprise Notification

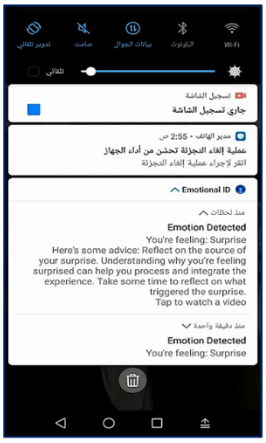
17

Select Delay Time to Send Notification

- You can choose one of these times.
- If we choose **one minute** for example, the notification will be sent after one minute.



Choose a Delay of One Minute



The Notification Appears After One Minute

18

Conclusions

- Our project addressed the challenge of identifying facial expressions by developing a **real-time facial emotion recognition app**.
- Focused on **Middle Eastern faces**, especially for women wearing **hijab**.
- To overcome the **bias** and **inaccuracy**.
- Used the efficient **CNN** algorithm to analyze facial expressions.
- It turned out that the retrained model on a **hybrid dataset** is the most accurate.
- Test accuracy of **88%** was achieved on the **hybrid test set** and test accuracy of **90%** on the **IEFDB test set**.
- We hope that our work will be beneficial for both academic and industrial purposes.

19

Future Work

- Expand the dataset.
- Link to cloud services.
- Integration with wearable devices, such as smartwatches.
- Detect emotions using voice or body language.

20

References

[1] EDPS, K. Vemou, and A. Horvath, "Facial Emotion Recognition," 2021. [Online]. Available: https://edps.europa.eu/data-protection/our-work/publications/techdispatch/techdispatch-12021-facial- emotion-recognition_en. [Accessed 11 November 2022].

[2] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," In 2005 IEEE international conference on multimedia and Expo, pp. 5-pp, IEEE, 2005.

[3] Softwebsolutions, A. Modawal, "Know the benefits of facial recognition and emotion detection tools," 13 May 2022. [Online]. Available: https://www.softwebsolutions.com/resources/benefits-of-facial-recognition.html. [Accessed 18 November 2022].

[4] F. Mo, J. Gu, K. Zhao, and X. Fu, "Confusion effects of facial expression recognition in patients with major depressive disorder and healthy controls," Frontiers in psychology, 12, 2021.

[5] Venturebeat, "Chinese school installs facial recognition cameras to monitor students," 17 May 2018. [Online]. Available: https://venturebeat.com/ai/chinese-school-installs-facial-recognition-cameras-to-monitor-students/. [Accessed 19 November 2022].

[6] Github, "Face Analyzer," [Online]. Available: https://github.com/ishaanjav/Face_Analyzer. [Accessed 4 January 2023].

[7] Apkgk, "Face Analyzer," [Online]. Available: https://apkgk.com/app.anany.faceanalyzer. [Accessed 4 January 2023].

[8] L. Bejjagam, and R. Chakradhara, "Facial Emotion Recognition using Convolutional Neural Network with Multiclass Classification and Bayesian Optimization for Hyper Parameter Tuning," 2022.

[9] Ersanpreet, S. Singh, "Face and eyes detection with Viola Jones along with python code," 10 March 2020. [Online]. Available: https://ersanpreet.wordpress.com/tag/haarcascade_frontalface_default-xml/. [Accessed 28 February 2023].

21

Thank you!

22