



The Egyptian E-Learning University

ـــــــــــــــــــــــــــــ

**Faculty of Computers and Information Technology**

**Face Emotion**

**Classification and Recognition**

**Supervised by:**

Dr. Hanaa Bayomi TA. Engy Emad

**DECLARATION**

We hereby certify that this report, which I now submit for assessment on the program of study leading to the award of Bachelor of Science in Information Technology Bachelor, is all my own work and contains no plagiarism. By submitting this report, I agree to the following terms:

*Any text, diagrams or other material copied from other sources (including, but not limited to, books, journals, and the internet) have been clearly acknowledged and cited followed by the reference number used either in the text or in a footnote/endnote. The details of the used references that are listed at the end of the report are confirming to the referencing style dictated by the final year*

*project template and are, to my knowledge, accurate and complete.*

I have read the sections on referencing and plagiarism in the final year project template. I understand that plagiarism can lead to a reduced or fail grade, in serious cases, for the Graduation Project course.

Student Name: Khaled Mahmoud Khaled Student Name: Abd-El Rahman Mostafa ID: 2000676 ID: 2000117

Student Name: Sara Mohamed Youssef Student Name: Mohamed Moawad Ahmed ID: 2000395 ID: 2000186

Student Name: Mohamed Khaled Hosny Student Name: Ziad Mohamed Abdel Rahim

ID: 2000630 ID: 2000758

Student Name: Ahmed Mohamed Salah Elden  
ID: 1901084

**STUDENTS CONTRIBUTION**

**LET`S FOCUS**

By:

Mohamed Khaled Hosny  
Mohamed Moawad Ahmed

Ahmed Mohamed Salah Elden

Khaled Mahmoud Khaled

Abdel Rahman Mostafa Kamal  
Ziad Mohamed Abdel Rahim

Sara Mohamed Youssef

|  |  |  |
| --- | --- | --- |
| *Chapters* | *Title* | *Contribution* |
| 1 | Introduction | Sara Mohamed Youssef  Ahmed Mohamed Salah Elden |
| 2 | Background | Ziad Mohamed Abdel Rahim Mohamed Khaled Hosny |
| 3 | Analyses and Design | Mohamed Moawad Ahmed  Khaled Mahmoud Khaled  Sara Mohamed Youssef |
| 4 | Similar Systems | Mohamed Moawad Ahmed  Ahmed Mohamed Salah Elden  Abdel Rahman Mostafa Kamal |
| 5 | Implementation | Ahmed Mohamed Salah Elden  Ziad Mohamed Abdel Rahim Mohamed Khaled Hosny  Sara Mohamed Youssef  Khaled Mahmoud Khaled  Abdel Rahman Mostafa Kamal |
| 6 | Conclusion and Future Work | Khaled Mahmoud Khaled  Ahmed Mohamed Salah Elden  Mohamed Moawad Ahmed  Mohamed Khaled Hosny |

**ACKNOWLEDGMENT**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

We highly indebted to (Doctor **Hanaa Bayomi**) for the guidance and constant supervision as well as for providing necessary information regarding the project & also for the support in completing the project.

# I would like to express my gratitude to wards my parents & member of (The Egyptian E-Learning University)

for the kind co- operation and encouragement which help us in completion of this project.

I would like to express my special gratitude and thanks to supervisor`s persons for giving me such attention and time.

My thanks and appreciations also go to my colleagues in developing the project and people who have willingly help me out with their ability

**Table of Contents**

[**1. Introduction to Project**](#_1.1_Introduction) **10**

1.1 Introduction 11

1.2 Objective of the Project 12

1.3 Significance 12

1.4 Methodology 12

1.5 Scope 13

1.6 Motivation 13

[**2.Motivation**](#Motivation) **16**

2.1 Enhancing Human-Computer Interaction 17

2.2 Advancing Mental Health Monitoring 17

2.3 Improving User Experience 17

2.4 Marketing and Entertainment Insights 18

2.5 Contributing to AI and Computer Vision Research 18

2.6 Humanizing Technology 18

2.7 Potential Real-World Applications 19  
2.8 Enhanced Human-Computer Interaction (HCI) 19

2.9 HEALTH CARE 19

2.10 Market Research and Customer Feedback 19

2.11 Security and Surveillance 20

2.12 Education 21

2.13 Entertainment and Media 21

[**3. Problem Definition**](#Problem_Definition) **22**

3.1 Facial Feature Extraction 23

3.2 Data Variability and Diversity 23

3.3 Subtle Emotional Cues 23

3.4 Real-Time Processing 23

3.5 Handling Occlusions 24

3.6 Machine Learning Model Selection 24

3.7 Generalization across Contexts 24

3.8 Ethical Considerations 24

3.9 User Feedback Integration 25

3.10 Validation and Performance Metrics 25  
3.11 Background and Context 25

3.12 Key Challenges 26

3.13 Changing Environmental Conditions 26

3.14 Large and Diverse Data 26

3.15 System Speed and Accuracy 27

[**4.Related Work**](#Related_Work) **27**

[**5. Analysis & Design**](#Analysis_Design) **28**

[**6. Implementation**](#implementation) **29**

6.1 Import Libraries 30  
6.2 Load Images Function Load Images for Each Class 30

6.3 Prepare Dataset &Split Dataset 31

6.4 Preprocess Data 31

6.5 Train Model & Model Prediction and Evaluation 31

6.6 Load and Predict New Image 33

6.7 Import Libraries 34

6.8 Load Images Function & Prepare Dataset 34

6.9 Prepare Dataset &Split Dataset 35

6.10 Preprocess Data 35

6.11 Create Model & Evaluation 37

6.12 Load and Predict New Image 38

6.13 Building a model using Teachable Machine 39

6.14 Creating a mobile application 46

6.15 Link it using TensorflowLite 46

[**7. Objectives**](#Objectives) **47**

7.1 High Accuracy and Precision: 48

7.2 Real-Time Processing 48

7.3 Robustness to Facial Expression Variability 49

7.4 Diverse and Representative Training Data 49

7.5 Cross-Cultural Applicability 49

7.6 Privacy Protection 49

7.7 Ethical AI Practices 50

7.8 Adaptability to Individual Differences 50

7.9 User-Friendly Integration 50

7.10 Continuous Model Improvement 50

7.11 Environmental Robustness 50

7.12 Emotion Ambiguity Handling 50

7.13 Interdisciplinary Collaboration 51

7.14 Validation and Performance Metrics 51

7.15 Integration with Human Behavior Analysis 51

[**8. Background**](#Background) **52**

8.1 what is SVM? 53

8.2 How Does an SVM Work? 54

8.3 Types of SVM System 56

8.4 Advantages of SVM 58

8.5 Disadvantages of SVM 59

8.6 What is CNN 61

8.7 How does CNN Work 63

8.8 Advantage of CNN 63

8.9 Desadvantage of CNN 64

8.10 What is Tenserflow 65

8.11 What is Flutter 69

8.12 What is Dart 73

[**9.Future Work**](#Future_Work) **76**

[**10.Conclusion**](#Conclusion)

[**11.References**](#Reference)

# Table of Figure

# Figure 1……………………………………………………………………..11

# Figure 2……………………………………………………………………..12

# Figure 3……………………………………………………………………..14

# Figure 4……………………………………………………………………..18

# Figure 5……………………………………………………………………..20

# Figure 6……………………………………………………………………..24

# Figure 7……………………………………………………………………..26

# Figure 8……………………………………………………………………..29

# Figure 9……………………………………………………………………..31

# Figure 10…………………………………………………………………….31 Figure 11…………………………………………………………………….32 Figure 12…………………………………………………………………….32 Figure 13…………………………………………………………………….33 Figure 14…………………………………………………………………….33 Figure 15…………………………………………………………………….33 Figure 16…………………………………………………………………….34 Figure 17…………………………………………………………………….34 Figure 18…………………………………………………………………….35 Figure 19…………………………………………………………………….36 Figure 20…………………………………………………………………….36 Figure 21…………………………………………………………………….37 Figure 22…………………………………………………………………….37 Figure 23…………………………………………………………………….38 Figure 24…………………………………………………………………….38 Figure 25…………………………………………………………………….39 Figure 26…………………………………………………………………….39 Figure 27…………………………………………………………………….40 Figure 28…………………………………………………………………….40 Figure 29…………………………………………………………………….41 Figure 30…………………………………………………………………….41 Figure 31…………………………………………………………………….42 Figure 32…………………………………………………………………….42 Figure 33…………………………………………………………………….43 Figure 34…………………………………………………………………….43 Figure 35…………………………………………………………………….44 Figure 36…………………………………………………………………….44 Figure 37…………………………………………………………………….45 Figure 38…………………………………………………………………….45 Figure 39…………………………………………………………………….47 Figure 40…………………………………………………………………….48 Figure 41…………………………………………………………………….50 Figure 42…………………………………………………………………….52 Figure 43…………………………………………………………………….54 Figure 44…………………………………………………………………….60 Figure 45…………………………………………………………………….65 Figure 46…………………………………………………………………….68

**1. Introduction to Project**

**ـــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــ**

### 1.1 Introduction

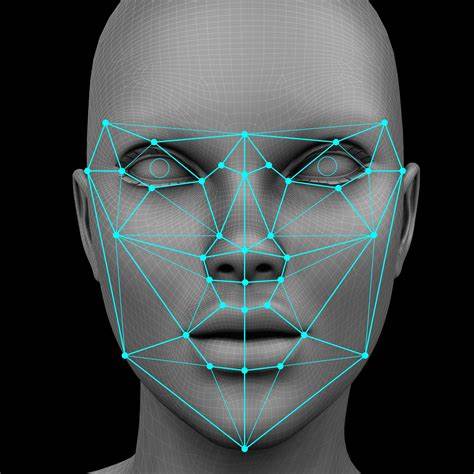
### 1.2 Objective of the Project

1.3 Significance

1.4 Methodology

1.5 Scope

1.6 Motivation

****

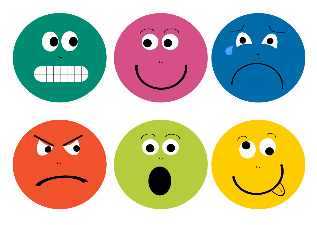
**1.1 Introduction**

Recent In years, the field of computer vision has witnessed significant advancements, particularly in the domain of facial emotion recognition. The ability to interpret and understand human emotions through facial expressions holds great promise for a wide range of applications, from human-computer interaction to mental health monitoring. This project aims to delve into the fascinating realm of face emotion analysis, leveraging cutting-edge technologies and methodologies to develop a robust and accurate system for recognizing and interpreting facial expressions.

Facial emotion recognition involves the detection and analysis of facial features, such as eye movements, mouth expressions, and overall facial muscle activity, to infer the emotional state of an individual. This capability has vast implications for industries like human-computer interaction, marketing, healthcare, and even security.

**1.2 Objective of the Project:**

The primary objective of this project is to design, implement, and evaluate a facial emotion recognition system capable of accurately identifying and categorizing a person's emotional state based on their facial expressions. By harnessing the power of machine learning and computer vision techniques, we aim to create a model that can not only recognize basic emotions such as happiness, sadness, anger, surprise, fear, and disgust but also provide insights into subtle variations within these emotional states.



**1.3 Significance:**

Understanding human emotions is crucial in various fields. In human-computer interaction, for instance, an intelligent system that can perceive and respond to the user's emotional state can enhance the overall user experience. In healthcare, facial emotion recognition can be utilized as a non-intrusive tool for monitoring mental health conditions. Additionally, in marketing and entertainment, recognizing audience reactions through facial expressions can provide valuable insights for content optimization.

**1.4 Methodology:**

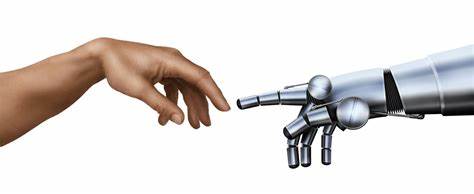
The project will employ a combination of image processing techniques, deep learning models, and possibly real-time data acquisition to achieve its objectives. Various datasets containing labeled facial expressions will be used for training and testing the model. The project will explore different algorithms and architectures, fine-tuning them to achieve the highest accuracy in recognizing and classifying facial emotions.

**1.5 Scope:**

As the project progresses, it will explore the potential applications of facial emotion recognition in real-world scenarios. This includes but is not limited to human-computer interaction systems, virtual reality environments, mental health assessment tools, and personalized marketing strategies.

**1.6 Motivation**

Understanding and interpreting human emotions play a crucial role in various aspects of our daily lives. As we increasingly integrate technology into our interactions, the ability to develop systems that comprehend and respond to human emotions becomes essential. The motivation behind undertaking a project on face emotion recognition stems from several key factors:

1. **Enhancing Human-Computer Interaction:**
   1. The project aims to contribute to the development of intelligent systems capable of understanding human emotions through facial expressions. This has the potential to revolutionize human-computer interaction by creating interfaces that respond empathetically to users, making technology more user-friendly and intuitive.
2. **Advancing Mental Health Monitoring:**
   1. Facial emotion recognition can serve as a valuable tool in the field of mental health. By analyzing facial expressions, the project seeks to explore the possibility of creating non-intrusive systems for monitoring emotional well-being. This could assist in early detection of mental health issues and provide support when needed.
3. **Improving User Experience:**
   1. In applications ranging from virtual reality to online platforms, the ability to recognize and respond to user emotions can significantly enhance the overall user experience. This project aims to contribute to the creation of emotionally intelligent systems that adapt to users' emotional states.
4. **Marketing and Entertainment Insights:**
   1. Recognizing facial expressions can provide valuable insights for marketers and content creators. Understanding how individuals react emotionally to advertisements, products, or entertainment content can inform strategies for personalized marketing and content optimization.
5. **Contributing to AI and Computer Vision Research:**
   1. Face emotion recognition represents a challenging problem in the domains of artificial intelligence and computer vision. By undertaking this project, there is an opportunity to explore and contribute to cutting-edge research in these fields, pushing the boundaries of what is possible in automated emotion analysis.  
        
      
6. **Humanizing Technology:**
   1. Integrating emotion recognition capabilities into technology contributes to humanizing our digital interactions. This project seeks to bridge the gap between human emotions and machines, fostering a more empathetic and responsive technological landscape.
7. **Potential Real-World Applications:**
   1. The project has the potential to influence various industries, including healthcare, education, customer service, and entertainment. By developing an accurate and reliable face emotion recognition system, the technology can find practical applications in real-world scenarios.

In conclusion, the motivation for the face emotion recognition project is rooted in the desire to create technology that is more attuned to human emotions, ultimately enhancing the way we interact with and benefit from modern digital systems

**2.Motivation**

**ـــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــ**

2.1 Enhancing Human-Computer Interaction

2.2 Advancing Mental Health Monitoring

2.3 Improving User Experience

2.4 Marketing and Entertainment Insights

2.5 Contributing to AI and Computer Vision Research

2.6 Humanizing Technology

2.7 Potential Real-World Applications

2.8 Enhanced Human-Computer Interaction (HCI)

2.9 HEALTH CARE

2.10 Market Research and Customer Feedback

2.11 Security and Surveillance

2.12 Education

2.13 Entertainment and Media

Understanding and interpreting human emotions play a crucial role in various aspects of our daily lives. As we increasingly integrate technology into our interactions, the ability to develop systems that comprehend and respond to human emotions becomes essential. The motivation behind undertaking a project on face emotion recognition stems from several key factors

2.1 **Enhancing Human-Computer Interaction:**

The project aims to contribute to the development of intelligent systems capable of understanding human emotions through facial expressions. This has the potential to revolutionize human-computer interaction by creating interfaces that respond empathetically to users, making technology more user-friendly and intuitive.

**2.2 Advancing Mental Health Monitoring:**

Facial emotion recognition can serve as a valuable tool in the field of mental health. By analyzing facial expressions, the project seeks to explore the possibility of creating non-intrusive systems for monitoring emotional well-being. This could assist in early detection of mental health issues and provide support when needed.

**2.3 Improving User Experience:**

In applications ranging from virtual reality to online platforms, the ability to recognize and respond to user emotions can significantly enhance the overall user experience. This project aims to contribute to the creation of emotionally intelligent systems that adapt to users' emotional states.

**2.4 Marketing and Entertainment Insights:**

Recognizing facial expressions can provide valuable insights for marketers and content creators. Understanding how individuals react emotionally to advertisements, products, or entertainment content can inform strategies for personalized marketing and content optimization.

**2.5 Contributing to AI and Computer Vision Research:**

Face emotion recognition represents a challenging problem in the domains of artificial intelligence and computer vision. By undertaking this project, there is an opportunity to explore and contribute to cutting-edge research in these fields, pushing the boundaries of what is possible in automated emotion analysis.

**2.6 Humanizing Technology:**

Integrating emotion recognition capabilities into technology contributes to humanizing our digital interactions. This project seeks to bridge the gap between human emotions and machines, fostering a more empathetic and responsive technological landscape.  
  


**2.7 Potential Real-World Applications:**

The project has the potential to influence various industries, including healthcare, education, customer service, and entertainment. By developing an accurate and reliable face emotion recognition system, the technology can find practical applications in real-world scenarios.

**2.8 Enhanced Human-Computer Interaction (HCI):**

FER improves the interaction between humans and computers by enabling systems to recognize and respond to user emotions, making interactions more natural and intuitive. This is particularly valuable in areas such as virtual assistants, customer service bots, and interactive gaming.

**2.9 HEALTH CARE:**

In healthcare, FER can assist in patient care by monitoring the emotional well-being of patients, especially those who may have difficulty communicating verbally. This is particularly useful for patients with neurological disorders or those recovering from surgeries.  
  
**2.10 Market Research and Customer Feedback:**

Businesses use FER to gauge customer reactions to products, advertisements, and services. By analyzing facial expressions, companies can obtain immediate and accurate feedback on consumer preferences and emotions, leading to better marketing strategies and product development.

**2.11 Security and Surveillance:**

In security applications, FER can be employed to identify suspicious or potentially dangerous behavior by analyzing facial expressions in real-time. This enhances public safety and assists in crime prevention  
  
  
  
  
  
  
  
.

**2.12 Education:**

In educational environments, FER helps in understanding student engagement and emotions during learning processes. This can lead to more personalized and effective teaching strategies, improving educational outcomes.

**2.13 Entertainment and Media:**

The entertainment industry uses FER to create more immersive experiences in video games and movies by capturing and replicating realistic human emotions. Additionally, it assists in content recommendation by analyzing viewers' reactions.

**3. Problem Definition**

**ـــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــ**

**3.1 Facial Feature Extraction**

**3.2 Data Variability and Diversity**

**3.3 Subtle Emotional Cues**

**3.4 Real-Time Processing**

**3.5 Handling Occlusions**

**3.6 Machine Learning Model Selection**

**3.7 Generalization across Contexts**

**3.8 Ethical Considerations**

**3.9 User Feedback Integration**

**3.10 Validation and Performance Metrics**

**3.11 Background and Context**

**3.12 Key Challenges**

**3.13 Changing Environmental Conditions**

**3.14 Large and Diverse Data**

**3.15 System Speed and Accuracy**

The face emotion recognition project addresses the challenge of developing a system that can accurately identify and interpret human emotions based on facial expressions. The project aims to tackle several key problems within this domain:

**3.1 Facial Feature Extraction:**

One of the primary challenges is the extraction of relevant facial features that contribute to expressing emotions. This involves identifying key regions of the face, such as the eyes, mouth, and eyebrows, and understanding how these features change during different emotional states.

**3.2 Data Variability and Diversity:**

Facial expressions can vary significantly among individuals and across cultures. The project needs to account for this variability by utilizing diverse datasets that capture a wide range of facial expressions from different demographics, ethnicities, and backgrounds.

**3.3 Subtle Emotional Cues:**

Recognizing subtle emotional cues poses a challenge, as some emotions may be expressed with minimal facial changes. The system needs to be sensitive enough to detect nuanced expressions and differentiate between similar emotions.

**3.4 Real-Time Processing:**

Achieving real-time processing for emotion recognition is crucial for applications like human-computer interaction and virtual reality. The project needs to address the computational complexities associated with analyzing facial expressions quickly and efficiently.

**3.5 Handling Occlusions:**

Facial occlusions, such as glasses or facial hair, can obstruct crucial facial features. The system should be robust enough to handle these occlusions and still accurately recognize emotions.

**3.6 Machine Learning Model Selection:**

Ensuring the ethical use of face emotion recognition technology is essential. The project needs to address potential biases in the data and model predictions, as well as privacy concerns related to the collection and use of facial data.

**3.7 Generalization across Contexts:**

The system should generalize well across different contexts and scenarios. It should be trained on diverse datasets to ensure that it can accurately recognize emotions in various environments, lighting conditions, and situations.

**3.8 Ethical Considerations:**

Ensuring the ethical use of face emotion recognition technology is essential. The project needs to address potential biases in the data and model predictions, as well as privacy concerns related to the collection and use of facial data.

****

**3.9 User Feedback Integration:**

Integrating user feedback into the system is crucial for continuous improvement. The project should explore methods to incorporate user feedback and adapt the model over time to enhance its performance and responsiveness.

**3.10 Validation and Performance Metrics:**

Establishing robust validation methodologies and performance metrics is essential. The project needs to define criteria for evaluating the accuracy, precision, recall, and overall performance of the face emotion recognition system.

**3.11 Background and Context:**

Facial Expression Recognition (FER) technology is a branch of artificial intelligence and computer vision that focuses on analyzing facial expressions to identify human emotions. FER is used in a wide range of applications such as human-computer interaction, healthcare, education, entertainment, security, and surveillance. Despite the immense potential of this technology, there are numerous challenges and issues that need to be addressed to achieve high accuracy and efficiency.

**3.12 Key Challenges:**

Variability in Human Expressions:

Facial expressions vary significantly among individuals based on age, gender, cultural background, and personal traits.

The need for a system capable of adapting to these differences to achieve high accuracy.

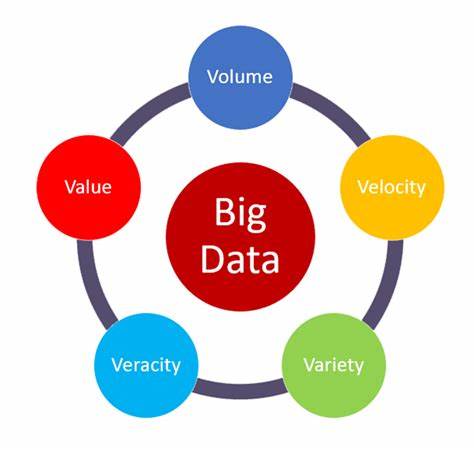
**3.13 Changing Environmental Conditions:**

Lighting, angles, distances, and backgrounds can greatly affect recognition accuracy.

The need for advanced image processing and enhancement techniques to overcome these challenges.

**3.14 Large and Diverse Data:**

Collecting, classifying, and processing large amounts of facial data to enhance model training.

Ensuring that the data used is diverse and comprehensive to represent all categories fairly.  


**3.15 System Speed and Accuracy:**

Meeting real-time requirements, especially in critical applications such as security and surveillance.

Achieving a balance between high accuracy and processing speed

**4.Related Work.**

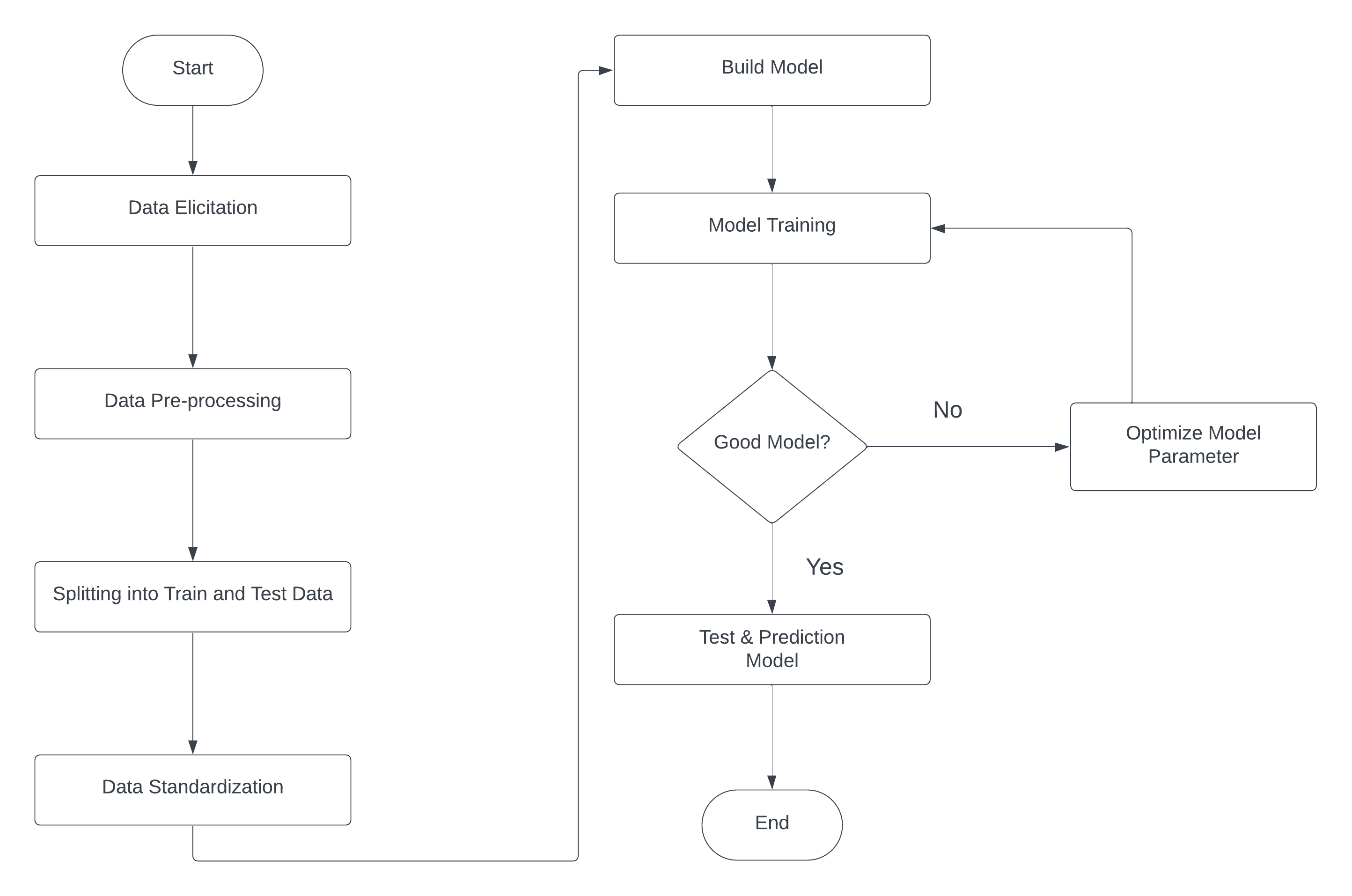
ـــــــــــــــــــــــــــــــــــــــــــــــــ

There were others who used the same dataset with the algorithm CNN, and it yielded results for them as well Discusses the implementation and testing of the proposed architecture on the MMI datasets. The architecture achieved an average accuracy of 98.63% on the MMI dataset.

There are those who used a database (CKP) and they used the algorithm and gave equal accuracy 99.6%

**5. Analysis & Design**

ـــــــــــــــــــــــــــــــــــــــــــــــــ



**6. Implementation**

ـــــــــــــــــــــــــــــــــــــــــــــــــ

**Phase 01**

**6.1 Import Libraries**

**6.2 Load Images Function Load Images for Each Class**

**6.3 Prepare Dataset &Split Dataset**

**6.4 Preprocess Data**

**6.5 Train Model & Model Prediction and Evaluation**

**6.6 Load and Predict New Image**

**Phase 02 (Part1)**

**6.7 Import Libraries**

**6.8 Load Images Function & Prepare Dataset**

**6.9 Prepare Dataset &Split Dataset**

**6.10 Preprocess Data**

**6.11 Create Model & Evaluation**

**6.12 Load and Predict New Image**

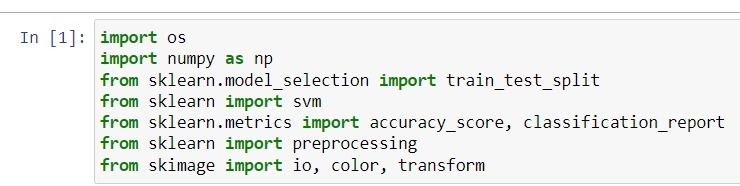
**Phase 02 (Part2)**

**6.13 Building a model using Teachable Machine**

**6.14 Creating a mobile application**

**6.15 Link it using TensorflowLite**

**6.1 Import libraries:**

Import necessary libraries like os, numpy, train\_test split 

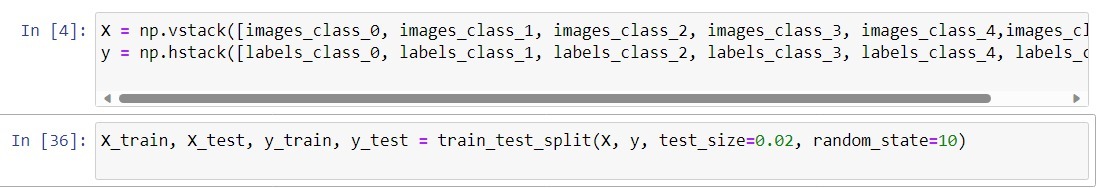
6.2 **Load Images Function Load Images for Each Class**

* A function load\_images is defined to load images from a given folder, convert them to grayscale, resize, flatten, and associate them with labels
* Load images from different folders for each class (0 to 5).

****

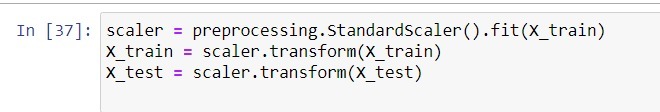
**6.3 Prepare dataset &Split dataset:**

* Combine images from all classes into a single dataset
* Split the dataset into training and test sets using train\_test\_split



**6.4 Preprocess data:**

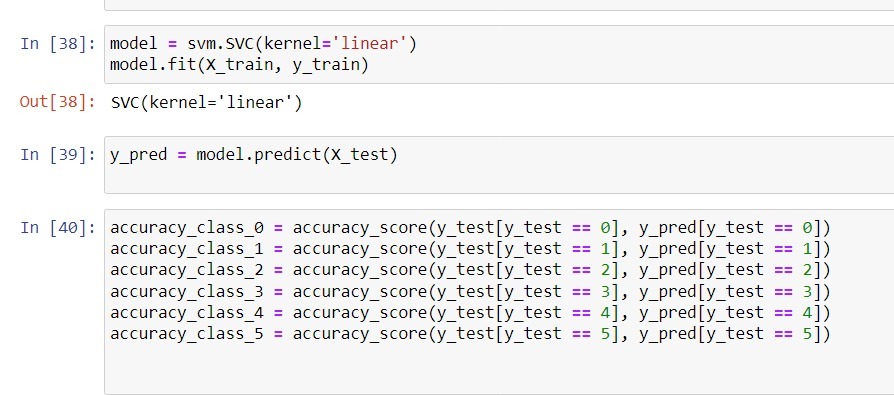
* Scale the data using StandardScaler

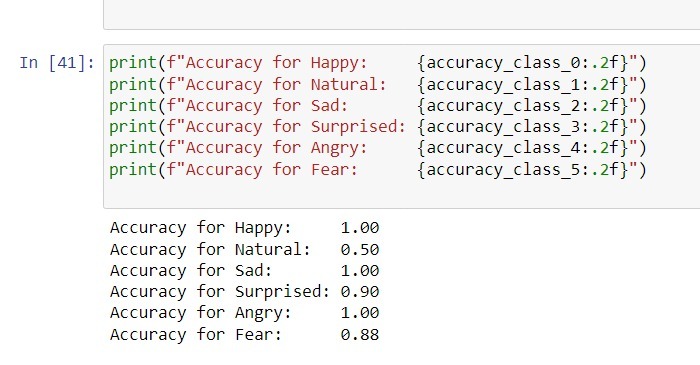


**6.5 Train model & Model prediction and evaluation:**

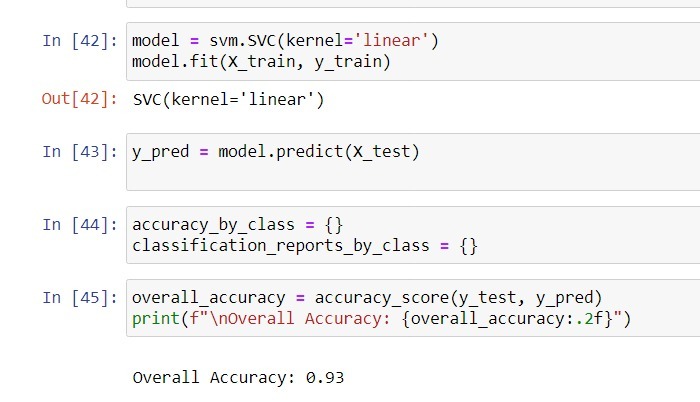
* Train a Support Vector Machine (SVM) model with a linear kernel on the training data.
* Predict on the test set and evaluate the model's performance for each class individually and overall.

**Accuracy for each category:**

****

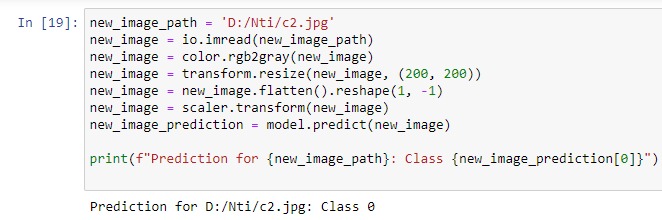
****

**For overall accuracy:**



**6.6 Load and Predict New Image:**

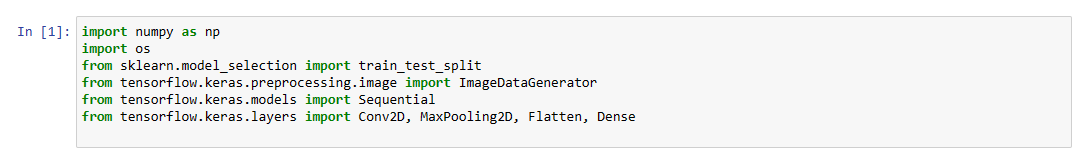
* Load a new image, preprocess it in the same way as the training data, and predict its class using the trained model.

****

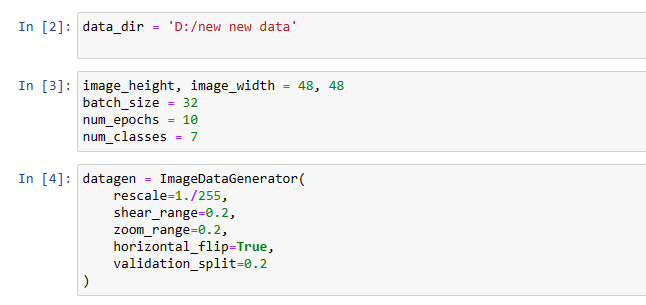
****

**6.7 Import libraries:**

Import necessary libraries like os, numpy, train\_test split, and ImageDataGenerator



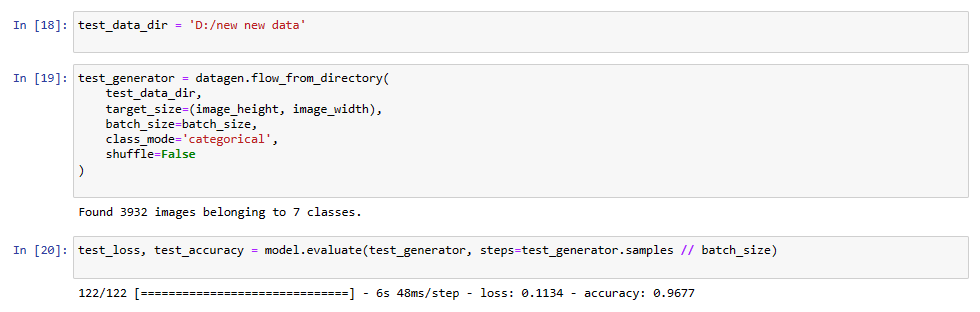
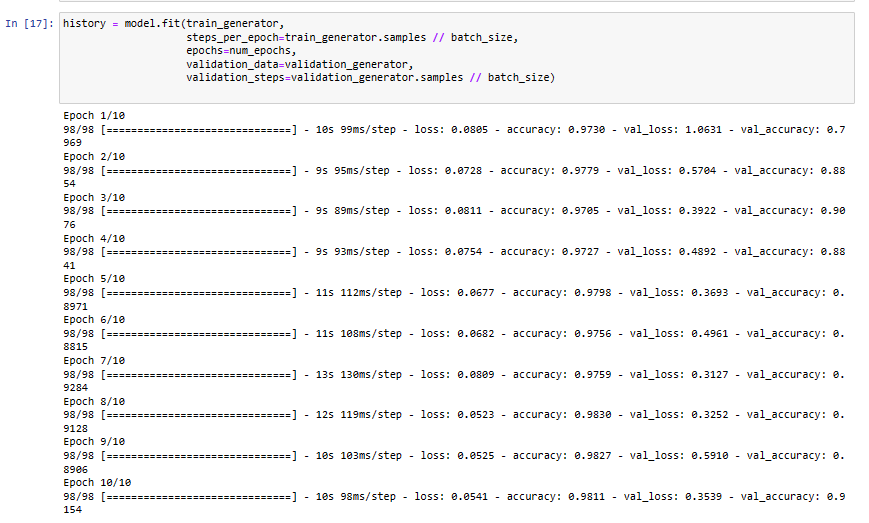
**6.8 Load Images Function & Prepare Dataset:**



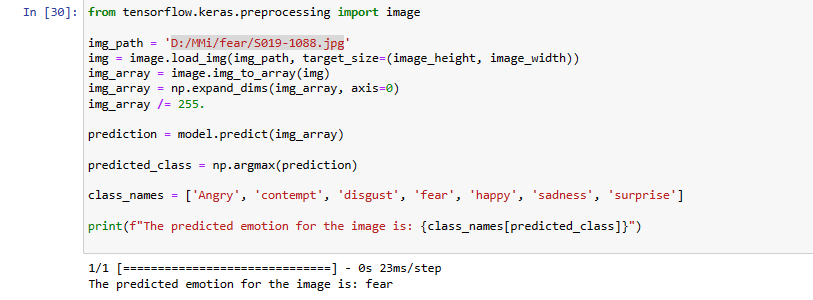
**6.9 Prepare Dataset &Split Dataset:**



**6.10 Create Model & evaluation:**

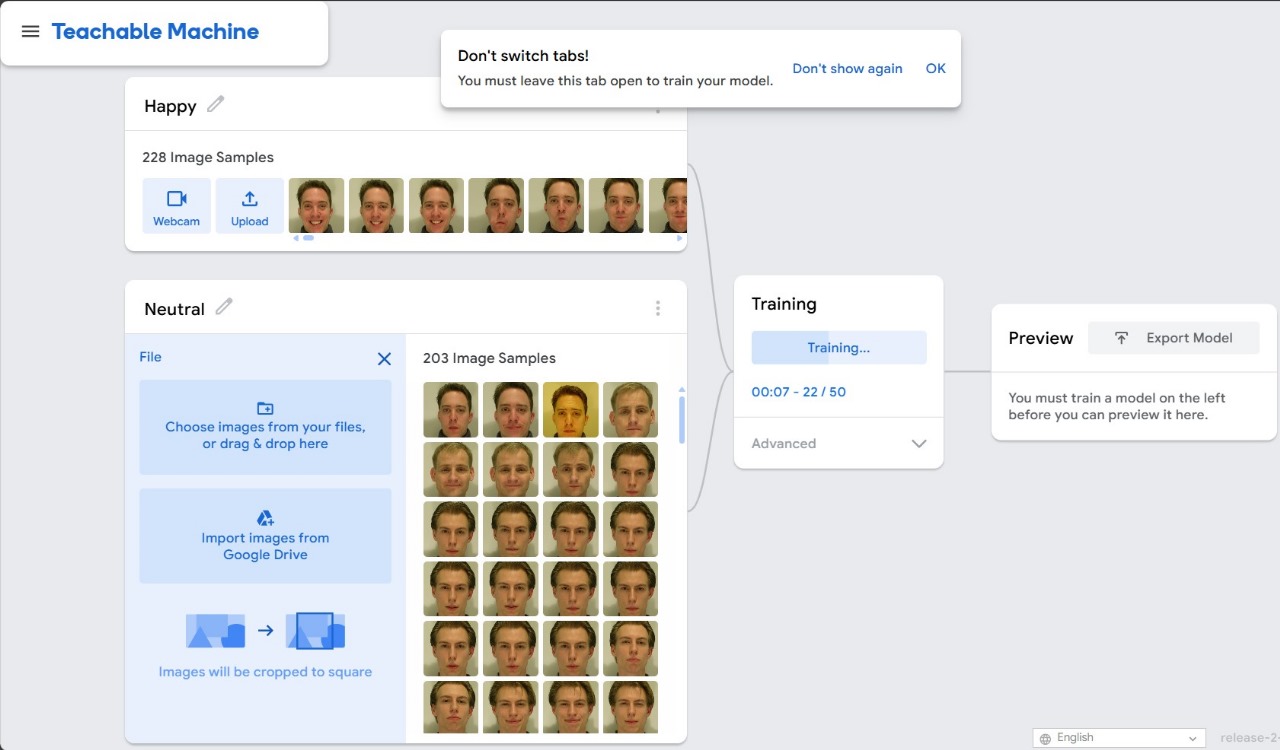


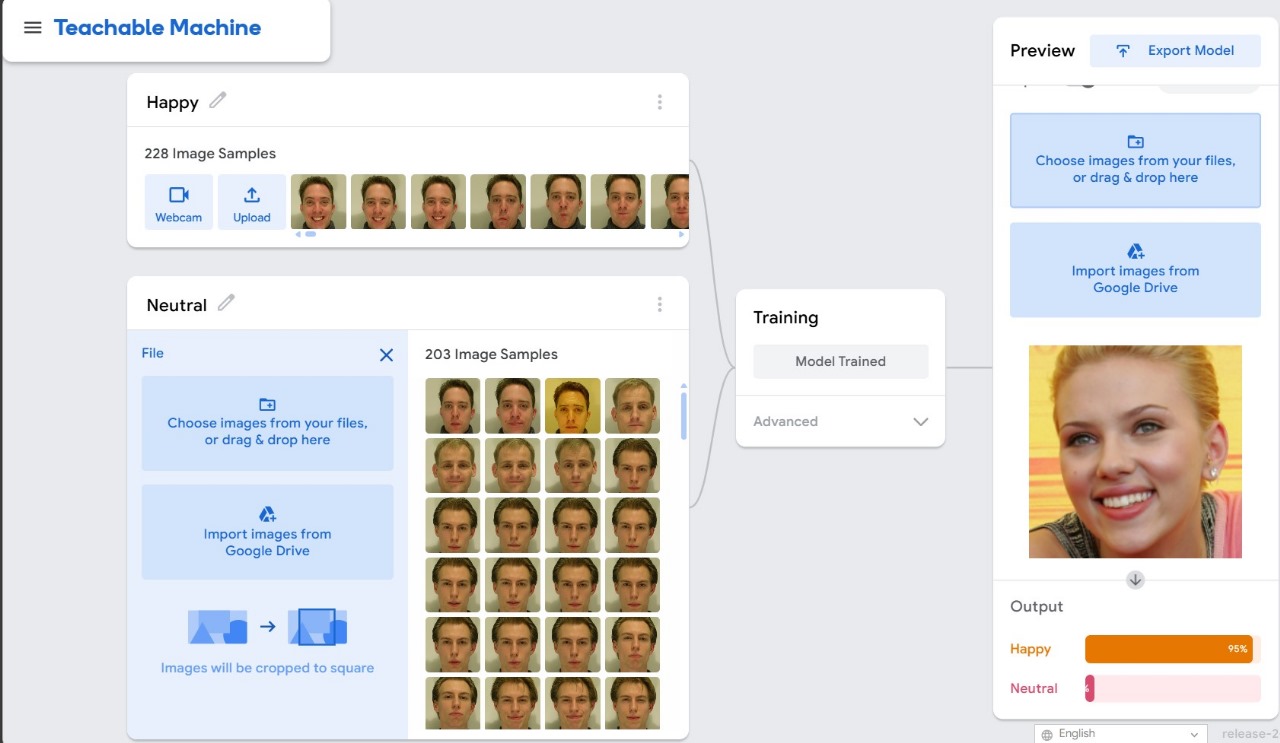
**6.11 Load and Predict New Image:**



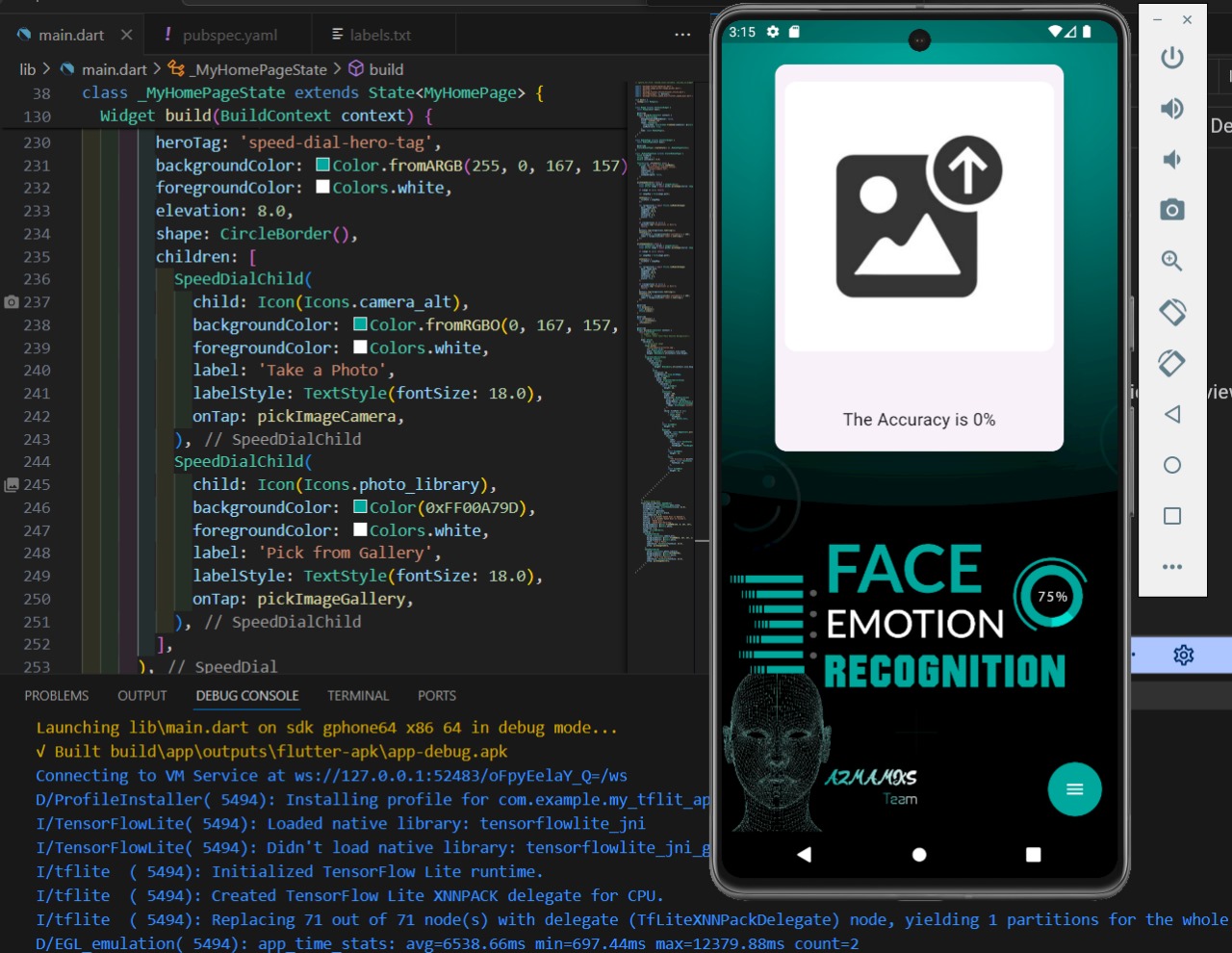
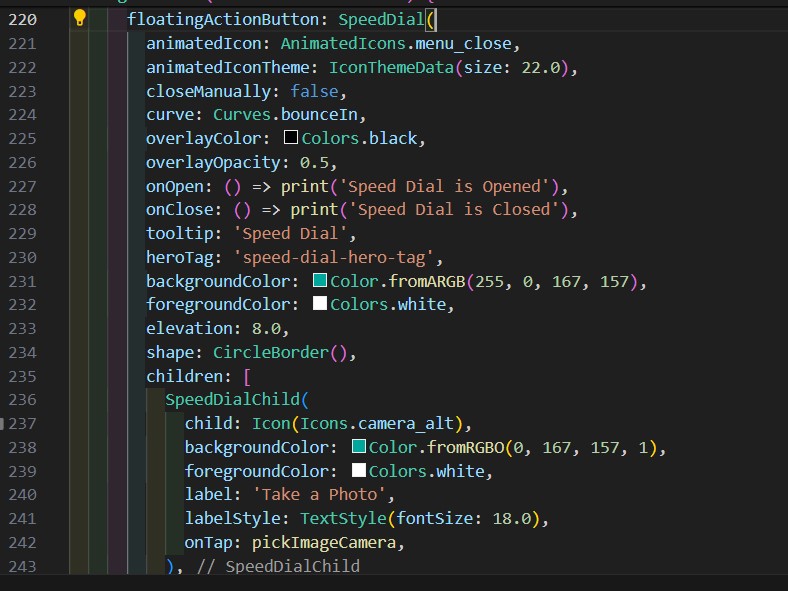
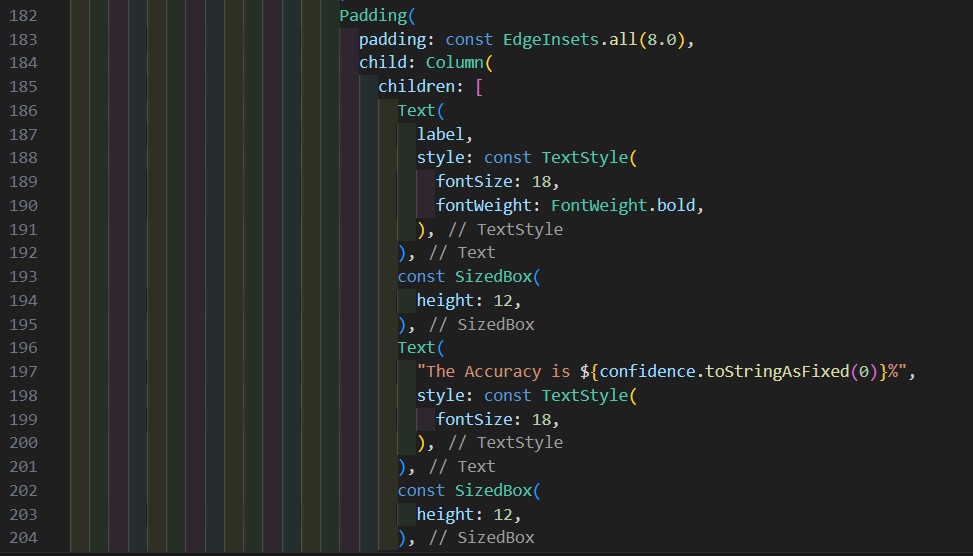
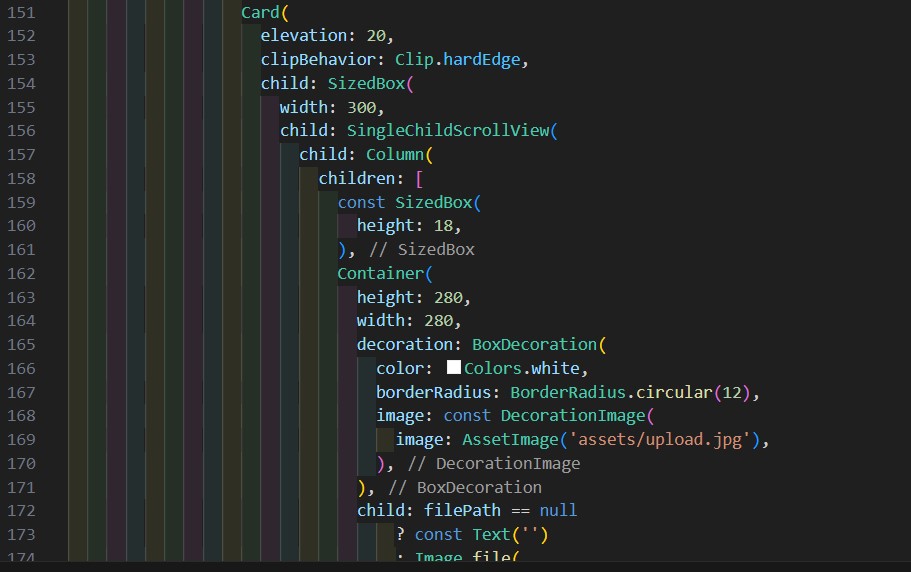
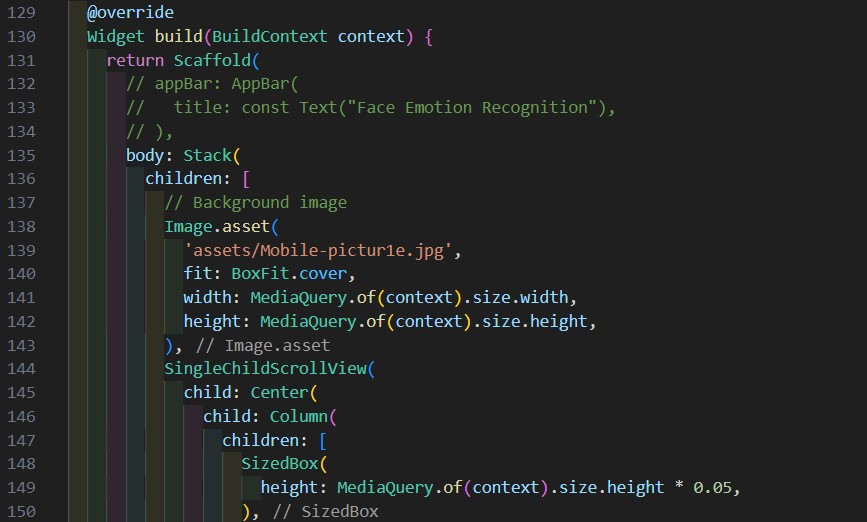
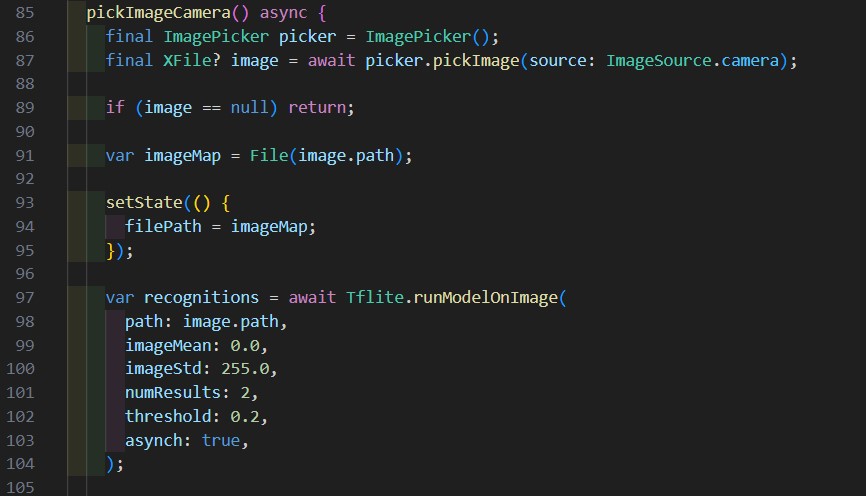
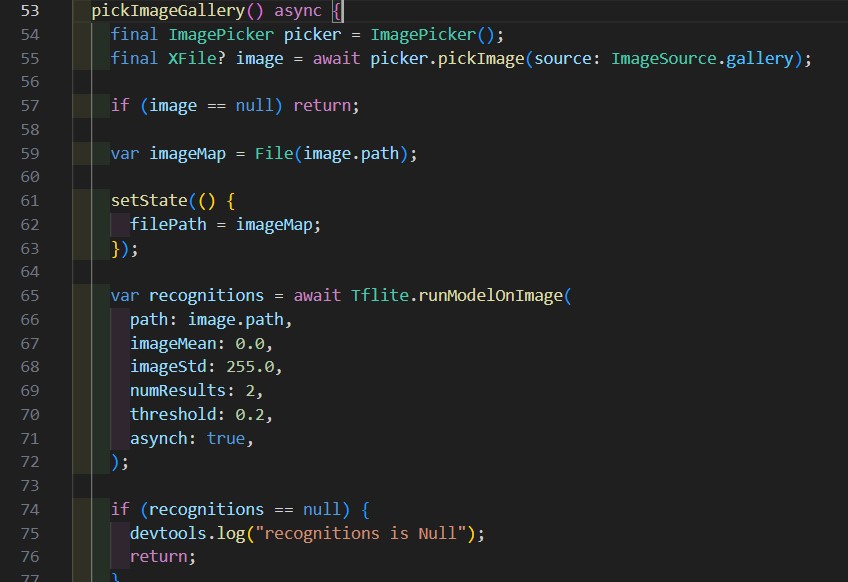
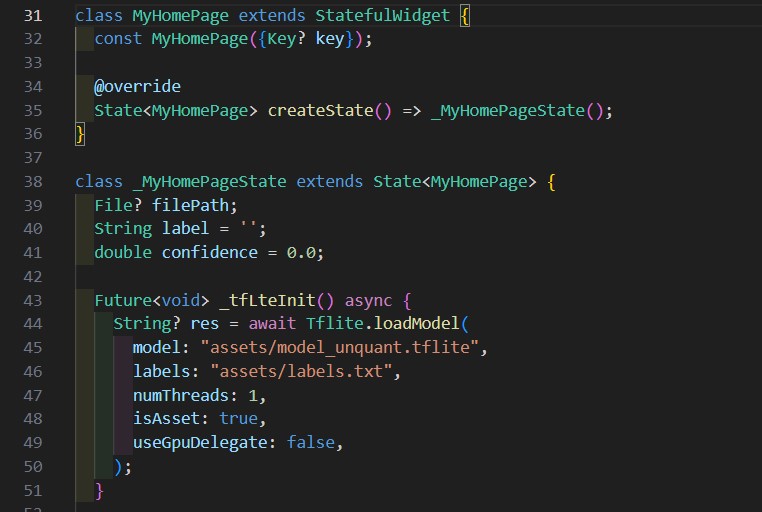
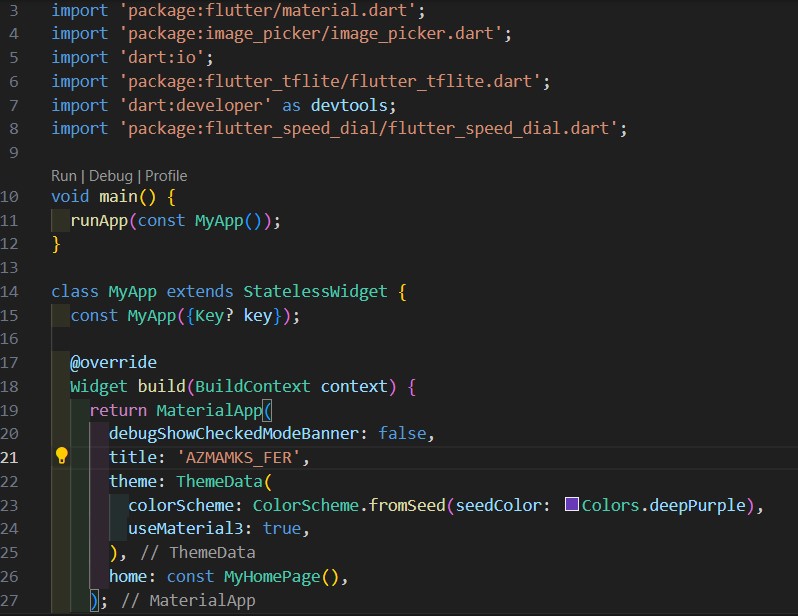


**6.12 Building a model using Teachable Machine:**

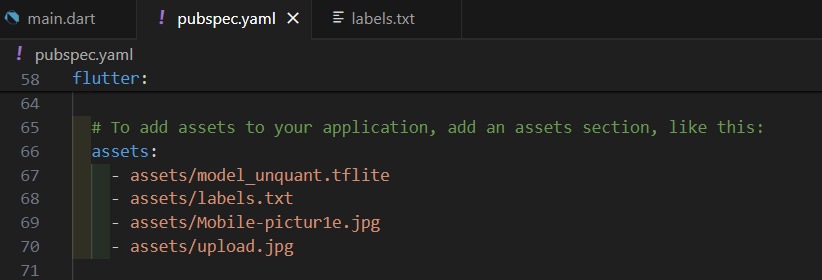
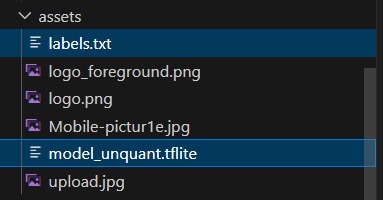




**6.13 Creating a mobile application:**



**6.14 Link it using TensorflowLite:**



**7. Objectives**

ـــــــــــــــــــــــــــــــــــــــــــــــــ

**7.1 High Accuracy and Precision:**

**7.2 Real-Time Processing:**

**7.3 Robustness to Facial Expression Variability:**

**7.4 Diverse and Representative Training Data:**

**7.5 Cross-Cultural Applicability:**

**7.6 Privacy Protection:**

**7.7 Ethical AI Practices:**

**7.8 Adaptability to Individual Differences:**

**7.9 User-Friendly Integration:**

**7.10 Continuous Model Improvement:**

**7.11 Environmental Robustness:**

**7.12 Emotion Ambiguity Handling:**

**7.13 Interdisciplinary Collaboration:**

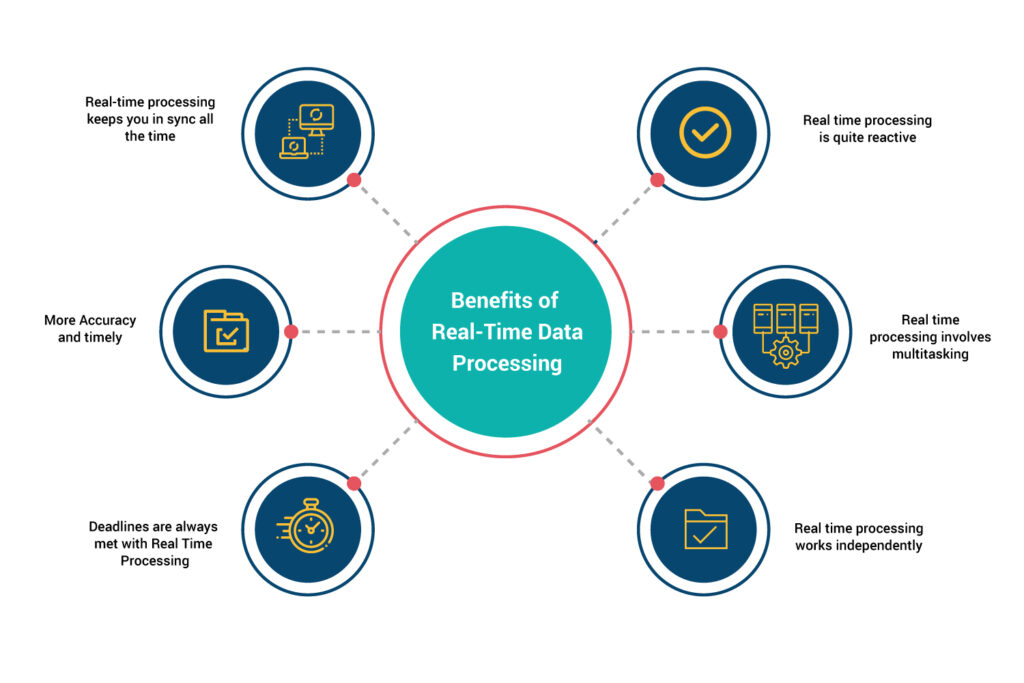
**7.14 Validation and Performance Metrics:**

**7.15 Integration with Human Behavior Analysis:**

**7.1 High Accuracy and Precision:**

Develop a face emotion recognition model with high accuracy and precision, ensuring reliable identification and interpretation of various emotional expressions

**7.2 Real-Time Processing:**

Achieve real-time processing capabilities to enable seamless integration of the emotion recognition system in applications such as human-computer interaction, virtual reality, and live video analysis.  
  
  
  


**7.3 Robustness to Facial Expression Variability:**

Enhance the model's robustness to accommodate diverse facial expressions, accounting for individual differences, cultural variations, and the subtleties of emotional nuances.

**7.4 Diverse and Representative Training Data:**

Collect and utilize diverse and representative datasets to train the model, addressing biases and ensuring that the system is inclusive and applicable across different demographics.

**7.5 Cross-Cultural Applicability:**

Ensure that the face emotion recognition model is culturally sensitive and can accurately recognize emotions expressed across various cultural contexts

7**.6 Privacy Protection:**

Implement strong privacy measures to protect user data and address privacy concerns associated with the collection and storage of facial information.  
  


**7.7 Ethical AI Practices:**

Incorporate ethical AI principles throughout the project, including transparency, fairness, and user consent. Ensure that the technology is developed and deployed responsibly.

**7.8 Adaptability to Individual Differences:**

Develop the model to adapt to individual differences in facial expressions, considering factors such as age, gender, ethnicity, and personal idiosyncrasies.

**7.9 User-Friendly Integration:**

Design the face emotion recognition system to be user-friendly and easily integrable into various applications and platforms, promoting widespread adoption.

**7.10 Continuous Model Improvement:**

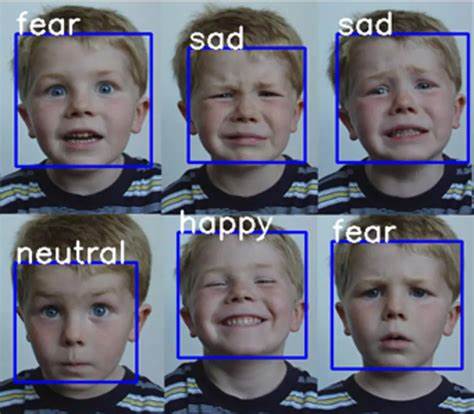
Establish mechanisms for continual model improvement based on user feedback, ensuring that the system evolves to better recognize and interpret facial expressions over time.

**7.11 Environmental Robustness:**

Improve the model's robustness to environmental factors, such as varying lighting conditions and background settings, to enhance its performance in different scenarios.

**7.12 Emotion Ambiguity Handling:**

Develop mechanisms to handle ambiguous facial expressions that may convey multiple emotions, ensuring that the system can provide accurate interpretations in complex situations.

**  
7.13 Interdisciplinary Collaboration:**

Foster collaboration with experts from psychology, sociology, and ethics to gain insights into human behavior, cultural nuances, and ethical considerations related to emotion recognition.

**7.14 Validation and Performance Metrics:**

Define rigorous validation methodologies and performance metrics to objectively assess the accuracy, precision, recall, and overall performance of the face emotion recognition system.

**7.15 Integration with Human Behavior Analysis:**

Explore the integration of facial emotion recognition with broader human behavior analysis to provide a more comprehensive understanding of emotional states in diverse contexts.

**8. Background**

ـــــــــــــــــــــــــــــــــــــــــــــــــ

Phase 01

**8.1 What is SVM?**

**8.2 How does SVM Work?**

**8.3 Types of SVM System**

**8.4 Advantages of SVM**

**8.5 Disadvantages of SVM**

**Phase 02**

**8.6 What is CNN**

**8.7 How does CNN Work**

**8.8 Advantage of CNN**

**8.9 Desadvantage of CNN**

**8.10 What is Tenserflow**

**8.11 What is Flutter**

**8.12 What is Dart**

**8.1 What is SVM?**  
SVM stands for Support Vector Machine, which is a supervised machine learning algorithm used for classification and regression tasks. The primary objective of an SVM is to find a hyperplane in a high-dimensional space that effectively separates different classes in the input data.



**8.2 How Does an SVM Work?**

The working principle of a Support Vector Machine (SVM) involves finding a hyperplane that effectively separates data points belonging to different classes. The key steps in the functioning of an SVM are as follows:

1. **Input Data:**

SVM is a supervised learning algorithm, so it requires labeled training data. Each data point in the training set is associated with a class label.

1. **Feature Representation:**

Each data point is represented as a vector in a multi-dimensional space. The number of dimensions is equal to the number of features in the dataset.

1. **Choosing a Kernel Function:**

SVM can handle non-linear decision boundaries by using a kernel function. The choice of the kernel depends on the characteristics of the data. Common kernels include linear, polynomial, radial basis function (RBF), and sigmoid kernels.

1. **Mapping to a Higher-Dimensional Space (if needed):**

If a non-linear kernel is chosen, the input data is implicitly mapped to a higher-dimensional space. This is done to transform the data in a way that makes it easier to find a hyperplane in that space, even if the original data is not linearly separable.

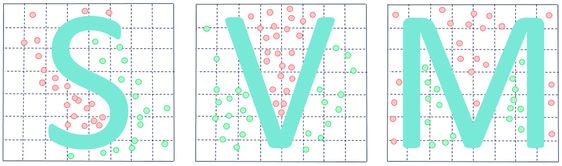
1. **Finding the Optimal Hyperplane:**

SVM aims to find the hyperplane that maximizes the margin between the support vectors of different classes. The margin is the distance between the hyperplane and the nearest data points of each class. Maximizing this margin helps improve the generalization ability of the model.

1. **Handling Non-Separable Data:**

In cases where the data is not perfectly separable by a hyperplane, SVM introduces a penalty term (C parameter) to allow for some misclassification. This parameter controls the trade-off between achieving a smooth decision boundary and correctly classifying training points.

1. **Training the SVM:**

The training process involves optimizing the parameters of the hyperplane to achieve the best separation of classes. This is typically done using optimization algorithms that minimize a cost function associated with misclassifications and the margin size.  
  


1. **Making Predictions:**

Once trained, the SVM can be used to classify new, unseen data points. The decision boundary (hyperplane) determined during training is applied to classify the input into one of the predefined classes.

In summary, SVM works by finding an optimal hyperplane that maximizes the margin between different classes in a transformed feature space, using a kernel function to handle non-linear relationships in the data. The objective is to achieve accurate classification while generalizing well to new, unseen data.

**8.3 Types of SVM System**   
Support Vector Machines (SVMs) can be categorized into different types based on the nature of the problem they are designed to solve. The two main types are:

1. **Classification SVM:**

This is the most common type of SVM and is primarily used for classification problems. The goal is to find a hyperplane that best separates the data points of different classes, maximizing the margin between them. The hyperplane is determined during the training phase, and once trained, the SVM can classify new data points into one of the predefined classes based on their position relative to the hyperplane.

1. **Regression SVM (Support Vector Regression - SVR):**

In addition to classification, SVM can be used for regression tasks. In regression SVM, the goal is to find a hyperplane that predicts the continuous output values. SVR aims to fit a hyperplane that captures the general trend of the data while allowing for some deviation or error. It is particularly useful when dealing with datasets where the relationship between input features and output values is not strictly linear.

These are the main types of SVMs, but it is worth noting that SVMs can also be adapted for multi-class classification, anomaly detection, and other tasks. Multi-class SVM, for example, can be achieved by training multiple binary classifiers and combining their results.

Additionally, SVMs can be classified based on the type of kernel used:

1. **Linear SVM:**
   * Uses a linear kernel and is suitable for problems where the data can be effectively separated by a straight line or hyperplane in the input feature space.
2. **Non-Linear SVM:**
   * Utilizes non-linear kernels (e.g., polynomial, radial basis function) to handle complex relationships and find non-linear decision boundaries in higher-dimensional feature spaces.

Each type of SVM has its strengths and is applicable to different types of problems. The choice of the SVM type and kernel depends on the characteristics of the data and the nature of the task.

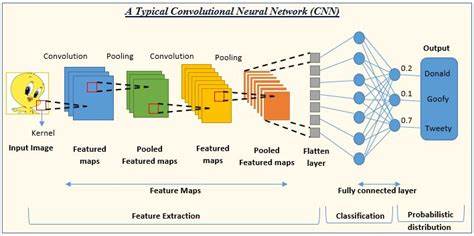
**8.4 Advantages of SVM**  
Support Vector Machines (SVMs) offer several advantages that contribute to their popularity and effectiveness in various machine learning applications. Some of the key advantages of SVMs include:

1. **Effective in High-Dimensional Spaces:**
   * SVMs perform well in high-dimensional spaces, making them suitable for tasks where the number of features is larger than the number of samples. This characteristic is particularly valuable in applications like text classification and image recognition.
2. **Robust to Overfitting:**
   * SVMs are less prone to overfitting, especially in high-dimensional spaces. The optimization objective of maximizing the margin helps create a more generalized model that tends to perform well on unseen data.
3. **Versatile Kernels for Non-Linearity:**
   * SVMs can handle non-linear decision boundaries effectively through the use of various kernel functions, such as polynomial, radial basis function (RBF), and sigmoid kernels. This versatility allows SVMs to capture complex relationships in the data.
4. **Global Optimization:**
   * SVMs involve a convex optimization problem, which means that the solution found is the global minimum rather than a local minimum. This property ensures that the trained model is optimal and not sensitive to the initial conditions of the optimization process.
5. **Memory Efficient:**
   * SVMs use a subset of training data points (support vectors) to define the decision boundary. This results in memory efficiency, especially when dealing with large datasets, as only a small fraction of the training data is used to make predictions.
6. **Effective in Small Sample Sizes:**
   * SVMs can perform well even with a small number of training samples, making them suitable for situations where the availability of labeled data is limited.
7. **Handle Imbalanced Data:**
   * SVMs can handle imbalanced datasets, where one class has significantly fewer samples than the other can. The class weights and the ability to adjust the cost parameter (C) contribute to their effectiveness in such scenarios.
8. **Theoretical Foundation:**
   * SVMs have a solid theoretical foundation in statistical learning theory. The margin concept and the VC (Vapnik-Chervonenkis) dimension provide a framework for understanding the generalization performance of the model.
9. **Works Well in Diverse Domains:**
   * SVMs have been successfully applied in various domains, including text classification, image recognition, bioinformatics, finance, and more. Their adaptability to different problem types makes them a versatile choice.

**8.5 Disadvantages of SVM**  
While Support Vector Machines (SVMs) offer several advantages, they also come with certain disadvantages and challenges. It's important to consider these aspects when deciding whether to use SVM for a particular machine learning task. Here are some of the disadvantages of SVM:

1. **Computational Complexity:**
   * SVMs can be computationally expensive, especially when dealing with large datasets. Training an SVM involves solving a quadratic optimization problem, and the time complexity can become prohibitive for very large datasets.
2. **Sensitivity to Parameter Tuning:**
   * The performance of an SVM is sensitive to the choice of hyper parameters, such as the regularization parameter (C) and the kernel parameters. Finding the right set of parameters through cross-validation can be time-consuming and requires careful tuning.
3. **Memory Usage:**
   * SVMs, particularly when using non-linear kernels, may require storing a significant amount of information about support vectors. This can lead to high memory usage, especially when dealing with large datasets.
4. **Interpretability:**
   * SVMs often provide accurate predictions, but the resulting models may be less interpretable compared to simpler models like decision trees. Understanding the influence of individual features on the decision boundary can be challenging.
5. **Limited Scalability:**
   * SVMs may not scale well to very large datasets, both in terms of computational time and memory requirements. Other machine learning algorithms, such as stochastic gradient descent-based methods, may be more suitable for massive datasets.
6. **Difficulty with Noisy Data:**
   * SVMs can be sensitive to noisy data and outliers. Outliers can have a significant impact on the position and orientation of the decision boundary, potentially leading to suboptimal models.
7. **Binary Classification:**
   * SVMs are inherently designed for binary classification. While there are techniques to extend them to multi-class problems, handling multiple classes requires training multiple binary classifiers, which may not be as straightforward as other algorithms designed for multi-class classification.
8. **Black Box Model:**
   * SVMs, especially with complex non-linear kernels, can be considered as black box models. Understanding the internal workings of the model and explaining its decisions to stakeholders or end-users may be challenging.
9. **Lack of Probabilistic Output:**
   * SVMs do not naturally provide probability estimates for predicted classes. While there are ways to obtain probability scores, they may not be as well-calibrated as those from algorithms explicitly designed for probability estimation, such as logistic regression.
10. **Choice of Kernel:**
    * Selecting the appropriate kernel for a specific problem can be challenging. The choice of kernel depends on the characteristics of the data, and there is no one-size-fits-all solution.

Despite these disadvantages, SVMs remain a powerful tool in various machine learning applications. Careful consideration of the specific requirements and characteristics of the problem at hand can help determine whether SVMs are a suitable choice or if other algorithms might be more appropriate.

**8.6 What is CNN**  
CNN stands for Convolutional Neural Network. It is a type of deep learning model commonly used in image and video recognition tasks. CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input images, which makes them highly effective for computer vision applications.  


Key Components of CNN:

Convolutional Layers:  
Filters/Kernels: Small matrices used to scan across the image to detect features like edges, textures, etc.

Stride: The number of pixels by which the filter moves across the image.

Padding: Adding borders to the input image to control the output size.

Activation Function:  
Commonly used activation function is ReLU (Rectified Linear Unit), which introduces non-linearity to the model.

Pooling Layers:

Max Pooling: Takes the maximum value from a patch of the feature map.

Average Pooling: Takes the average value from a patch of the feature map.

These layers help reduce the dimensionality of the feature map, making the computation more efficient.

Fully Connected Layers:

Neurons in these layers are fully connected to all activations in the previous layer.

These layers are usually at the end of the CNN and are used for making predictions.

Output Layer:

For classification tasks, this layer typically uses a softmax function to output probabilities for each class.

**8.7 How does CNN Work**  
Input Image: The model receives an input image, which is usually a fixed size.

Feature Extraction: Convolutional layers scan the image using filters to extract relevant features.

Downsampling: Pooling layers reduce the spatial dimensions, retaining the most important information.

Classification: Fully connected layers process the extracted features and output the final prediction.

**8.8 Advantage of CNN**  
Automatic Feature Extraction:

CNNs can automatically learn and extract features from raw image data without needing manual feature engineering, making them highly effective for complex image processing tasks.

Parameter Sharing:

Convolutional layers share weights across different parts of the image, reducing the number of parameters and computational load compared to fully connected networks.

Translation Invariance:

CNNs can recognize objects regardless of their position in the image. This property is due to the spatial hierarchy of features learned during the training process.

Hierarchical Feature Learning:

CNNs learn features in a hierarchical manner, starting from simple features like edges and corners in the early layers to more complex features in deeper layers, which enhances their ability to understand and represent images.

Effective Use of Data:

CNNs can achieve high performance with relatively less data compared to traditional machine learning algorithms, thanks to techniques like data augmentation and transfer learning.

High Performance in Image-Related Tasks:

CNNs have set new benchmarks in various image-related tasks, such as image classification, object detection, and segmentation, demonstrating superior performance over traditional methods.

**8.9 Disadvantage of CNN**

High Computational Cost:

Training CNNs, especially deep networks, requires significant computational resources, including powerful GPUs, large memory, and extensive storage.

Large Amount of Labeled Data Needed:

CNNs typically require a large amount of labeled data to achieve high accuracy, which can be a limitation in domains where labeled data is scarce or expensive to obtain.

Sensitivity to Input Variability:

CNNs can be sensitive to variations in the input data, such as changes in lighting, angle, and occlusion, which can affect their performance.

Complexity in Designing Architectures:

Designing an effective CNN architecture requires expertise and experimentation, as there are many hyperparameters to tune (e.g., number of layers, filter sizes, strides).

Lack of Interpretability:

The decision-making process of CNNs is often considered a "black box," making it difficult to understand and interpret why the network makes certain predictions.

Risk of Overfitting:

CNNs can easily overfit, especially when trained on small datasets. Regularization techniques such as dropout, data augmentation, and early stopping are often required to mitigate overfitting.

Vulnerability to Adversarial Attacks:  
CNNs can be susceptible to adversarial attacks, where slight, often imperceptible changes to the input image can lead to incorrect predictions, posing security concerns in critical applications.

Despite these disadvantages, CNNs have proven to be extremely powerful tools for a wide range of applications, particularly in computer vision, due to their ability to learn and generalize from large datasets.

CNNs can be susceptible to adversarial attacks, where slight, often imperceptible changes to the input image can lead to incorrect predictions, posing security concerns in critical applications.

Despite these disadvantages, CNNs have proven to be extremely powerful tools for a wide range of applications, particularly in computer vision, due to their ability to learn and generalize from large datasets.

**8.10 What is TensorFlow:**

TensorFlow is an open-source machine learning framework developed by the Google Brain team. It is widely used for building and deploying machine learning and deep learning models. TensorFlow provides a comprehensive ecosystem of tools, libraries, and community resources that support a range of tasks, from research to production-level deployment.  
  
  


**Key Features of TensorFlow:**

Scalability:

TensorFlow can scale from running on a single CPU to running on multiple GPUs and TPUs (Tensor Processing Units), making it suitable for both small and large-scale machine learning tasks.

Flexibility:

TensorFlow supports a wide range of machine learning and deep learning algorithms. It allows users to build custom models and train them using various optimization techniques.

Ecosystem:

TensorFlow includes a variety of components such as TensorFlow Lite for mobile and embedded devices, TensorFlow.js for machine learning in JavaScript, and TensorFlow Extended (TFX) for end-to-end machine learning pipelines.

Eager Execution:

TensorFlow offers an imperative programming environment called Eager Execution, which allows for immediate evaluation of operations and intuitive debugging.

TensorFlow Hub:

A repository of pre-trained models that can be easily reused and fine-tuned for specific tasks, helping to accelerate model development.

TensorFlow Serving:

A flexible, high-performance serving system for machine learning models designed for production environments.

Keras Integration:

TensorFlow integrates seamlessly with Keras, a high-level neural networks API, making it easier to build and train models with a user-friendly interface.

**Components of TensorFlow:**

Tensors:

The basic data structure in TensorFlow, representing multi-dimensional arrays of data.

Graphs:

TensorFlow uses dataflow graphs to represent computations. Each node in the graph represents a mathematical operation, while edges represent the tensors communicated between them.

Session:

A session in TensorFlow is used to execute the operations defined in the graph. It allocates resources (like GPU or CPU memory) and manages the state of the variables.

Variables:

Variables in TensorFlow are used to store model parameters that can be updated during training.

**Advantages of TensorFlow:**

Community and Support:

A large and active community provides extensive documentation, tutorials, and support through forums and GitHub.

Versatility:

TensorFlow supports various applications beyond neural networks, including statistical models and general numerical computation.

Performance:

Optimized for performance, TensorFlow can take full advantage of hardware acceleration provided by GPUs and TPUs.

Production-Ready:

TensorFlow is designed to be production-ready, offering tools and libraries to deploy models on different platforms and environments.

**Disadvantages of TensorFlow:**

Complexity:

TensorFlow can be complex for beginners due to its extensive features and flexibility.

Verbose Syntax:

TensorFlow code can be verbose and less intuitive compared to some other machine learning libraries, though this has improved with the integration of Keras.

Resource-Intensive:

Training large models in TensorFlow can be resource-intensive, requiring powerful hardware and substantial memory.

Overall, TensorFlow is a powerful and versatile tool that supports the entire machine learning workflow, from model building and training to deployment and serving.

**8.11 What is Flutter:**

Flutter is an open-source UI software development kit (SDK) created by Google. It is used to develop natively compiled applications for mobile (iOS and Android), web, and desktop from a single codebase. Flutter is known for its fast development cycles, expressive and flexible UI, and native performance.



Key Features of Flutter:

Single Codebase:

Write once, run anywhere. A single codebase can be used to develop applications for multiple platforms, including iOS, Android, web, and desktop.

Fast Development:

Hot Reload: Allows developers to see the changes in the code immediately in the running app without restarting the app, facilitating rapid iteration and development.

Expressive and Flexible UI:

Flutter provides a rich set of customizable widgets to create complex and visually appealing UIs. The widget-based architecture allows for a highly flexible and customizable user interface.

Native Performance:

Flutter’s rendering engine is optimized for performance, ensuring smooth animations and high-speed rendering. It compiles to native ARM code for both iOS and Android, which ensures high performance.

Dart Programming Language:

Flutter uses Dart, an object-oriented, class-defined programming language, which is easy to learn and offers features like async/await, garbage collection, and strong typing.

Extensive Libraries and Plugins:

Flutter has a vast ecosystem of packages and plugins that provide access to native device features, such as GPS, camera, and network, among others.

Open Source:

Flutter is open source, with a large and active community contributing to its development, providing extensive documentation, tutorials, and third-party libraries.

Components of Flutter:

Widgets:

The basic building blocks of a Flutter app. Everything in Flutter is a widget, including layout models, buttons, and even the app itself.

Flutter Engine:

The core of Flutter, written in C++, providing low-level rendering support via Skia (a 2D graphics library), handling Dart runtime, and implementing the Flutter framework’s core API.

Foundation Library:

A set of libraries written in Dart, providing basic classes and functions that are used to construct Flutter apps, including APIs for graphics, animation, and more.

Flutter Framework:

Provides a reactive framework, a rich set of pre-designed widgets, and tools for designing, building, and debugging applications.

Advantages of Flutter:

Cross-Platform Development:

Develop apps for multiple platforms from a single codebase, reducing development time and effort.

High Performance:

Compiles to native code, resulting in high-performance applications with smooth animations and fast startup times.

Rich Set of Widgets:

Offers a comprehensive set of widgets for building highly customizable UIs that adhere to the Material Design and Cupertino (iOS) design guidelines.

Rapid Development:

Features like hot reload, extensive libraries, and plugins enable rapid development and iteration.

Strong Community Support:

A large, active community and extensive documentation make it easier to find solutions and resources.

*Flutter is an open-source UI software development kit (SDK) created by Google, used to build natively compiled applications for mobile (iOS and Android), web, and desktop from a single codebase. It is known for its expressive and flexible UI, native performance, and fast development cycles.***8.12 What is Dart:**

Dart is an open-source, general-purpose programming language developed by Google. It is designed for building web, server, desktop, and mobile applications. Dart is known for its performance, ease of use, and the ability to create high-quality, production-ready code. It is the primary language used for developing applications with Flutter, Google's UI toolkit for building natively compiled applications.

Key Features of Dart:

Object-Oriented:

Dart is an object-oriented language, which means it supports concepts such as classes, objects, inheritance, polymorphism, and encapsulation.

Strongly Typed:

Dart is a statically typed language, which allows for type checking at compile time, reducing runtime errors and enhancing code quality and maintainability.

JIT and AOT Compilation:

Dart supports both Just-In-Time (JIT) compilation for faster development cycles with features like hot reload and Ahead-Of-Time (AOT) compilation for optimized, high-performance production code.

Asynchronous Programming:

Dart has built-in support for asynchronous programming using async and await, making it easier to write non-blocking code for applications that perform I/O operations, such as web servers or network requests.

Rich Standard Library:

Dart comes with a comprehensive standard library that provides a wide range of functionalities, including collections, date and time manipulation, file and network I/O, and more.

Sound Null Safety:

Dart includes sound null safety, which helps prevent null reference errors by distinguishing between nullable and non-nullable types at compile time.

Cross-Platform Development:

Dart can be used to build applications for various platforms, including web, mobile (iOS and Android), desktop (Windows, macOS, and Linux), and server environments.

Advantages of Dart:

Fast Development:

Dart’s JIT compilation and hot reload feature significantly speed up the development process, making it easy to see the effects of code changes instantly.

High Performance:

Dart’s AOT compilation produces optimized native code, resulting in high-performance applications.

Strong Tooling Support:

Dart has excellent tooling support, including the Dart Analyzer for static code analysis, the Dart DevTools for debugging, and integration with popular IDEs like IntelliJ IDEA, Android Studio, and Visual Studio Code.

Productivity:

Dart’s modern syntax, rich standard library, and robust type system enhance developer productivity and code quality.

Cross-Platform Consistency:

Using Dart with Flutter allows for consistent behavior and appearance across multiple platforms, reducing the effort needed to create platform-specific code.

**9.****Future Work**

ـــــــــــــــــــــــــــــــــــــــــــــــــ

Combine visual data with other modalities like audio (voice tone) or physiological signals (heart rate) for a more comprehensive understanding of emotions.

Multi-Person Detection and Recognition by Extend the app to detect and recognize emotions for multiple individuals within the frame simultaneously.

Refine the app to not only recognize basic emotions but also classify the intensity level (e.g., mild sadness vs. intense sadness).

**10.Conclusion**

ـــــــــــــــــــــــــــــــــــــــــــــــــ

In the end, we were able to create a model that works on facial expression recognition using Python, and for this, we used several libraries to help us build this model. For example(os- numpy- train \_ test \_ split- svm- accuracy score- preprocessing- io, color, transform)We used a dataset consisting of images divided into six category (Happy-Sad-Fair-Angry- Surprised- Natural)Using the algorithm (svm), after converting the images to grayscale, it gave us the accuracy for each of the category (Happy 1.0\_Sad 1.0\_Fear 0.88Angry 1.0Surprised 0.90\_Natural 0.50)And it gave us an overall accuracy percentage for the entire set of category (over all\_0.93)We tested the model by inputting various images, including personal ones, separate from the dataset images, to evaluate the accuracy of the model. From this, it identified which category each image belonged to.

**11****. References:**

ـــــــــــــــــــــــــــــــــــــــــــــــــ

<https://www.sciencedirect.com/science/article/pii/S1566253523003354>

<https://www.nature.com/articles/s41598-023-35446-4>

<https://mmifacedb.eu/>

<https://arxiv.org/abs/1509.05371>

<https://www.youtube.com/watch?v=95RhuC30R5U&list=PLtsZ69x5q-X9j44MdSX-NGuOhGXOY0aqH>  
  
  
<https://www.kaggle.com/>

<https://www.youtube.com/watch?v=iUGV8qBQtCc&list=PLH0em1f_fBoQCvAIJeWJtqSVoE3quQXue>

<https://pub.dev/>

<https://www.coursera.org/articles/machine-learning-algorithms>

<https://www.datacamp.com/blog/how-to-learn-machine-learning>

<https://link.springer.com/article/10.1007/s00500-023-08531-z>

<https://link.springer.com/article/10.1007/s11554-021-01107-w>

<https://github.com/topics/face-emotion-recognition>

<https://learnopencv.com/facial-emotion-recognition/>