



الجمهورية الجزائرية الديمقراطية الشعبية
The People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة محمد بوضياف بالمسيلة
University Mohamed Boudiaf of M'sila



قسم الإعلام الآلي
Department of Computer Science

كلية الرياضيات والإعلام الآلي
Faculty of Mathematics and Informatics

Domain: Mathematics and Computer Science

Thesis Presented to Fulfill the Partial Requirement
for **Master's Degree** in Computer Science

Specialty: Information System and Software Engineering ISSE

Prepared By: Douaa Hanan Kadi, Nada Erayhane Heltali

Supervised By:
Said Gadri

Entitled

Facial Emotion Recognition Using Deep Learning Approach

Jury Members

Allaoua Hemmak	President
Said Gadri	Supervisor
Maroune Kihal	Examiner

Academic Year 2024/2025

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Mr. Saïd Kadri, for his valuable guidance, constructive feedback, and continuous support throughout the development of this work. His scientific insight has greatly contributed to the successful completion of this thesis. I also extend my thanks to the examination committee for their time, interest, and thoughtful evaluation of this research.

My appreciation goes to the Faculty of Mathematics and Computer Science, and especially to its Dean, for his continuous support, for providing a conducive academic environment, and for facilitating all necessary administrative and academic resources.

Finally, I am thankful to everyone who, directly or indirectly, contributed to this work—through reference materials, technical assistance, or academic support.

الاهداء

(إِنَّ اللَّهَ وَمَلَائِكَتَهُ يُصَلُّونَ عَلَى النَّبِيِّ يَا أَيُّهَا الَّذِينَ آمَنُوا صَلُّوا عَلَيْهِ وَسَلِّمُوا تَسْلِيمًا)

(الأحزاب: 56)

إلى سيّدٍ وجبّ رسول الله الجهاد الشهيد المُنخَن في أعداء الله صلوات ربّي وسلامه عليه
إلى والديّ الحبيبين، اللّذين كانا لي نورًا يهدي الطريق، وسندًا لا ينكسر، بدعائهما وحبهما كان هذا العمل مكلًا بالنجاح،

• فلهم مني كلُّ الشكر والامتنان

إلى إخوتي وأخواتي، عائلتي التي هي سندي ودفني في هذه الحياة، فبفضلهم صيرت وكافحت ولم أكلّ
إلى صديقتي العزيزة ندى، التي كانت معي في كلِّ خطوة من هذا الطريق، وشريكتي في الأفراح والأحزان، ورافقتني
بدعمها وصدققتها الصادقة

إلى رانيا الصديقة الوفية ادام الله علينا الصداقة والمحبة

إلى نجاح ، التي غادرتنا إلى جنان الرحمة، رحمها الله وأسكنها فسيح جناته، وأبقى ذكراها حيّة في قلبي، فهي جزء لا

• يتجزأ من هذه الرحلة

إلى كلِّ من وقف بجاني، وسأهم بشكل مباشر أو غير مباشر في إنجاز هذا العمل، فلکم مني أسمی آیات الشکر والتقدير

قاضي دعاء حنان

(فَرِحِينَ بِمَا آتَاهُمُ اللَّهُ مِنْ فَضْلِهِ)

الحمد لله رب العالمين الذي بنعمته تتم الصالحات وبتوفيقه تتحقق الغايات والذي بفضلها انا اليوم اعلى سلم النجاح

وقد اصبح الحلم واقعا افتخر به، اما بعد: اهدي عملي المتواضع....

الى نفسي التي لم تخذلني يوما اهديك هذا الانجاز عرفانا بكل لحظة صبر مرت ...

الى من زين اسمي باجمل الالقاب الى من رباني وكافح من اجلي الى المصباح الذي انار دربي والدي الغالي طاب بك

العمر

ياسيد الرجال ،الى من جعل الله الجنة تحت اقدامها ، واحتضني قلبها قبل يديها، الى من سهلت لي الشدائد بدعائها الى

قوتي بعد الله والدي الحبيبة ،الى من قيل فيهم : "سنشد عضدك باخيك" الى ضلعي الثابت الذي لا يميل اخوتي مهدي

وسيف والى من وهبني الله نعمة وجودهما في حياتي الى العقد المتين اختي حبيبتي بثينة وبنيتها الكنكوتة روشان.

الى خطيبي وزوجي المستقبلي ورفيق دربي بطلو الايام ومرها دمت لي حتى المشيب، الى من كاتفوني نحو طريق العلم

والنجاح وكانو عوننا لي صديقتاي الوفيتين رانيا ودعاء وفقهم الله والى صديقتي المرحومة باذن الله عمرو نجاج.

الى خالتي القريبة على قلبي وابنتها مريم اسال الله ان يلحقها المراتب العلى، الى جدتي حفظها الله ورعاها، الى جميع

الاهل والاحية بدون استثناء، الى كل طالب علم سعى بعلمه ليفيد الاسلام والمسلمين بكل ما اعطاه الله من علم ومعرفة،

الى كل ما يبهجهم نجاحي شكرا لكم

فلولا توفيق الله ثم انتم ما كنت لاصل الى ما انا عليه اليوم فالحمد لله حمدا كثيرا طيبا مبارك فيه والصلاة والسلام على

خاتم الانبياء والمرسلين محمد ﷺ (وَآخِرُ دَعْوَاهُمْ أَنِ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ).

هلتالي ندى الريحان

Table of Content

Acknowledgements.....	ii
Dedication.....	iii-iv
General Introduction.....	1-2
Chapter 1: Generalities on Facial Emotion Recognition.....	3
1 Introduction.....	4
2 Facial recognition.....	4
3 Image and Image processing.....	4
3.1 <i>Low-Level Processing</i>	5
3.2 <i>Medium Level Processing</i>	5
3.3 <i>High Level Processing</i>	5
4 Importance of Image Processing.....	5
5 Types of Images.....	6
5.1 <i>Gray Scale Images</i>	6
5.2 <i>RGB Images</i>	6
6 Emotion Recognition.....	7
7 Techniques for Emotion Recognition.....	7
7.1 <i>Facial Expression Recognition</i>	7
7.2 <i>Speech-Based Emotion Recognition</i>	7
7.3 <i>Text-Based Emotion Recognition</i>	8
7.4 <i>Physiological Signal-Based Emotion Recognition</i>	8
7.5 <i>Body Language-Based Emotion Recognition</i>	8
8 Facial Expression Recognition.....	9
9 Applications of Facial Emotion Recognition.....	9
9.1 <i>Provision of Personalized Services</i>	10
9.2 <i>Customer Behavior Analysis & Advertising</i>	10
9.3 <i>Healthcare</i>	10
9.4 <i>Employment</i>	10
9.5 <i>Crime Detection & Fraud Prevention</i>	10
10 Recent Trends in Facial Emotion Recognition (FER).....	11
11 Challenges in Facial Emotion Recognition (FER):.....	11

12 Conclusion.....	13
Chapter 2: Machine Learning Algorithms and Model Using.....	14
1 Introduction.....	15
2 Machine Learning.....	15
2.1 Definition.....	15
2.2 Machine Learning techniques.....	15
3 Deep Learning.....	19
3.1 Definition.....	19
3.2 Types of deep neural networks.....	20
4 Convolutional Neural Networks(CNNs).....	22
5 Working Principle.....	23
5.1 Convolution Layer.....	23
5.2 Activation Function (typically ReLU).....	23
5.3 Pooling Layer.....	24
5.4 Fully Connected Layers.....	25
5.5 The loss function (LOSS).....	25
6 Data processing techniques.....	25
6.1 Data Standardization Techniques.....	26
6.2 Data Augmentation.....	27
6.3 Optimizers and Learning Rate.....	28
6.4 Learning Rate.....	29
6.5 Weight initialization and pretrained networks.....	29
6.6 Regularization.....	30
7 CNN Facial Recognition Apps.....	30
7.1 How facial feelings can be recognized.....	30
7.2 CNN Facial Recognition Apps.....	32
8 Conclusion.....	33
Chapter 3: Model Development and Performance Evaluation.....	34
1 Introduction.....	35
2 Tools and Work Environment.....	35
2.1 Jupyter Notebook.....	35
2.2 Google Colab.....	36
2.3 Visual Studio Code Environment.....	36

2.3.1 Key Features	36
2.3.2 Reason for Selection	36
2.4 Python Tools, Libraries and Frameworks	36
2.4.1 Web FrameWork	37
2.4.2 Libraries and Tools Used	37
3 Dataset Description	38
3.1 Data Augmentation	38
3.2 Data Loading and Label Assignment	40
3.3 Image Resizing	40
3.4 Grayscale Conversion	40
3.5 Normalization of Pixel Values	41
3.6 Reshaping Input Data	41
3.7 Data Splitting	42
3.7.1 Train set	42
3.7.2 Test set	42
3.7.3 Importance of Data Splitting	43
4 Model Design (CNN)	43
4.1 Convolutional Layers	44
4.2 Pooling Layers	44
4.3 Dropout Layers	44
4.4 Flatten Layer	45
4.5 Dense Layers	45
4.6 Purpose of This Design	45
5 Training Parameters	46
6 Model Evaluation Strategy	47
7 Prediction Mechanism	47
7.1 Prediction on Test Data	47
7.2 Prediction on New External Data	48
8 Results Analysis	48
8.1 Training and Validation Accuracy and Loss Curves	48
8.1.1 Accuracy Curve	49
8.1.2 Loss Curve	49
8.2 Confusion Matrix Analysis	49
8.3 Classification Report	50

9 Conclusion	51
Chapter 4: Application Implementation and User Interface Description	52
1 Introduction.....	53
2 Application Description.....	53
3 Application Objective.....	53
4 Application Features.....	54
5 Application Interface.....	54
5.1 Homepage	54
5.2 About Section.....	55
5.3 Contact Section	56
5.4 Emotion Detection.....	56
5.5 Model Evaluation	59
5.6 Train a New Model.....	61
5.6.1 <i>Train on RGB/Grayscale Data Interfaces</i>	61
5.7 Clarification of the Unified Training and Evaluation Process for Models	66
6 Conclusion	67
General Conclusion	68
7 Bibliographie.....	51

List of figures:

Figure 1:Types of Images processing	6
Figure 2:Machine Learning Techniques	15
Figure 3:Logistic Regression	16
Figure 4:SVMs Algorithm	17
Figure 5:Naive Bayes Classifier	17
Figure 6:KNN Algorithm.....	18
Figure 7:Deep Neural Network	19
Figure 8:Deep Neural Network(DNN)	20
Figure 9:A Long short-term memory network,LSTM,long short-term memory.....	21
Figure 10:Convolutional Layer	23
Figure 11:Activation Function	24
Figure 12:Pooling Layer	24
Figure 13:FC Layer	25
Figure 14:Identifying facial features.....	31
Figure 15:Facial Expression Regression Workflow	31
Figure 16:CK+ Dataset	38
Figure 17:Types of Data Augmentatio	38
Figure 18:Before and After Data Augmentation and Balancing	39
Figure 19:Load & Resize data.....	40
Figure 20:Gray, Norm and Reshape data.....	42
Figure 21:The train_test_split Function.....	43
Figure 22:CNN Model Architecture	43
Figure 23:The layered architecture of the proposed CNN model.....	45
Figure 24:CNN for Emotion Classification	46
Figure 25:Model Accuracy and Loss Curves	49
Figure 26:Confusion Matrix Result.....	Error! Bookmark not defined.
Figure 27:Classification Report.....	51
Figure 28:System Architecture Overview.....	53
Figure 29:Main Homepage Interface of FaceFeel Application	55
Figure 30>About Section Interface of FaceFeel Application.....	55
Figure 31 : Contact via Footer.....	56
Figure 32:Input Method Selection “By Image” or “By Camera”.....	56
Figure 33:Drag-and-Drop Upload Interface for Image Selection.....	57
Figure 34: Notification Displayed During Prediction.....	57
Figure 35: Emotion Prediction Results	58
Figure 36: Real-time Emotion Detection via Webcam.....	58
Figure 37 : Another Example for Real-time Emotion Detection.....	59
Figure 38 : Notification When No Face Is Found	59
Figure 39 : Model Evaluation Interface of FaceFeel Application	60
Figure 40: Model Accuracy Recommendations	60
Figure 41: Model Training Progress with warning msg	62
Figure 42: Training Results in Grayscale Model.....	63
Figure 43: Downloading the Pre-trained Model.....	63
Figure 44 : Testing Grayscale Model Interface	64
Figure 45: Model Being Tested’ message window	64
Figure 46 : Model Testing Results in Grayscale Model	65
Figure 47 : Emotion Prediction Grayscale Interface.....	66
Figure 48: Emotion Prediction Result Grayscale Interface.....	66

Figure 49 :Example of RGB Prediction in the RGB Prediction Interface67

List of abbreviations:

- CNN: Convolutional Neural Network
- FER: Facial Expression Recognition
- MLP: Multi-Layer Perceptron
- GPU: Graphics Processing Unit
- API: Application Programming Interface
- JSON: JavaScript Object Notation

General Introduction

Artificial Intelligence (AI) stands as one of the most transformative technologies of the 21st century, enabling machines to perform tasks traditionally associated with human intelligence, including reasoning, learning, and problem-solving. Within the broad spectrum of AI applications, Facial Emotion Recognition (FER) emerges as a particularly impactful domain. It aims to equip machines with the ability to detect and interpret human emotions based on visual facial cues.

FER systems heavily depend on computer vision, image processing, and deep learning methodologies. Convolutional Neural Networks (CNNs), in particular, have become fundamental in enabling systems to extract emotional information from facial images with high levels of accuracy and efficiency.

Despite significant advancements, facial emotion recognition continues to face several challenges. Variations in individual facial expressions, lighting conditions, head orientation, occlusions (such as glasses or masks), and the subjective nature of emotions complicate the development of robust and generalizable models. Additionally, deploying these systems in real-time scenarios, especially on resource-constrained devices like smartphones, introduces further technical difficulties.

Accordingly, the primary research question addressed in this work is: **How can a CNN-based facial emotion recognition system be designed and implemented to achieve high accuracy and real-time performance, while also managing practical issues such as image variability, differences in facial features, and limited computational power?**

This thesis presents the development of a facial emotion recognition system based on CNN architecture. The structure of the thesis is as follows:

Chapter 1: Generalities on Facial Emotion Recognition.

Chapter 2: Machine Learning Algorithms and Model Using.

Chapter 3: Model Development and Performance Evaluation.

Chapter 4: Application Implementation and User Interface Description

Finally, the thesis ends with a general conclusion summarizing the work and suggesting future research directions

Chapter 1

Generalities on Facial Emotion Recognition

1 Introduction

With the advent of modern technology our desires went high and it binds no bounds. In the present era a huge research work is going on in the field of digital image and image processing. The way of progression has been exponential and it is ever increasing. Image Processing is a vast area of research in present day world and its applications are very widespread. The area of signal processing known as image processing deals with input and output signals that are both pictures. Facial expression recognition is one of the most significant uses of image processing. The looks on our faces convey our feelings. Expressions on the face are crucial in communication amongst people. A nonverbal scientific gesture that conveys our feelings through our faces is called facial expression. The generation needs automatic facial expression detection since it is crucial to robotics and artificial intelligence.

2 Facial recognition

The process of utilizing a person's face to identify or verify their identification is known as facial recognition. People may be recognized in real time or in images and videos using facial recognition software.

One type of biometric security is facial recognition. Voice recognition, fingerprint recognition, and retina or iris recognition are further types of biometric software. Although there is growing interest in other applications, the technology is mostly utilised for security and law enforcement.

[1]

3 Image and Image processing

image refers specifically to a binary representation of visual data. These digital pictures can consist of individual video frames, graphics, or photographs. For the purposes of this discussion, an image is defined as any picture that has been generated, copied, and stored in an electronic format .

Image processing refers to a set of techniques applied to digital images in order to enhance their quality, extract meaningful information, or prepare them for further analysis or visualization. Image processing operations are generally categorized into three levels based on the complexity and purpose of the tasks [2].

3.1 Low-Level Processing

Low level processing involves basic operations such as image preprocessing, image enhancement, image restoration, image sharpening [3]. The main characteristic of low level processing is that both its inputs and outputs are images. Low-level processing involves fundamental operations that directly manipulate pixel values without extracting semantic information. Typical examples include:

- Preprocessing to remove noise
- Contrast enhancement,
- Image sharpening

3.2 Medium Level Processing

Mid level processing involves tasks like image classification, object identification, image segmentation, etc. The main characteristic of medium level processing is that its inputs are generally images whereas its outputs are attributes associated with image which are extracted from it.

Mid-level processing focuses on the extraction of meaningful features or structures from the image [2]. This includes:

- Segmentation
- Edge detection
- Object extraction

3.3 High Level Processing

High level processing involves making sense of ensemble of recognised object and cognitive tasks associated with computer vision. High-level processing aims at interpreting the contents of the image to obtain a deeper understanding [2]. It typically involves:

- Image analysis
- Scene interpretation

4 Importance of Image Processing

Since digital images are essentially numerical data and not inherently visible to the human

eye, it becomes necessary to process and prepare them for visualization on output devices such as monitors and laser printers. Moreover, image processing enhances the appearance and clarity of structures within the image, optimizing it for specific applications. There are three major types of transformations used in image processing :

Image-to-Image Transformation: Modification of an image to improve its quality or to produce another image.

Image-to-Information Transformation: Extraction of relevant information from an image for analysis.

Information-to-Image Transformation: Visualization of data or information in image form. [4]

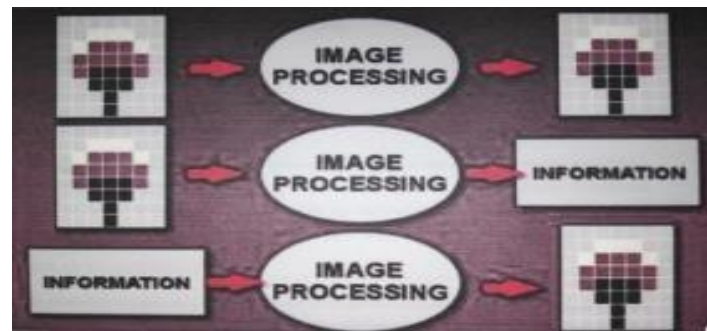


Figure 1:Types of Images processing

5 Types of Images

5.1 Gray Scale Images

A gray scale picture is a capacity $I(x, y)$ of the two spatial directions of the picture plane. $I(x, y)$ is the force of the picture force of picture at the point (x, y) on the picture plane. $I(x, y)$ take non-negative expect the picture is limited by a rectangle [5].

5.2 RGB Images

It can be spoken to by three capacities, $R(x, y)$ for red, $G(x, y)$ for green and $B(x, y)$ for blue. A picture might be persistent as for the x and y facilitates and furthermore in adequacy. Changing over such a picture to advanced shape requires that the directions and the adequacy to be digitized. [5].

6 Emotion Recognition

Emotion detection, also known as facial emotion recognition, is a fascinating field within the realm of artificial intelligence and computer vision. It involves the identification and interpretation of human emotions from facial expressions. Accurate emotion detection has numerous practical applications, including human-computer interaction, customer feedback analysis, and mental health monitoring. Convolutional Neural Networks (CNNs) have emerged as a powerful tool in this domain, revolutionizing the way we understand and process emotional cues from images [6].

7 Techniques for Emotion Recognition

Emotion recognition technologies are used to detect and analyze human emotions through various signals such as facial expressions, voice tone, body language, and even physiological patterns. These technologies rely on artificial intelligence, deep learning, and signal processing. Below are the key techniques used:

7.1 Facial Expression Recognition

Facial emotion recognition is a fast evolving field with the center focus of computer vision, machine learning, and affective computing. Due to its numerous applications in fields like marketing, human-robot interaction, virtual reality, and healthcare, among others, emotion recognition from facial expressions has drawn a lot of interest [7].

7.2 Speech-Based Emotion Recognition

Speech Emotion Recognition, abbreviated as SER, is the act of attempting to recognize human emotion and affective states from speech. This is capitalizing on the fact that voice often reflects underlying emotion through tone and pitch. This is also the phenomenon that animals like dogs and horses employ to be able to understand human emotion. Emotion recognition is the part of speech recognition which is gaining more popularity and need for it increases enormously. Although there are methods to recognize emotion using machine learning techniques, this project attempts to use deep learning to recognize the emotions from data. SER(Speech Emotion Recognition) is used in call center for classifying calls according to emotions .

the performance parameter for conversational analysis thus identifying the unsatisfied

customer, customer satisfaction and so on.. for helping companies improving their services. It can also be used in-car board system based on information of the mental state of the driver can be provided to the system to initiate his/her safety preventing accidents to happen [8].

7.3 Text-Based Emotion Recognition

Text-Based Emotion Recognition (TBER) is the process of identifying human emotions from written text using Natural Language Processing (NLP) and machine learning techniques. Since emotions are often conveyed through words, sentence structures, and contextual cues, TBER analyzes text to detect underlying sentiments and affective states [9].

7.4 Physiological Signal-Based Emotion Recognition

Physiological Signal-Based Emotion Recognition is a field that aims to determine a person's emotional state by analyzing the physiological signals produced by the body when experiencing different emotions [10].

This technique relies on measuring and interpreting biometric data such as:

Heart Rate (HR): The heart rate changes with emotions, such as an increased pulse during fear or stress.

Galvanic Skin Response (GSR): Measures the skin's electrical response, which varies due to sweating caused by emotions like anxiety and fear.

Electroencephalography (EEG): Records brain activity and can distinguish different emotions, such as happiness or sadness, by analyzing brain waves.

Electrocardiography (ECG): Used to monitor heart activity changes that reflect emotional influence.

Electromyography (EMG): Measures electrical activity in muscles, helping to detect emotions like tension or anger.

Respiration Rate: The breathing rate changes with different emotions, accelerating during stress or slowing down during relaxation.

7.5 Body Language-Based Emotion Recognition

Body language-based emotion recognition is the process of identifying human emotions through the analysis of body movements, postures, and gestures. Unlike facial or voice-based recognition, this method focuses on non-verbal cues such as hand movements, body posture,

and repetitive actions to infer emotions [11].

This technology has various applications, including :

Healthcare: Detecting early signs of mental health issues like depression or anxiety.

Human-Computer Interaction: Enhancing AI systems to respond to user emotions

Surveillance & Security: Identifying stress or nervous behavior in public spaces

Gaming & Virtual Reality: Creating adaptive and immersive experiences.

8 Facial Expression Recognition

Facial Emotion Recognition (FER) is the process of analyzing and extracting emotions from facial expressions using various techniques. It relies on the Facial Action Coding System (FACS), which describes facial muscle movements through Action Units (AUs) [12].

The FER system follows several stages:

- **Data Collection:** Capturing facial expressions using static images or videos, which help represent the temporal characteristics of emotions.
- **Preprocessing :** Enhancing image quality by removing noise, adjusting contrast and brightness, solving illumination issues (e.g., histogram equalization), and detecting faces using algorithms like Viola-Jones.
- **Feature Extraction :** Techniques such as Gabor-wavelets and Principal Component Analysis (PCA) are used to extract significant features from images.
- **Classification :** Emotions are categorized based on facial actions or predefined classifications like Ekman's six basic emotions (Anger, Disgust, Fear, Happiness, Sadness, Surprise).

9 Applications of Facial Emotion Recognition

Facial Emotion Recognition (FER) has a wide range of applications across various fields, leveraging artificial intelligence and facial analysis to interpret human emotions. This technology is used to enhance user experiences, improve healthcare diagnostics, optimize workplace productivity, and even support crime detection. Despite its benefits, FER must be implemented responsibly to ensure ethical use and data privacy [13]. Below are the key application areas of FER:

9.1 Provision of Personalized Services

- Analyzing emotions to display personalized messages in smart environments.
- Providing personalized recommendations (e.g., music selection, cultural content).
- Predicting individual reactions to movies based on facial expressions.

9.2 Customer Behavior Analysis & Advertising

- Analyzing customers' emotions while shopping, focusing on products or store layout.
- Using facial recognition and tracking systems for advertising (e.g., digital signage in railway stations).

9.3 Healthcare

- Detecting autism or neurodegenerative diseases.
- Predicting psychotic disorders or depression to identify individuals in need of support.
- Suicide prevention.
- Detecting depression in elderly people.
- Monitoring patients' conditions during treatment.

9.4 Employment

- Assisting employers in hiring decisions.
- Identifying uninterested candidates during job interviews.
- Monitoring employees' moods and attention levels.
- Analyzing crime scene footage to determine potential motives.

9.5 Crime Detection & Fraud Prevention

- Detecting and reducing fraudulent insurance claims.
- Implementing fraud prevention strategies.

Identifying shoplifters

10 Recent Trends in Facial Emotion Recognition (FER)

Recent developments in Facial Emotion Recognition (FER) have focused on improving both the accuracy and efficiency of recognition systems. Several trends have emerged in this field:

- ***Deep Learning Approaches:***Leveraging models like CNNs, ResNet, and EfficientNet to achieve higher accuracy in emotion detection.
- ***Lightweight Models for Real-Time Applications:***Developing compact models that can run efficiently on mobile and embedded devices for real-time emotion recognition.
- ***Multimodal Emotion Recognition:***Combining multiple data sources such as facial expressions, speech, and textual data to improve recognition accuracy.
- ***Transfer Learning:***Utilizing pre-trained models on related tasks (e.g., face recognition) and fine-tuning them for emotion recognition to save time and resources.
- ***Synthetic Data Augmentation:***Generating artificial facial expression images to overcome the lack of diverse training data and enhance generalization.
- ***Real-Time FER Systems:***Designing algorithms capable of detecting emotions quickly and accurately for real-time applications like smart surveillance.
- ***Explainable FER Models:***Incorporating explainable AI techniques to provide transparency and understanding of how and why specific emotional predictions are made, especially in sensitive domains
- ***Micro-Expression Analysis:***Recognizing subtle and brief facial expressions that reveal true emotions, often hidden in conscious behavior

11 Challenges in Facial Emotion Recognition (FER):

Despite the significant progress achieved in the field of Facial Emotion Recognition (FER), several challenges continue to hinder the development of robust and reliable systems. These challenges include:

- ***Variations in Lighting, Pose, and Facial Features:*** Different lighting conditions, head orientations, and individual facial structures can significantly affect the accuracy of emotion recognition.
- ***Similarity Between Some Emotions:*** Emotions like fear and surprise often share similar facial expressions, making it difficult for models to distinguish between them accurately.
- ***Need for Large and Balanced Datasets:*** Deep learning models require vast and well-balanced datasets to learn effectively. Imbalanced data can cause bias toward certain emotions.
- ***Grayscale vs. Color Images:*** Models trained on grayscale images may perform poorly on colored images and vice versa, requiring the development of specific models for each type.
- ***Lack of GPU Resources:*** Limited access to powerful computing resources (e.g., GPUs) can slow down training and experimentation, especially with large models.
- ***Generalization to Real-World Scenarios:*** Many models perform well on lab datasets but fail in real-life situations due to differences in environment, camera quality, and spontaneous expressions.
- ***Cultural and Individual Differences:*** People from different cultures or individuals may express emotions differently, requiring more diverse data and robust models.
- ***Privacy and Ethical Concerns:*** Using facial data raises privacy issues, especially in surveillance and public applications, demanding careful handling and ethical considerations.

12 Conclusion

Facial emotion recognition is considered a fundamental aspect of emotion detection, with other modalities such as audio, text, psychological factors, and body posture also contributing to improved performance. Although significant progress has been made in this field, further development is still required to achieve a comprehensive and effective system. The methods used for feature extraction and classification vary, with some relying on individual techniques while others combine multiple approaches to take advantage of their respective strengths. There is no unified methodology in this field, but current research trends are leaning toward the use of deep learning techniques, especially Convolutional Neural Networks (CNN), which have shown promising results.

Chapter 2

Machine and Deep Learning : Algorithms and Models Used

1 Introduction

Deep learning is an advanced branch of machine learning that relies on the use of artificial neural networks to process data and extract complex patterns. Among the most prominent algorithms used in this field are Recurrent Neural Networks (RNN), Long Short-Term Memory Networks (LSTM), and Generative Adversarial Networks (GAN). Each of these algorithms is used depending on the nature of the data and the task at hand. Additionally, there is also the Convolutional Neural Networks (CNN) algorithm, which is used in various other applications such as image, text, and speech processing, depending on the type of data and the specific task.

2 Machine Learning

2.1 Definition

Machine learning is a subfield of computer science, which originates from the study of pattern recognition and computational learning theory in artificial intelligence. Machine learning methods may be referred to as predictive analytics on data. " A field of study that gives computers the ability to learn without being explicitly programmed [14].

2.2 Machine Learning techniques

There are four machine learning techniques described in.

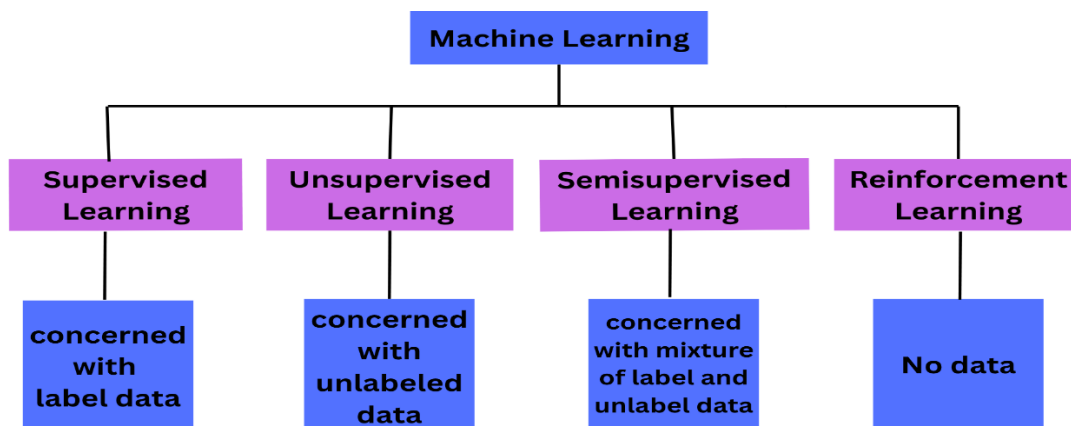


Figure 2: Machine Learning Techniques

2.2.1 Supervised learning

The supervised learning is a classic machine learning type. The training data consists of tuples (x_i, y_i) , where x_i is the input and y_i the corresponding target class. Meaning, the case in which the target value y_i is discrete (known, labeled). The goal of supervised learning is to build an artificial system that can learn the mapping between input and output and predict the output of the system given new inputs. Supervised learning tasks are divided into two categories: classification and regression [15].

2.2.2 Supervised Learning Algorithms

Decision trees: It is a supervised algorithm for classification. It can be used for regression in some cases. It classifies the input data by sorting the input data based on attributes or feature values. The decision tree model is composed of nodes, each node represents the function in the instance we want to classify, and each branch represents the value of the node. Instances are classified from the root node and sorted according to their characteristic values. The decision rule is the path from the root node to the last node in the tree [15].

Logistic regression: Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given data set of independent variables. This type of statistical model (also known as *logit model*) is often used for classification and predictive analytics. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability

success divided by the probability of failure [16].

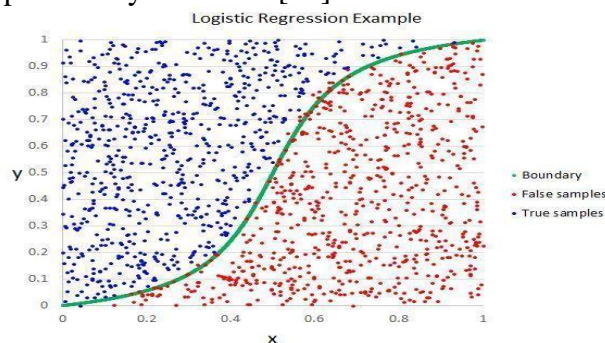


Figure 3: Logistic Regression

This is also commonly known as the log odds, or the natural logarithm of odds.

Support Vector Machines: It is one of the most powerful and popular machine learning methods, which can be used in either classifications and regression tasks, and is usually used for classification problems. SVM can process both linear separable and non-separable data. In the second case, SVM uses a kernel to transfer data points to a higher-dimensional space that can be linearly separated [15].

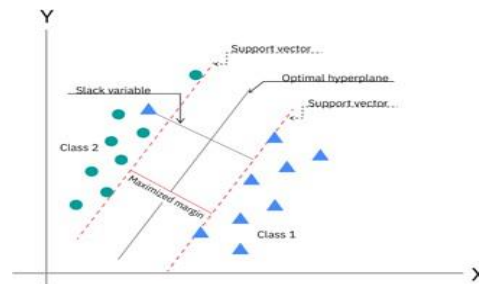


Figure 4: SVMs Algorithm

Naïve Bayes: Naive Bayesian classification is a type of simple probabilistic classification. A naive Bayes classifier is an algorithm that uses theorem of Bayes with a strong independence (called naïve) of the hypotheses. These classifiers are widely used for machine learning because they are simple to implement [17].

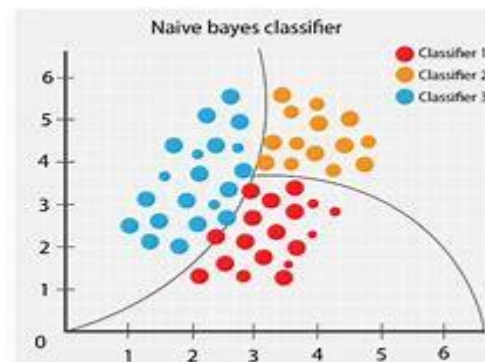


Figure 5: Naive Bayes Classifier

KNN algorithm: The k-nearest neighbors (KNN) algorithm is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. It is one of the popular and simplest classification and regression classifiers used

in machine learning . While the KNN algorithm can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another [18].

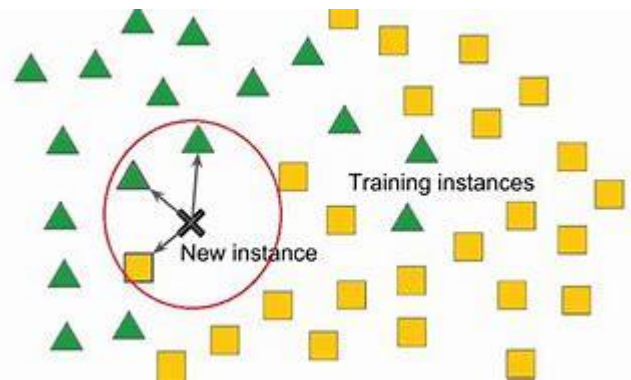


Figure 6:KNN Algorithm

2.2.3 Unsupervised learning

One important type of machine learning when the data is unlabeled and there is no priori identified classes, then it aims to find in the collection of data if there are clusters, groups or classes to be discovered. usually members of same cluster or group are similar to each other and dissimilar to member of other groups. The process is simple, we try to maximize the distance between the clusters and minimize the distance between the members of each cluster. Unsupervised learning problems can be further grouped into clustering and association problems [15].

2.2.4 Semisupervised learning

Semi-supervised learning is a machine learning approach that combines a small amount of labeled data with a large amount of unlabeled data during training. This method aims to improve model performance while reducing the need for extensive labeled datasets by leveraging the underlying structure of the unlabeled data. It is particularly useful when labeling data is expensive, time-consuming, or requires expert knowledge.

2.2.5 Reinforcement learning

Reinforcement Learning is a type of Machine Learning which allows machines to automatically determine the ideal behavior within a specific context, in order to maximize its performance. The reinforcement learning method aims at using observations gathered from the interaction with the environment to take actions that would maximize the reward or minimize the risk. [15].

3 Deep Learning

3.1 Definition

Deep learning, as a new field of machine learning research, is a process that enables computers to learn to perform tasks inherent in the brain, such as image recognition. Currently, deep learning (DL) methods have had a profound impact on computer vision and image analysis applications, such as image classification, segmentation, and image completion. Deep learning focuses on a specific category of machine learning, called artificial neural networks, which is inspired by the functions of the human brain. Modern deep learning provides a very powerful framework for supervised learning. By adding more layers and more units to the layers, deep networks can represent increasingly complex functions.

Motivation for Deep Learning :

- Datasets are widely available.
- The development of graphics processing units (GPUs)
- Hardware costs are reasonably priced, and processing methods and procedures have evolved.

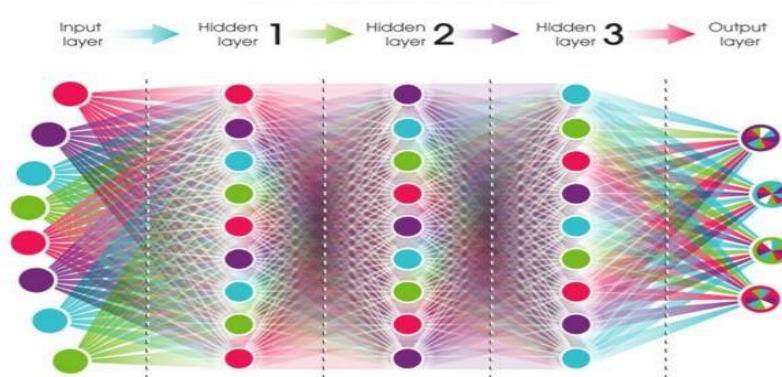


Figure 7: Deep Neural Network

3.2 Types of deep neural networks

3.2.1 A straightforward Deep Neural Network (DNN)

A neural network having more than two layers (one input layer ,many hidden layers ,etc.

Is referred to as a Deep Neural Network , a single output layer).

These neural process data through complex mathematical modelling . These models modify outputs in response to inputs [20].

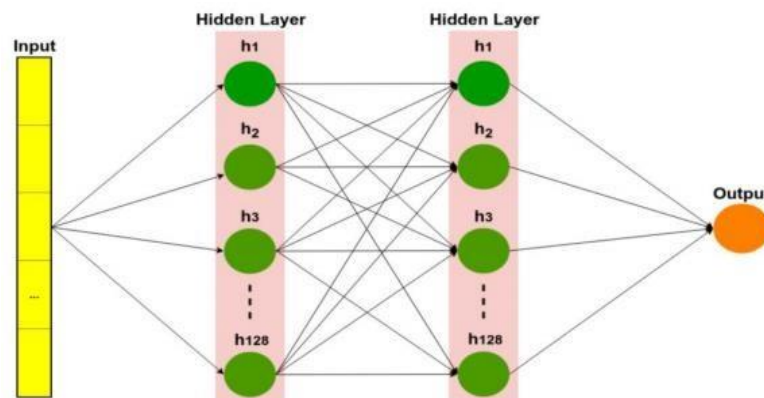


Figure 8:Deep Neural Network(DNN)

3.2.2 Convolutional Neural Networks CNN

A Neural Network with Convolutions CNN is a unique feed-forward neural network architecture that finds use in a variety of domains, including natural language processing (NLP), recommender systems (RS), and computer vision (CV). It's a architecture of a deep neural network. Convolutional layers, sampling layers (also known as pooling or subsampling layers), and a fully-connected layer are the many types of layers that make up a CNN network. Convolution layers filter the input data to extract the most pertinent elements, and sampling layers, also known as pooling layers, lower the resolution (the dimension) of the elements chosen by the convolution layers. This lowers complexity, prevents overfitting, and boosts the CNN network's resilience and performance [20].

3.2.3 Recurrent Neural Networks (RNN)

Recurrent neural networks are a type of neural network in which feedback loops are produced by the directed cycle formed by the connections between neurones. An RNN's primary purpose is to process of sequential to collect. RNNs, in contrast to ordinary neural networks, are able to remember past informational computations and apply them to the subsequent input element. The so-called LSTM (Long Short-Term Memory) network is a unique kind of RNN that may use long memory as an activation function in hidden layers. Hochreiter and Schmidhuber (1997) proposed the LSTM architecture [21].

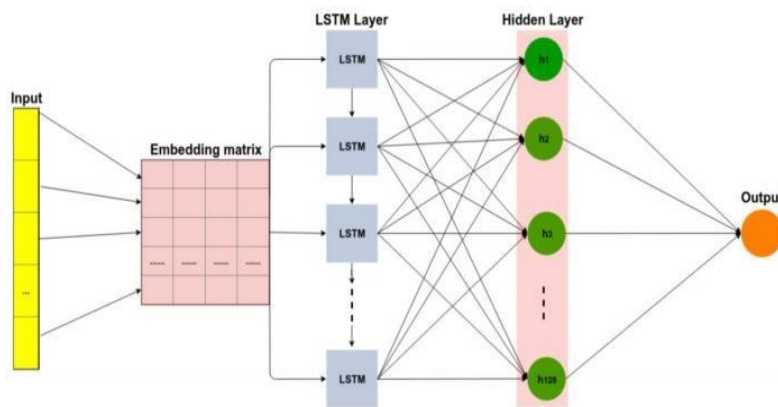


Figure 9: A Long short-term memory network, LSTM, long short-term memory

3.2.4 Recursive Neural Networks (Recursive Neural Network RecNN)

is a type of neural network that may be thought of as an extension of RNN networks. RecNNs are commonly employed for data-driven directed acyclic graph structure learning. The left and right hidden state vector

The hidden state vector of the current node may be computed using the graph's nodes [22].

3.2.5 Deep Belief Networks (DBN)

Several layers of graphical models with both directed and undirected edges make up a DBN network. A group of neurones makes up each of the several layers that make up each network. not linked to one another, but rather to neurones in the subsequent layer. A particular approach known as the "greedy layer-wise learning algorithm" is used to learn a DBN network [23].

3.2.6 Hybrid Deep Learning Network (HDLN)

Another type of neural network is known as a "Hybrid Deep Learning Network," which blends two or more deep models, including RBMs, CNNs, LSTMs, and probabilistic PNNs. model (RBM) [24].

data derived from the internal.

4 Convolutional Neural Networks(CNNs)

Convolutional Neural Networks (CNNs) were employed as the primary architecture for feature extraction and emotion classification. The input to the model consisted of grayscale facial images resized to 48×48 pixels, which is a common preprocessing step in FER to reduce computational complexity while preserving relevant expression features. The CNN architecture implemented in this project consisted of multiple convolutional layers followed by ReLU activation functions and max-pooling layers. These layers enabled the model to learn hierarchical spatial features—starting from low-level edges and corners in early layers to complex emotion-related patterns in deeper layers.

Batch normalization was applied after some convolutional layers to stabilize training and accelerate convergence. Dropout layers were also introduced between fully connected layers to reduce overfitting and enhance generalization. The final layers of the network were fully connected, culminating in a softmax activation function to predict the probability distribution over the target emotion classes (e.g., happy, sad, angry, surprised, etc.).

This architecture was trained using categorical cross-entropy as the loss function and optimized using the Adam optimizer with an appropriate learning rate schedule. The CNN model demonstrated strong performance in recognizing facial expressions, making it a suitable choice for FER tasks, especially when deployed in real-time applications or mobile environments after compression or quantization.

memory that the directed cycles were able

5 Working Principle

CNNs operate by passing the input image through a series of layers to gradually extract important features. The main idea is to take advantage of the spatial structure of image data to process dimensions efficiently.

The basic steps include:

5.1 Convolution Layer

This layer applies a set of filters (kernels) to the input image to extract features such as edges, corners, and textures. Each filter slides over the image and produces a feature map that highlights a specific pattern [20].

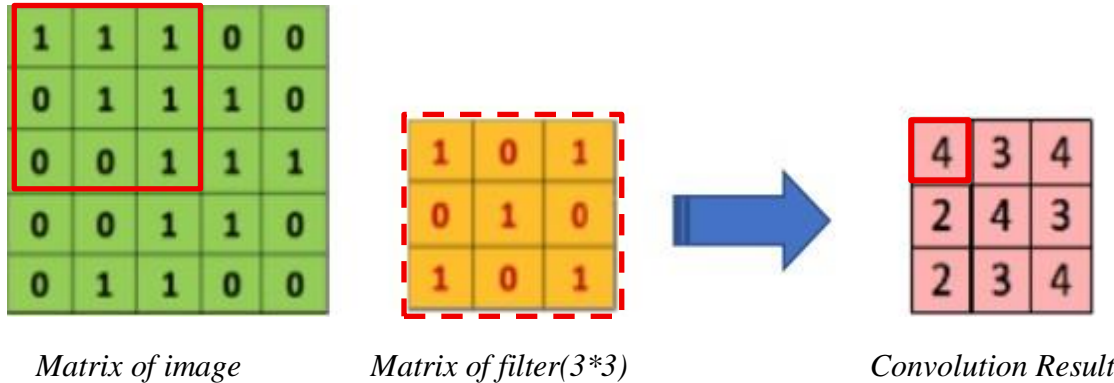


Figure 10: Convolutional Layer

5.2 Activation Function (typically ReLU)

A non-linear function is applied to the feature maps to introduce non-linearity and enable the model to learn complex patterns. ReLU is a function that must be applied to each pixel of an image after convolution. It replaces every negative value with zero. If this function is not applied, the resulting function will be linear, and the XOR problem will persist since no activation function is applied in the convolution layer [25].

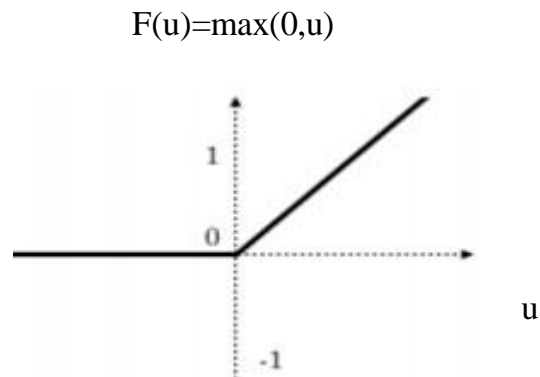


Figure 11: Activation Function

There exist other activation functions such as: Elu, Sigmoid, Softmax, Tanh...

5.3 Pooling Layer

This layer reduces the spatial dimensions of the feature maps, decreasing the number of parameters and computations while retaining the most important features.

Max Pooling is a process where a small window (2 or 3 pixels) slides over an image with steps (strides) equal to the window size. At each step, the maximum value within the window is kept. This reduces the image size while preserving important features [20].

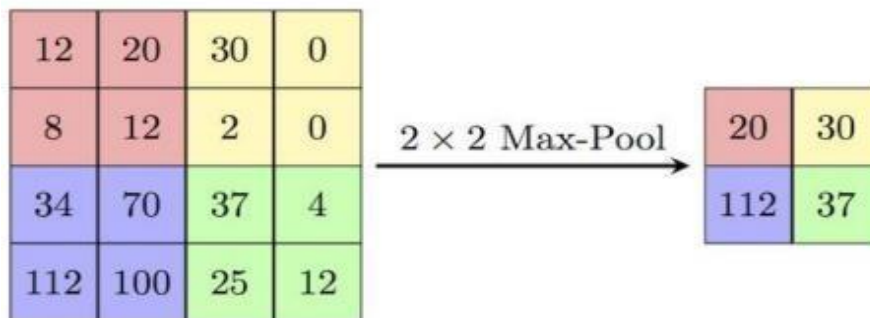
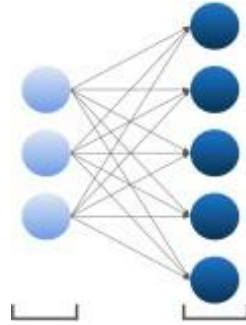


Figure 12: Pooling Layer

5.4 Fully Connected Layers

The fully connected layer (FC): is a perceptron-like layer, where each neuron is connected to all the neurons in the previous layer. It works by combining the characteristics previously learned to make predictions or classifications on input data [20].



Previous layer FC layer

Figure 13:FC Layer

5.5 The loss function (LOSS)

Is used to assess the difference between the predictions of the model and actual values. It measures the error of the model and is used to adjust the weight and network parameters during training, to optimize performance [26]. The types of loss function:

- ***categorical_crossentropy***: For multiple labeling (e.g. emotions)
- ***binary_crossentropy***: For binary classification
- ***mean_squared_error***: To predict values (regression)

6 Data processing techniques

Before training a neural network, proper preprocessing on data is essential for good results. Many beginners struggle not due to poor architectures or lack of regularization, but because they skip key preprocessing steps. The two most important ones are data standardization and data augmentation.

6.1 Data Standardization Techniques

Standardization, also known as Z-score scaling or zero-mean scaling, is a common method used in data preprocessing to scale and center features in machine learning. This method transforms the data in a way that makes it suitable for algorithms that assume a standard normal distribution. Standardization makes the data more Gaussian-like, which is useful for some machine-learning algorithms [27].

6.1.1 Mean Subtraction

Mean subtraction is a fundamental preprocessing step in computer vision tasks. It involves calculating the average pixel value for each channel (e.g., R, G, B for color images or a single channel for grayscale) across the entire training dataset. This computed mean is then subtracted from every pixel in the training and testing datasets [27].

The primary objectives of this technique are:

- **Centering the data:** Shifting the mean of the distribution to zero helps in accelerating convergence during training.
- **Removing global illumination bias:** Lighting conditions affect pixel intensities uniformly; subtracting the mean reduces such consistent noise across images.
- **Stabilizing gradients:** Zero-centered data often leads to more stable and faster convergence when training neural networks.

6.1.2 Scaling

After centering the data using mean subtraction, scaling is used to standardize the variance across features. The standard method divides each pixel by the standard deviation of its respective channel. This process ensures that the transformed data has a standard deviation of 1, making the features more comparable in scale [27].

The benefits include:

- Improved performance in gradient-based optimization, as features on similar scales make it easier for the optimizer to converge.
- Reduced impact of features with high dynamic range, which could otherwise dominate the learning process.

Alternate approaches to scaling:

- Robust scaling: Instead of using standard deviation, one may divide by the interquartile range (IQR) or use the 95th percentile to minimize the impact of outliers.
- Min-max scaling: In some tasks, especially in image display, scaling pixel values to a
- [0, 1] or [-1, 1] range is preferred.

6.1.3 Specialized Methods (used in medical imaging)

In specialized domains like medical imaging, general-purpose normalization techniques may not be sufficient due to variations in data acquisition devices, staining techniques, or imaging protocols [27]. The following are examples of tailored standardization approaches:

- ***Chest X-rays***: Local energy-based normalization helps handle variations from different machines and tube conditions.
- ***H&E stained slides***: Custom normalization reduces color and intensity variations due to lab and time differences.
- ***Raw data (e.g., sinogram from CT)***: A log-transform may be applied as an extra preprocessing step.

6.2 Data Augmentation

Neural networks perform better with large datasets. Data augmentation is the process of artificially expanding a dataset by creating new samples through label-preserving transformations.

Data augmentation is an integral process in deep learning, as in deep learning we need large amounts of data and in some cases it is not feasible to collect thousands or millions of images, so data augmentation comes variability in the dataset [28]. Operations in data augmentation The most commonly used operations are:

- Rotation
- Shearing
- Zooming
- Cropping
- Flipping
- Changing the brightness level

6.3 Optimizers and Learning Rate

Training a neural network involves the minimization of a loss function, which represents the discrepancy between the predicted outputs and the true labels. This process is formalized as the minimization of the empirical risk over the training data. The optimization is performed in two main steps:

- Computing the gradient of the loss with respect to the model parameters (using backpropagation).
- Updating the parameters using optimization algorithms.
- Several optimization algorithms are used in deep learning, each with distinct characteristic

to the rescue. It helps us to increase the size of the dataset and introduce

6.3.1 Stochastic Gradient Descent (SGD)

The most fundamental optimizer, which updates parameters using gradients computed on mini-batches. While simple, it can be slow to converge and sensitive to the choice of learning rate.

Momentum enhances SGD by adding a fraction of the previous update to the current one, helping the optimizer escape shallow local minima and smooth out oscillations. Adam (Adaptive Moment Estimation) One of the most widely used optimizers, Adam combines the advantages of

Momentum and RMSProp. It adapts the learning rate for each parameter individually based on estimates of the first and second moments of the gradients. It generally leads to faster convergence and better performance across a wide range of tasks. RMSProp Maintains a moving average of the squared gradients and normalizes the update step, especially effective for non-stationary objectives [19].

6.4 Learning Rate

The learning rate is a hyperparameter that controls the size of the steps taken toward the minimum of the loss function during training. Its proper setting is critical:

- **Too high:** May cause the model to diverge or miss the optimal solution.
- **Too low:** Leads to slow convergence or getting stuck in local minima.

Learning rate scheduling techniques (e.g., step decay, exponential decay, or cyclic learning rate) are often used to dynamically adjust the learning rate during training for better performance [29].

6.5 Weight initialization and pretrained networks

If two neurons (or filters) start with identical weights and biases, they will always receive identical gradients, leading them to learn the same features. This reduces the network's ability to learn diverse patterns — a problem called symmetry [25].

6.5.1 Breaking Symmetry – Random Initialization

A key problem in neural networks is symmetry: if two neurons in the same layer start with identical weights and biases, they will receive identical gradients during training, causing them to learn the same features. This redundancy limits the network's representational power. To address this, random initialization is used to break symmetry and promote diversity in learning.

6.5.2 Using Pretrained Networks

Rather than training a deep network from scratch, it is often more efficient to leverage pretrained models that have been trained on large-scale datasets such as ImageNet. This approach is especially useful when the available dataset is small or the computational resources are limited.

6.6 Regularization

Regularization is a fundamental technique in machine learning aimed at reducing overfitting and enhancing the model's generalization capability. Common methods include L1 regularization, which promotes sparsity by driving some weights to zero, and L2 regularization, which encourages smaller, evenly distributed weights. Additionally, the Max-Norm constraint limits the magnitude of weight vectors to maintain training stability. One of the most effective techniques is Dropout, which randomly deactivates a subset of neurons during training, thereby reducing dependency on specific units and improving overall model robustness [20].

7 CNN Facial Recognition Apps

7.1 How facial feelings can be recognized

7.1.1 Preprocessing of the face image

This stage is very important in order to improve the performance of facial recognition technology, as it includes many processes such as image clarity, measurement, contrast adjustment and other optimizations, where the data is often incomplete and lacks certain features that may interest us, is not formatted, or contains many errors [30].

7.1.2 Face Detection

The task of face detection has become one of the tasks that are in great demand in many applications, so it has been used in the field of security systems and many applications, where at this stage the color image is converted into a gray image, because the color image consists of three RGB compounds, and thus the detection of the face in a color image becomes more complex and takes more time compared to the gray image that consists of one channel, so we convert the images into gray images. In the image, the computer is trained on a large number of images so that they contain only the main facial features such as the eye, eyebrow, mouth, nose. After repeating the process of training on thousands of images, the computer is able to locate the face [30].

7.1.3 Feature Extraction

At this stage, features are extracted from the face to classify expression, where these features are located in the eyebrow – the outer edge of the eye – the inner edge of the eye – around the mouth – around the nose, and these features are found through the training of the neural folding network.

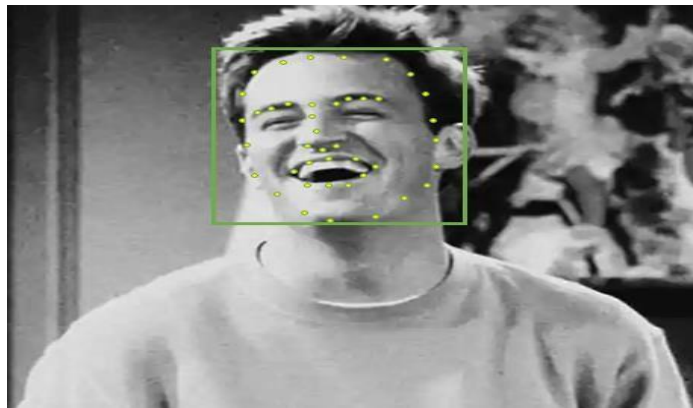


Figure 14: Identifying facial features

7.1.4 Classification of emotional expression

The final stage of facial expression recognition is through facial expression analysis; it is the classification stage where the network is trained on a set of images to classify the emotional state of the face into a specific expression.

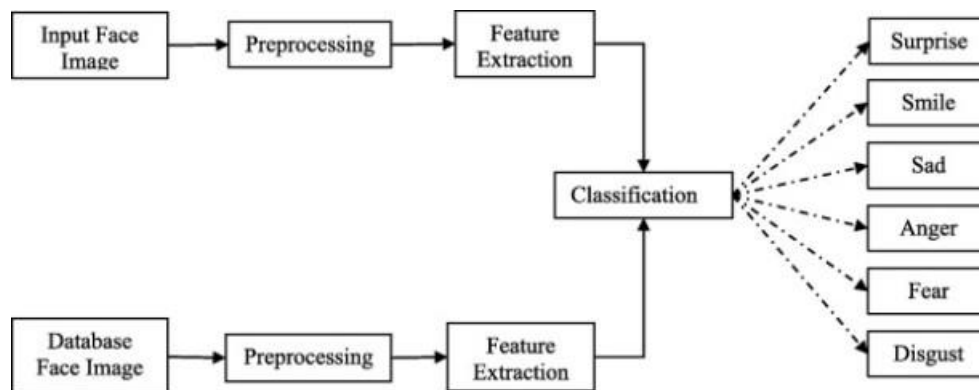


Figure 15: Facial Expression Regression Workflow

7.2 CNN Facial Recognition Apps

One of the most significant areas of artificial intelligence, particularly in the study of photos and videos, is the use of Convolutional Neural Networks (CNN) for facial emotion recognition. The most significant applications are explained in detail below [31]:

7.2.1 Systems for Monitoring Emotions in Real Time Explanation

A CNN model trained on an emotion dataset (like CK+) is applied to the image after a person's face is captured by a camera. Emotions including happiness, rage, despair, and surprise are all recognized by the model. It is employed in meetings to assess participants' emotional states or in smart classrooms to monitor student interaction.

7.2.2 Human-Computer Interaction

This makes it possible for devices (such as robots or apps) to comprehend user emotions and respond appropriately. Application: Depending on the player's feelings, games may alter the music or level of difficulty. Your emotional state is detected by the camera in smart assistant apps (like Alexa or Siri), and they react with compassion or understanding.

7.2.3 Apps for smartphones

CNN models are included into smartphone applications (using TensorFlow Lite or CoreML technologies), which use the front camera to analyse facial expressions and provide immediate results.

Application: Apps for mental health, like those that track a user's mood all day.

7.2.4 psychology and medicine

CNN is used to precisely assess patients' emotions, particularly those who struggle with verbal expression.

7.2.5 Examine how viewers feel about videos

CNN is used to analyse multiple people's emotions simultaneously, whether in live streaming or YouTube recording.

Use: Promoting and enhancing the user experience based on their actual responses.

7.2.6 Analysis of Marketing and Customer Experience

The faces of customers are examined while they interact with a product or service. Application: Cameras in stores use facial expression analysis to gauge customer contentment, which aids companies in refining their tactics.

7.2.7 Identifying deception or malicious intent in security

CNN is able to identify minute variations in facial expressions that could be signs of stress or dishonesty.

Use: in detention facilities or airports.

7.2.8 Education

Affection affects learners' learning status. Using facial expression recognition technology, computers can judge learners' status by recognizing their facial expressions.

In education, teachers can use the analysis results to understand a student's learning and ability to accept, and then formulate reasonable teaching plans. At the same time, they can pay attention to students' internal emotions, which contributes to their physiological health, especially in distance learning.

8 Conclusion

Facial expression recognition technology, like any technology, is not without flaws and challenges. One challenge is that datasets are categorized by people, and different people can read and interpret emotions in different ways. This means that this technology is easy to fool. Computers associate certain facial expressions with certain emotions, but they fail to distinguish, for example, between a malicious smile and a genuine smile. Therefore, this technology should not be used in decisions that affect people's lives and access to opportunities, such as hiring decisions, because it is not accurate enough and can lead to biased decisions.

Chapter 3

Model Development and Performance Evaluation

1 Introduction

This chapter deals with the practical implementation of a facial expression recognition system based on Convolutional Neural Networks (CNNs). In the beginning of the chapter, the work environment is presented and the dataset used in the study is described. Then the procedure of data preprocessing and model design are introduced, followed by a description of training parameters. Next, model evaluation strategy implementation is covered, along with a description of the prediction technique for classifying new images. Finally, we provide an in-depth analysis of the results obtained through various evaluation metrics and report visualizations.

2 Tools and Work Environment

A Python-based environment was used to run these procedures. The following setup and tools assisted in carrying out this work:

- **Operating System:** Windows 10
- **Programming Language: Python** 3.12.0 is the newest major release of the Python programming language, and it contains many new features and optimizations.

The proposed application was developed using a Python-based environment alongside a collection of libraries and tools dedicated to building the web interface, real-time interaction, image processing, emotion detection, and result visualization.

2.1 Jupyter Notebook

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality.

JupyterLab was chosen as the IDE in this project for preparing, training, and evaluating the CNN model. This transported the organization of codes and results into one single interface with more advanced productive features than does the traditional Jupyter Notebook environment [32].

2.2 Google Colab

during some occasions, this environment was used, providing GPU support integrated with it that made training models much faster, especially while working on huge data sets.

2.3 Visual Studio Code Environment

Visual Studio Code is a fast and powerful source-code editor for Windows, Linux, and macOS, created by Microsoft. The editor is mostly widely used for the development of Python and Flask applications. With extension support, this code editor supports almost anything, thus making it an efficient IDE [33].

2.3.1 Key Features

- **Python Support:** The official python extension supports IntelliSense (code auto-completion) and debugging.
- **Flask Integration:** VS Code supports the development of Flask applications, including project creation, running local servers, and debugging.
- **Built-in Git Support:** Direct integration with Git version control system allows easy

source code management: <https://code.visualstudio.com/docs/languages/python>

2.3.2 Reason for Selection

Visual Studio Code was selected as the main development environment for this project due to its flexibility and ease and its excellent Python and Flask support with an active developer community.

2.4 Python Tools, Libraries and Frameworks

This application depends on a list of Python libraries and tools, each serving a different purpose- from creating the web interface to image processing, deep learning, and performance evaluation. The libraries can thus be divided into different categories:

2.4.1 Web Framework

To build the web application and enable real-time communication features, the following frameworks were used:

Flask: Flask is an extremely lightweight Python web framework that is supposed to be used for creating web applications, simple or complex. It is very easy to learn and works well in tandem with many other Python libraries. Flask was invoked in this project simply because it is lightweight, easily extensible, and efficiently supports web-based AI applications.

Flask-SocketIO: extends Flask to enable real-time, bi-directional communication between the web client and server, which is an excellent fit for applications that require instant updates.

2.4.2 Libraries and Tools Used

This project was developed in a Python environment, employing numerous libraries to facilitate web interface development, image processing, emotion analysis, the evaluation of the performance of models, and visualization of results. The following are libraries and tools utilized:

OpenCV, PIL, Matplotlib, Seaborn: For image processing and data visualization. These libraries were introduced in Chapter 3 and are reused in this application.

werkzeug.utils.secure_filename: To securely handle file names when uploading images.

TensorFlow, Keras, ImageDataGenerator: For building, training, and managing Convolutional Neural Network (CNN) models. These libraries were also introduced in Chapter 3.

Scikit-learn (sklearn): For evaluating the model's performance through classification reports and confusion matrices.

Os, shutil, zipfile, time, contextlib: Utility tools for interacting with the operating system, managing files, handling archives, controlling execution timing, and managing code contexts.

Emoji: To display the appropriate emoji based on the detected emotion.

emotion_recognition.prediction: A custom module containing the core functions for face detection (`get_face_from_frame`) and emotion prediction (`get_emotions_from_face`).

3 Dataset Description

The facial expression image database utilized in this work is the Extended Cohn-Kanade (CK+) dataset, which is a popular and tested facial expression image database for tasks involving the recognition of emotions. This set contains images with labels under various expressions captured in a controlled environment. The default data size is 981 grayscale pictures of 48×48 pixel dimensions representing 7 different categories of emotion type. This specific version used was obtained from Kaggle:<https://www.kaggle.com/datasets/shawon10/ckplus>

- **The dataset includes the following emotion classes:** (Angry, Contempt, Disgust, Fear, Happy, Sad, and Surprise).
- **Image Dimensions:** All images in the dataset are originally 48×48 pixels and in grayscale format, which eliminates the need for additional preprocessing to convert images.



Figure 16:CK+ Dataset

3.1 Data Augmentation

Since the original dataset was relatively small (981 images), a data augmentation process was applied to increase the number of images in each class and improve model generalization. The augmentation operations included : (rotation, flipping,

Zooming, blur...)

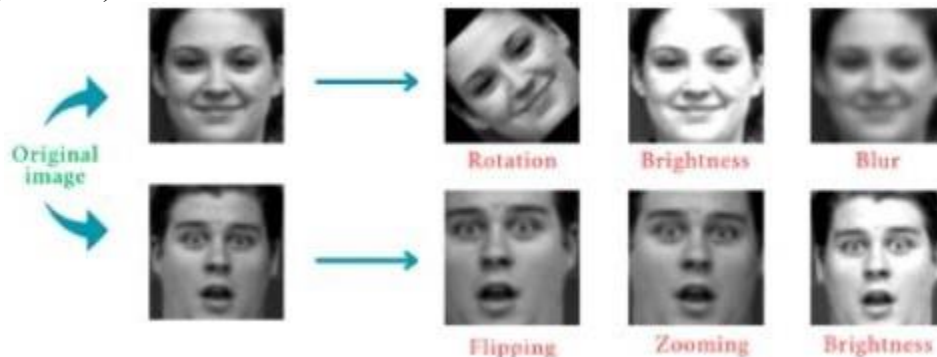


Figure 17:Types of Data Augmentation

After data augmentation, the dataset was balanced by ensuring that each emotion class contained exactly 1,000 images, resulting in a total of 7,000 images. This balancing process through augmentation was crucial for enhancing the model’s performance, addressing class imbalance issues, and reducing the risk of overfitting.

EMOTIONS		BEFORE	AFTER
Angry	😡	135	1000
Contempt	😏	54	1000
Disgust	🤢	177	1000
Fear	😱	75	1000
Happy	😄	207	1000
Sad	😞	84	1000
Surprise	😲	249	1000

Figure 18: Before and After Data Augmentation and Balancing

In preparation for the efficient and reliable training of the proposed CNN, an important set of preprocessing operations were considered. The steps were undertaken to ensure consistency in quality and suitability of the input data for deep learning tasks. The preprocessing phase entailed the following steps:

3.2 Data Loading and Label Assignment

Since the images resided in separate folders assigned with seven emotions classes, the very first step was to maintain an iterative loading of the images, one by one, from the folder. For every image, a number was assigned as labels to them bearing the name of a class. This numeric class encoding is necessary for supervised learning models in case they learn during training from numeric labels.

Purpose: Link each image to its respective emotion category in an interpretable format for model assessment.

3.3 Image Resizing

Although the original dataset had uniform dimensions of 48×48 pixels, the loading procedure involved resizing each individual image systematically. This procedure guaranteed that any image considered inconsistent or corrupted would be resized to the standard dimension, thus maintaining the uniformity of input definiteness for the CNN.

- **Purpose:** To ensure that the input shape for the model is made consistent and that an error is not

```
# Data preparation
# =====
data = []
for category in CATEGORIES:
    folder = os.path.join(DIRECTORY, category)
    label = CATEGORIES.index(category)
    for img in os.listdir(folder):
        img_path = os.path.join(folder, img)
        img_arr = cv2.imread(img_path)
        if img_arr is not None:
            img_arr = cv2.resize(img_arr, (48,48))
            data.append([img_arr, label])
        else:
            print(f"Failed to load image {img}")
print(f"Total data samples: {len(data)}")
```

Data Loading & Label Assignment

Image Resizing

Result: Total data samples: 7000

Figure 19: Load & Resize data

3.4 Grayscale Conversion

At this point, although the original dataset images in CK+ are already in grayscale, for this reason, programmatic conversion is performed using the OpenCV library (`cv2.cvtColor`) to create

a truly uniform dataset. Especially after data augmentation operations, some images may inadvertently be generated in RGB format.

- encountered during training.
- **Purpose:** The main intent behind this step was to assure that all the images remain in a single-channel (grayscale) format with (48×48×1) dimensions. This assured them of

reducing the computational cost of the CNN model all the while preserving the critical facial features that might impact accurate emotion classification.

3.5 Normalization of Pixel Values

After conversion into greyscale, the pixel intensity values, which typically range from 0 to 255, were normalized in the range of 0 to 1 by dividing all values with 255. This normalization is a common standardization in deep learning preprocessing since this step speeds up the convergence of the learning algorithm and stabilizes the gradients while being backpropagated.

- **Purpose:** For better convergence of the model and numerical stability in the training phase.

3.6 Reshaping Input Data

After normalization, the other dimension was added to each image array for the color channels. Since grayscale images have only one color channel, the dimension was set to **1**. Finally, every image has a shape of (**48, 48, 1**), corresponding to the desired **Keras CNN** input format.

- **Purpose:** To set the shape of the input data to fit the CNN model's input requirements.

```

x, y = [], []
for features, label in data:
    # تحويل الصورة إلى رمادية
    gray_image = cv2.cvtColor(features, cv2.COLOR_BGR2GRAY)
    x.append(gray_image)
    y.append(label)

# تطبيع القيم Numpy تحويل إلى مصفوفة
X = np.array(x) / 255.0
X = X.reshape(-1, 48, 48, 1)
Y = np.array(y)

print(f"X shape: {X.shape}")
print(f"Y shape: {Y.shape}")

```

Grayscale Conversion (indicated by a red arrow pointing to the `cv2.cvtColor` line)

Normalizing Values (indicated by a red arrow pointing to the `X = np.array(x) / 255.0` line)

Reshaping Input Data (indicated by a red arrow pointing to the `X = X.reshape(-1, 48, 48, 1)` line)

Result: X shape: (7000, 48, 48, 1)
Y shape: (7000,)

Figure 20: Gray, Norm and Reshape data

3.7 Data Splitting

The `train_test_split()` function from `sklearn.model_selection` is used to divide a dataset into two main subsets: a training set and a test set. The data is first separated into features (X) and labels (y), then split into `X_train`, `X_test`, `y_train`, and `y_test`. It is generally recommended to allocate a larger portion of the data to the training set than to the test set to allow the model to learn effectively.

3.7.1 Train set

The training dataset is a set of data that was utilized to fit the model. The dataset on which the model is trained. This data is seen and learned by the model.

3.7.2 Test set

The test dataset is a subset of the training dataset that is utilized to give an accurate evaluation of a final model fit.

3.7.3 Importance of Data Splitting

- Prevents Overfitting by ensuring the model generalizes well to unseen data.
- Provides an objective way to evaluate model performance on new, untouched data.
- Allows for safe hyperparameter tuning during training without affecting final test results.

```
# 3. data splitting
# =====
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.20, random_state=42, stratify=Y)
print(f"Training data shape: {X_train.shape}")
print(f"Testing data shape: {X_test.shape}")
```

Splitting the Dataset

Result:

```
Training data shape: (5600, 48, 48, 1)
Testing data shape: (1400, 48, 48, 1)
```

Figure 21: The `train_test_split` Function

4 Model Design (CNN)

In this project, a Convolutional Neural Network (CNN) model was constructed, specifically designed for classifying images according to emotion categories. The model consists of several interconnected layers, each performing a distinct function in processing and analyzing the data, ultimately leading to final classification.

```
# 4. Building the model
# =====
model = Sequential([
    Input(shape=X.shape[1:]),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Dropout(0.25),

    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Dropout(0.25),

    Conv2D(32, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Dropout(0.25),

    Flatten(),
    Dense(128, activation='relu'),
    Dense(64, activation='relu'),
    Dropout(0.4),
    Dense(len(CATEGORIES), activation='softmax')
])

model.summary()
```

Figure 22: CNN Model Architecture

The following is a detailed description of the model's components:

4.1 Convolutional Layers

The model includes *three consecutive Convolutional layers*.

Each layer applies a set of filters with a size of (3×3) on the input images to extract distinguishing features.

- The first layer uses 128 filters, followed by 64 filters in the second layer, and 32 filters in the third layer.
- This reduction in the number of filters progressively captures features from general to more detailed ones:
 - The first layer detects basic edges and lines.
 - The second layer identifies more complex patterns.
 - The third layer focuses on fine details specific to facial expressions.

The *ReLU (Rectified Linear Unit) activation function* is applied after each Convolutional operation to introduce non-linearity and improve the model's ability to learn complex relationships.

4.2 Pooling Layers

After each Convolutional layer, a *MaxPooling2D* layer with a pool size of (2×2) is applied. This operation:

- Decreases the spatial dimensions of the feature maps.
- Retains the most dominant extracted features.
- Reduces the computational cost and speeds up the training.

4.3 Dropout Layers

After every *MaxPooling* operation, a Dropout layer of **25%** is employed, and another *Dropout layer* with a **40%** rate is placed right before the output.

Its function lies in preventing overfitting, as dropout randomly excludes some neurons during the training process.

In doing so, the model is then forced to generalize learning instead of just over fitting to certain neurons.

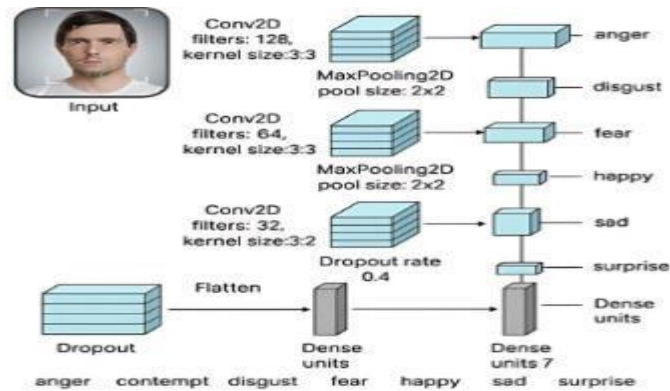


Figure 24: CNN for Emotion Classification

5 Training Parameters

To train the proposed CNN model effectively, specific training parameters were selected to optimize performance and ensure stable learning:

Optimizer: It employs an Adam optimizer because of its fast convergence properties and adaptive capabilities in a training environment.

Learning Rate: It was set to 0.0001 as a balanced option for ensuring convergence speed and its stability.

Loss Function: It uses Categorical Crossentropy, which is best to handle cases involving multi-class classification.

Evaluation Metric: Accuracy is taken to practically evaluate the model in operation.

Number of Epochs: It was set at 30 to allow maximum learning of the model.

Batch Size: It has chosen a batch size of 64 for training in an efficient and stable way.

Callback ModelCheckpoint: This callback monitors validation loss, saving weights of the model whenever it observes an improvement.

Callback EarlyStopping: This utility halts the training process early if the validation loss has not improved for 4 epochs, thereby helping prevent overfitting and saving unnecessary time for training.

6 Model Evaluation Strategy

The model's evaluation was done by using the test dataset that was set apart during the splitting of data to get an unbiased estimate of the actual performance.

Two valuable metrics were derived through the `evaluate()` function of Keras:

- **Test Loss:** This quantifies the prediction error of the model with respect to the given test data.
- **Test Accuracy:** This defines the ratio of correctly predicted instances to the total tested samples.

To understand the system's classification possibilities more thoroughly:

- A confusion matrix was created, showing the various counts of correct and incorrect predictions for every emotion category.
- A classification report was derived, loaded with precision, recall, and F1-score regarding every class, which fully evaluates the predictive power of the model.

This evaluation strategy facilitates a far-reaching analysis that situates the model's strengths and weaknesses in classifying facial emotions within multiple categories.

7 Prediction Mechanism

The prediction technique works on making class predictions for new data using the model's parameters that have already been trained through some learning mechanism. For this approach to work, there are two types of data: the test dataset and external data from the real world. In both cases, the model generates probability distributions regarding the categories of emotions using the softmax activation function in the final dense layer. The emotion with the highest probability becomes the predetermined class. These distributions are labeled as probability distributions and are all turned into the required class labels through the `np.argmax()` function.

7.1 Prediction on Test Data

In the model's evaluation, the model is put through its paces on the test set which is an allocated portion of the greater dataset which was set aside during the training phase. The model generates probability distributions for each sample and saves them using the `predict()` function. Through the use of the `np.argmax()` function, each image's class will now correspond to a predicted value which holds the highest probability outcome.

This process allows for an objective assessment of the model's generalization ability, as it operates on data it has never seen before but shares the same characteristics and preprocessing steps as the training data.

7.2 Prediction on New External Data

In addition, the model can be used to predict insight on new external data. External data comes from the real-world scenario and can be utilized in design. However, if the outcome is to be valuable and precise, the provided data needs to be in the same condition as the data on which the model was trained.

To accomplish this goal, an external data set must also go through the same preprocessing that was done during training, such as:

- The size of the image should be 48 x 48 pixels.
- The image must be in grayscale.
- Use the same scaling, resizing and normalization that were applied during training.

If these requirements are not met, the model's predictive performance may decline, as it was optimized to recognize patterns based on specific data formats and structures. Ensuring feature consistency allows the model to effectively interpret new inputs and produce accurate classifications.

8 Results Analysis

This section deals with the performance assessment of the proposed CNN model using a series of evaluation methodologies. The model learned quickly and generalized well, with training and testing accuracies of 99% and 98%, respectively. To validate this, we have shown and briefly discussed a set of results depicting accuracy and loss curves, the confusion matrix, and the classification report.

8.1 Training and Validation Accuracy and Loss Curves

The initial set of graphs shows training-validation accuracy and loss values, calculated for various training epochs.

8.1.1 Accuracy Curve

The graph depicts training and validation accuracies increasing steadily as the number of epochs increased. Initially, accuracy values were low, but after a few epochs, the two curves started rising rapidly. Training accuracy reached 99%, while validation accuracy stuck at 98%, signaling that the model could efficiently learn the patterns from the data without overfitting.

8.1.2 Loss Curve

In opposition to the accuracy curves, the loss curves continue to go down for both training and validation datasets. At the start of the training, loss values stayed high, indicating poor predictions. But, as training process proceeded, observes a gradual decrease in loss until it eventually flattens out, almost touching 0.02 for training loss and going near 0.05 for validation loss. This confirms the stability of the model during its training and ensures it performs well.

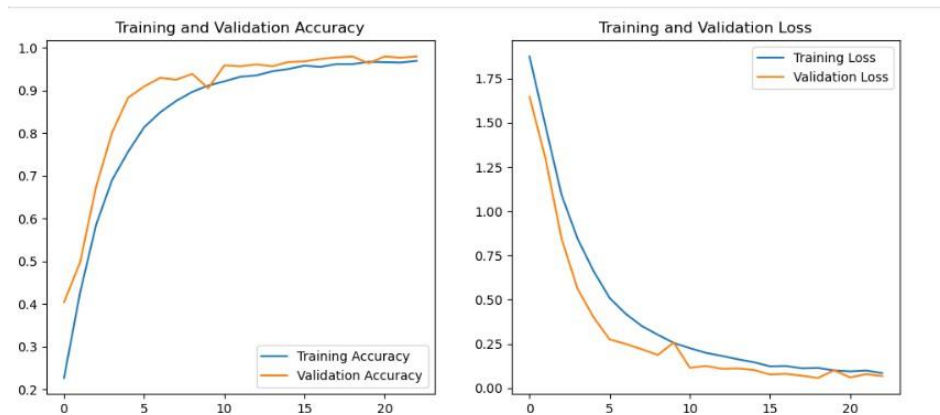


Figure 25: Model Accuracy and Loss Curves

8.2 Confusion Matrix Analysis

- A confusion matrix breaks down the prediction evaluation details with accuracy and error in each box for the assigned class of true labels.
- The diagonal boxes have the highest value, which means there were plenty of correct predictions for each class.
- Misclassification is limited and off-diagonal.
- For example, class 0 was predicted correctly 194 times out of 200, while for class 5, 193 predictions were correct.

Hence, our model performs quite well in distinguishing different facial expressions with a little amount of apparent confusion.

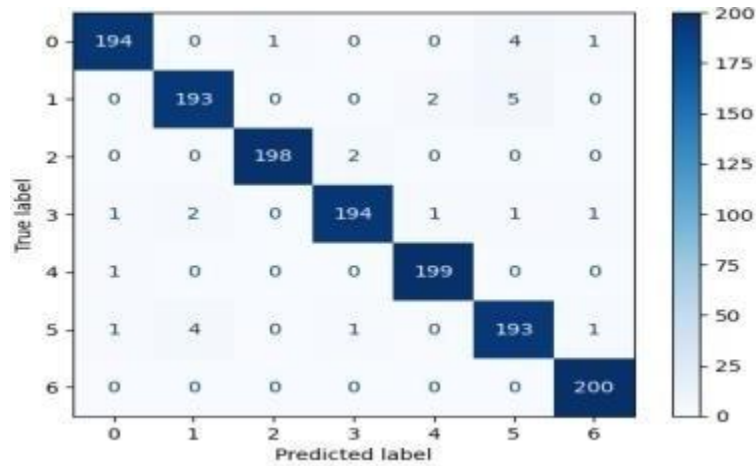


Figure 26: Confusion Matrix Result

8.3 Classification Report

The classification report provides the model's precision, recall, and F1-score for each class:

- **Precision:** The number of true positive predictions divided by the number of all positive predictions.
- **Recall:** The number of true positive predictions divided by the number of all actual positives.
- **F1-Score:** A balanced measure, being the harmonic mean of precision and recall.

In this report:

- Most classes have had precision and recall values above 0.95, signifying excellent predictive performance.
- The overall accuracy is 98%, further confirming the accuracy and loss curves' findings.
- The two values assigned to the macro average and the weighted average are 0.98, showing uniformity among the classes.

	precision	recall	f1-score	support
0	0.98	0.97	0.98	200
1	0.97	0.96	0.97	200
2	0.99	0.99	0.99	200
3	0.98	0.97	0.98	200
4	0.99	0.99	0.99	200
5	0.95	0.96	0.96	200
6	0.99	1.00	0.99	200
accuracy			0.98	1400
macro avg	0.98	0.98	0.98	1400
weighted avg	0.98	0.98	0.98	1400

Figure 27: Classification Report

9 Conclusion

In summary, this chapter elaborated on the practical development of a CNN-based facial expression system for recognition. With respect to adequate data preparation, model design, and evaluation, the system attained an outstanding performance: 99% training accuracy and 98% test accuracy. The analysis of results ratified the model in terms of generalization abilities on unseen data. However, some avenues for improvement were realized during this study. These findings validate the effectiveness of the proposed approach and set the stage for potential future enhancements.

Chapter 4:

Application Implementation and User Interface Description

1 Introduction

This chapter comprises the concentrating on the realized system and explaining each functionality steps by steps. It covers the important phases: data input, model training, model evaluation, and predicting emotions on images. The chapter is divided into parts that deal with both the application of the pre-trained model and the creation of new models for grayscale and RGB images. Results, both visual and textual, are generated interactively so that users can download and reuse the trained models, which indeed substantiates the adaptability and user-friendliness of the system.

2 Application Description

The app uses deep learning to detect emotions based on facial expressions in photos or live camera feeds. It can identify emotions like joy, sadness, anger, and confusion without targeting a specific audience. The results are presented in the form of emojis, emotion labels, and accuracy percentages to make them easy to understand. The app can be used in many fields such as education, marketing, mental health, and human-computer interaction.

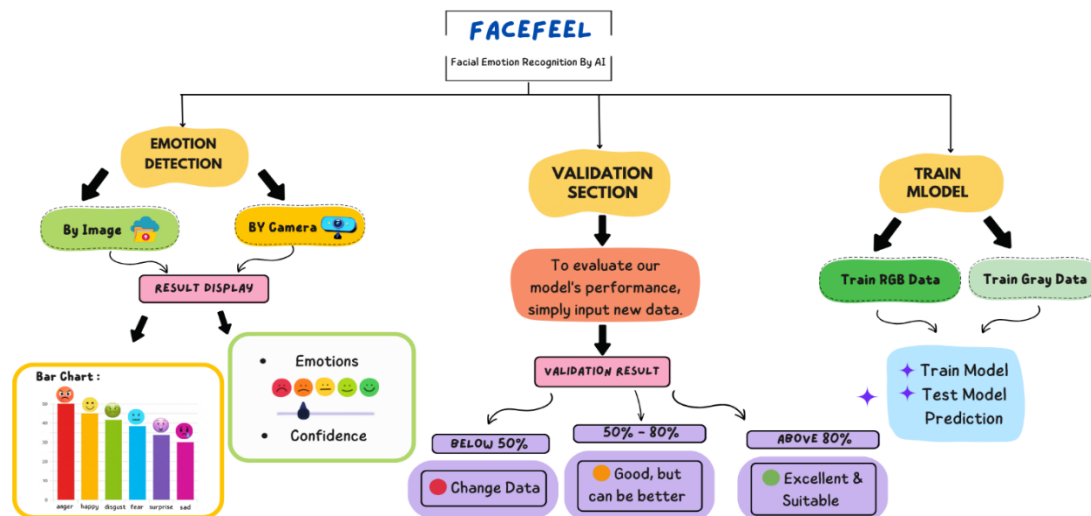


Figure 28: System Architecture Overview

3 Application Objective

This application aims to empower users to accurately detect emotions through images or camera

input, besides allowing the training of new customized models on grayscale or colored images. Being user-friendly with an interactive interface for easy data uploading, model training, performance evaluation, and emotion prediction set it as a good tool that is available to all kinds of individuals, even those without any deep technical backgrounds.

4 Application Features

The application provides several important features for flexible and reliable facial emotion recognition. It allows users to identify emotions, upload and train data, and test models easily. The most important features are:

- Detect emotions from a webcam using our pre-trained model that supports both colored and grayscale images.
- Upload one or more images to detect emotions using our pre-trained model.
- Allow uploading new datasets (Grayscale or RGB) and train models using them.
- Test the trained model using new dataset and evaluate its performance.
- Predict emotions using the trained and stored model.
- Present analysis and classification results through graphical representations, confusion matrices, and statistical reports.

5 Application Interface

This section presents the distinct details of the application's graphical user interface (GUI). It describes the main page, emphasizing how users interact with the system to accomplish various functions including emotion detection, image uploading, model training, and result visualization.

5.1 Homepage

The homepage serves as the main entry point of the application, introducing users to its purpose and core functionalities. It includes:

- *Application name and logo FaceFeel* at the top
- *Navigation bar with direct access to:* (Home, About, Emotion Detection, Model

Evaluation, Train New Model → (Train on RGB Data, Train on Grayscale Data), Contact)

- **“Try Now” Button Overview:** Provides quick access to emotion detection, model evaluation, and an overview of the app’s key features.

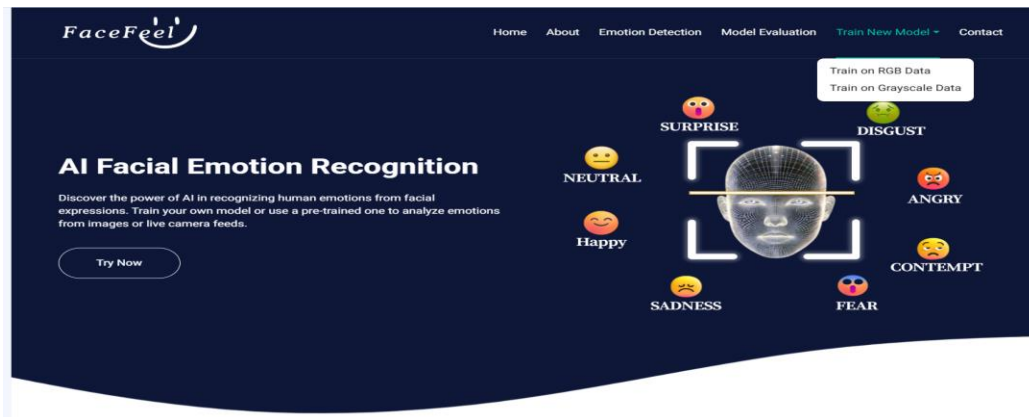


Figure 29: Main Homepage Interface of FaceFeel Application

5.2 About Section

This part contains an outline of the goals of the project concerning the functions of artificial intelligence, specifically, facial emotion recognition, along with the project description and prominent attributes listed distinctly.

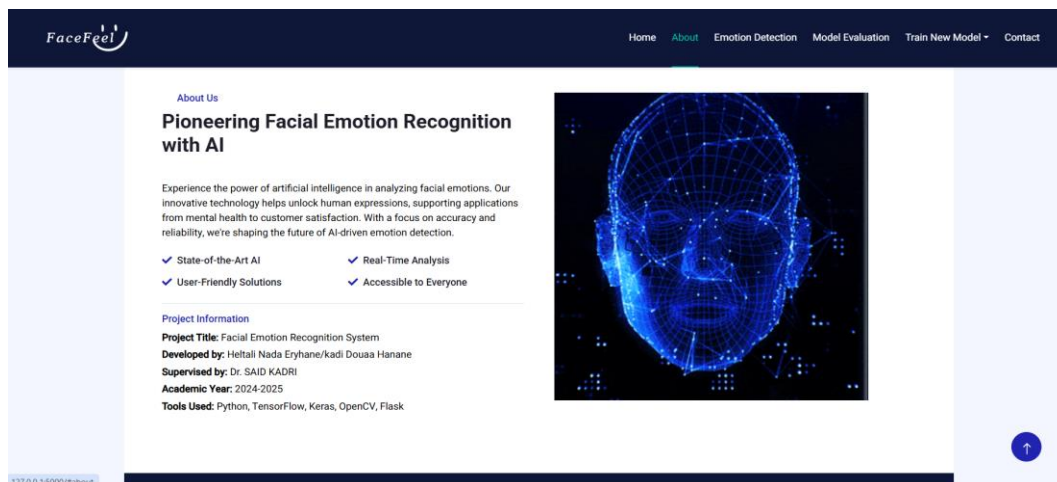


Figure 30: About Section Interface of FaceFeel Application

5.3 Contact Section

This section can be reached via the navigation bar. It will take users to the footer of the application where contact details and other links are available.

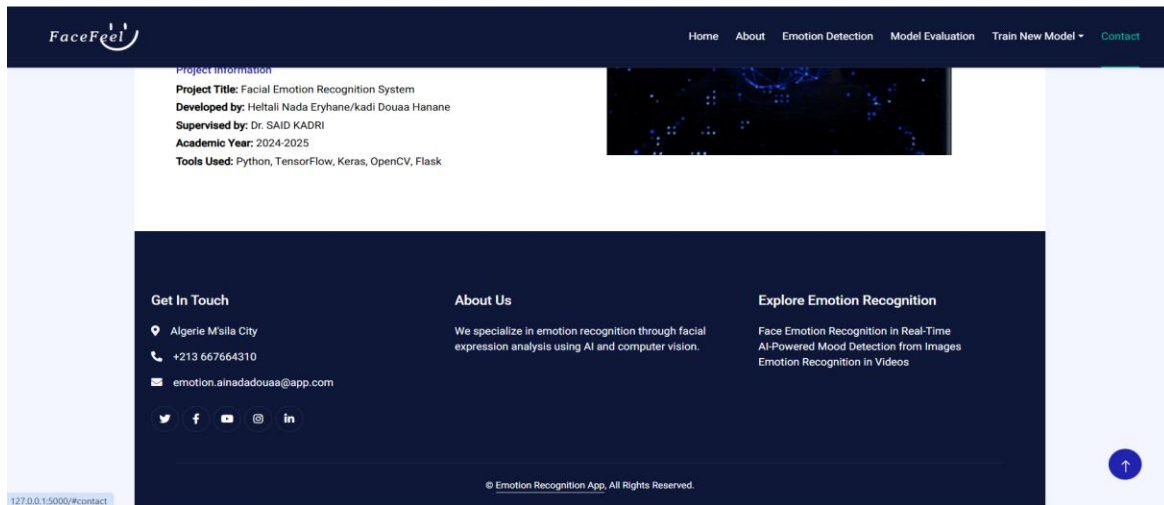


Figure 31 : Contact via Footer

5.4 Emotion Detection

This section allows users to transform their photos into expressive emojis using a pre-trained model. The user can choose between two input methods:

By Image: Select an image file from the device.

By Camera: Use the device's webcam for real-time emotion detection.

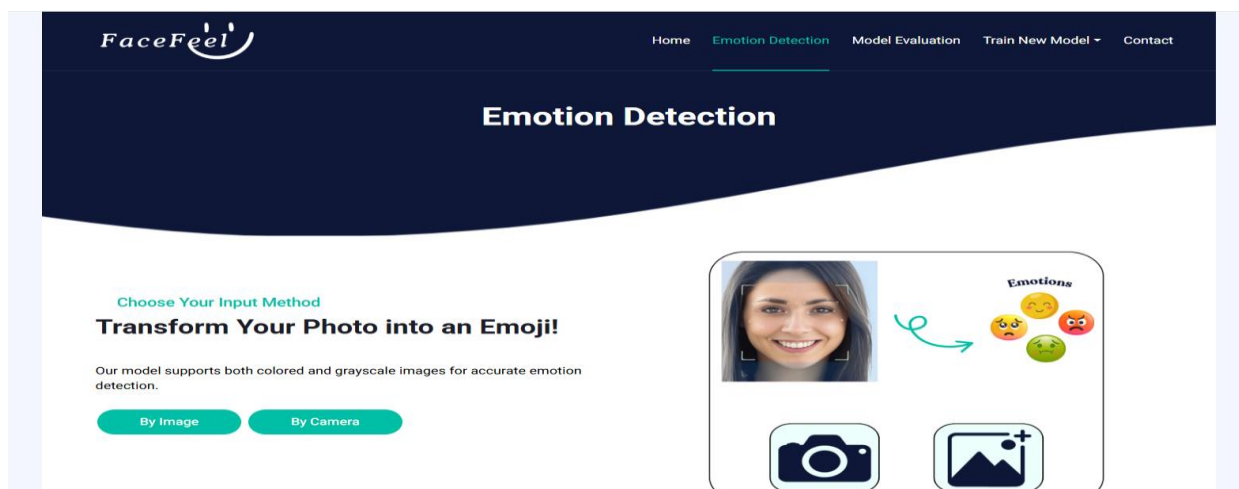


Figure 32: Input Method Selection “By Image” or “By Camera”

- If **By Image option** is selected opens a drag-and-drop upload interface where users can drop one or multiple images or select them via a file picker.

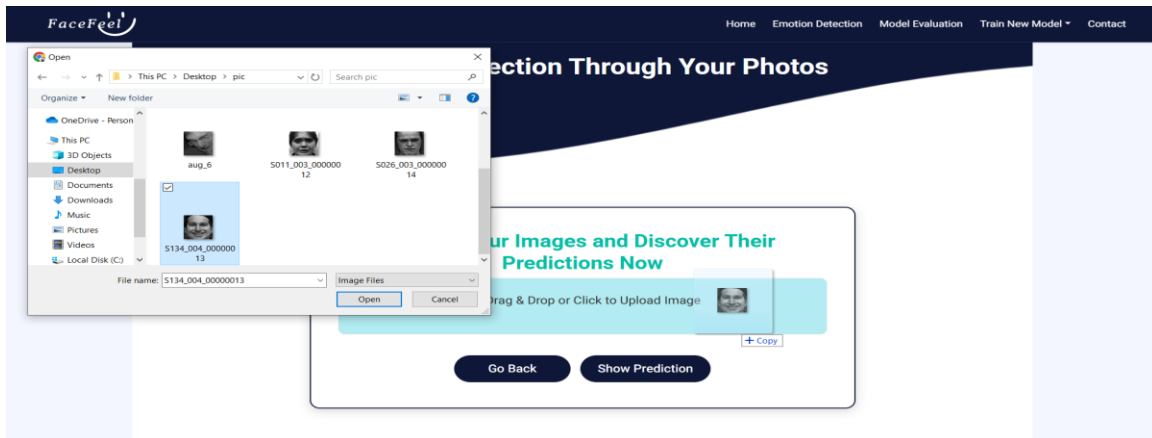


Figure 33: Drag-and-Drop Upload Interface for Image Selection

After choosing images, the user may click Show Prediction , see “Detecting emotions now...” message for a couple of seconds, and then will be redirected to the result page.

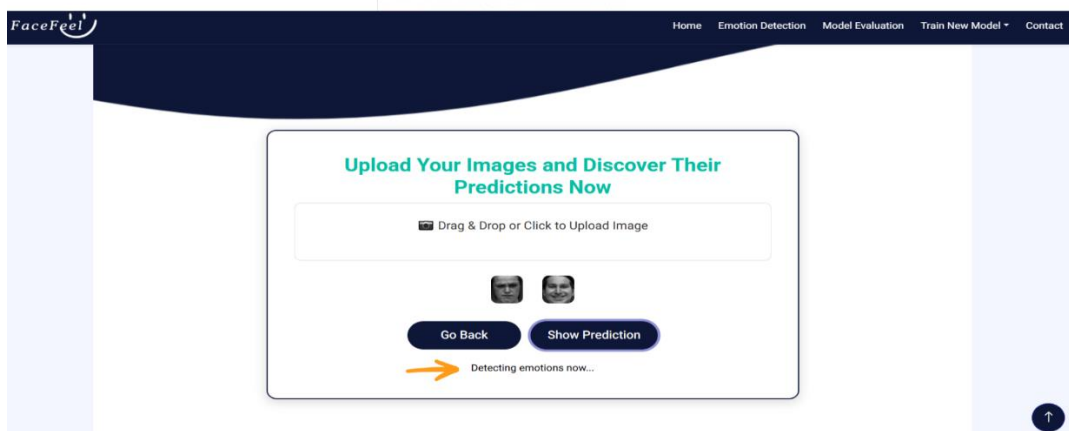


Figure 34: Notification Displayed During Prediction

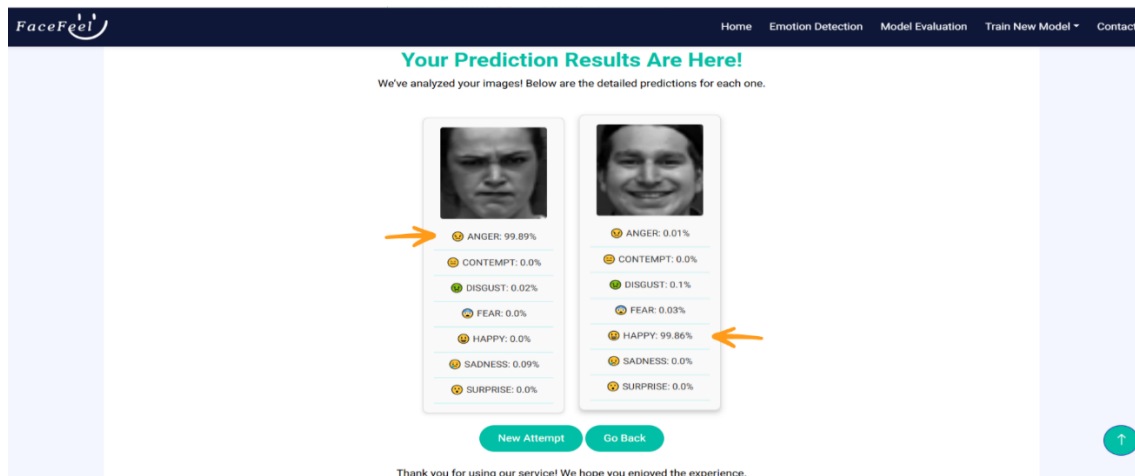


Figure 35: Emotion Prediction Results

- If **By Camera option** is selected, the application activates the webcam, draws a bounding box around each detected face, and performs emotion analysis in real time. The detected emotions will be displayed as a live bar chart representing real-time emotion analysis results.

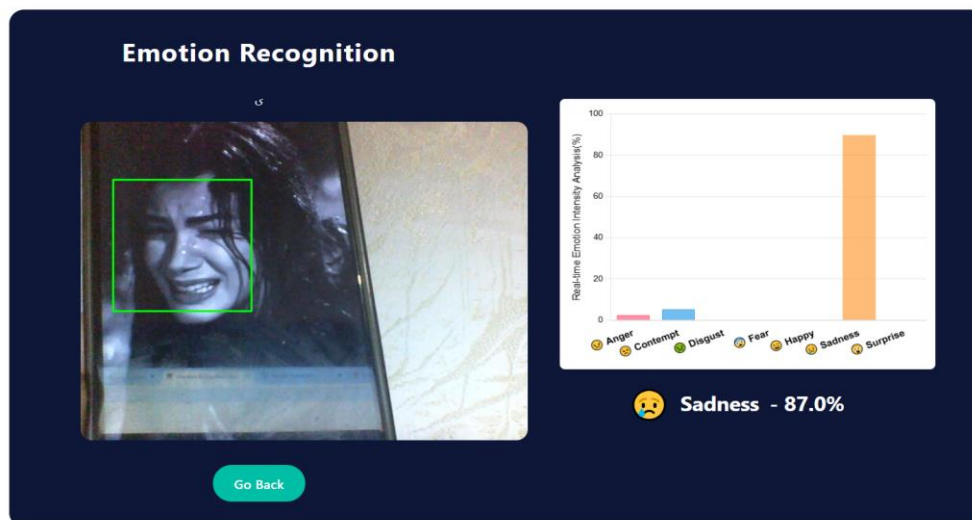


Figure 36: Real-time Emotion Detection via Webcam

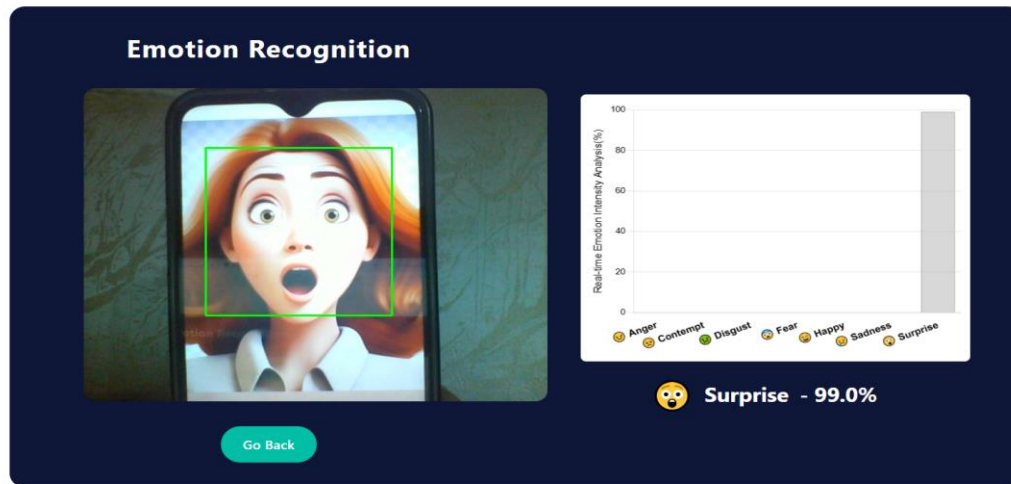


Figure 37 : Another Example for Real-time Emotion Detection

If no face is detected, the application will generate the message

“No Face is detected.”

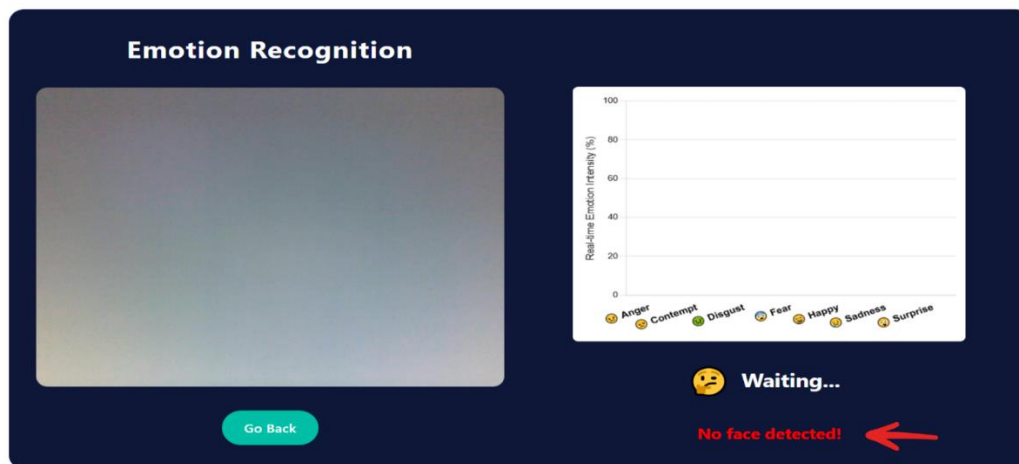


Figure 38 : Notification When No Face Is Found

5.5 Model Evaluation

This interface was developed for the user to test the trained model's performance with their own external dataset. The user should upload a data file (ZIP format), and the accuracy of the model will be calculated and displayed interactively.

Some clear recommendations are provided, depending on the accuracy reached:

- **Less than 50%:** "The accuracy is so poor. It is recommended that the user changes the

data or improves it."

- **Between 50% and 80%:** "The model gives good results-theoretically it could be better if the data were of better quality."
- **Over Than 80%:** "The model is very well performing, and the data can be considered perfect for use with characteristics contained therein."

This evaluation guide the user in looking at the efficiency of the model on the data and to some extent for real use.

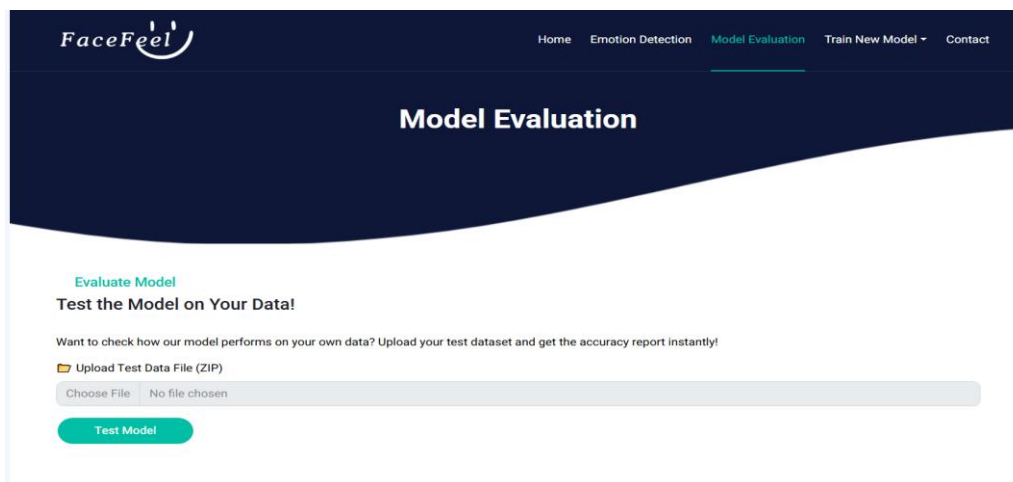


Figure 39 : Model Evaluation Interface of FaceFeel Application

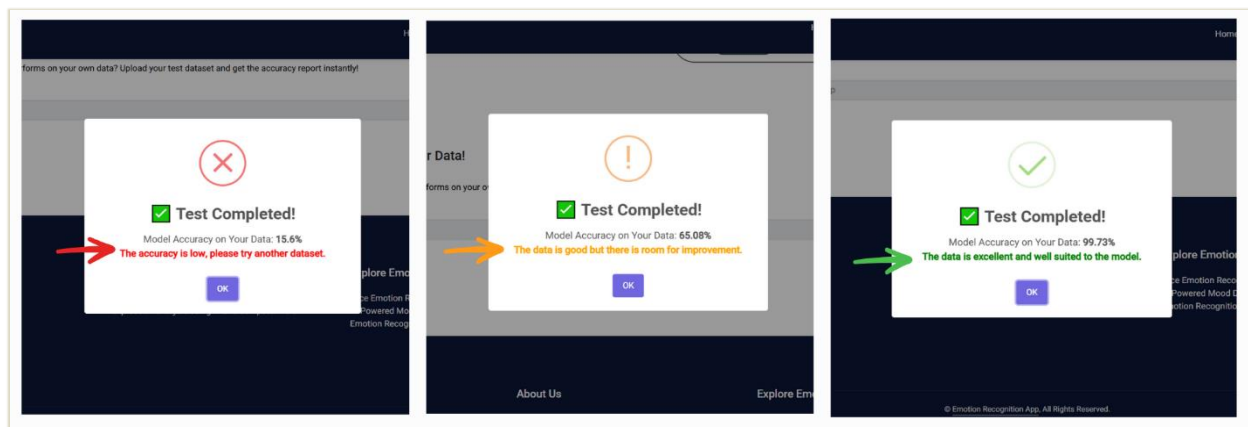


Figure 40: Model Accuracy Recommendations

5.6 Train a New Model

This section allows the user to train a new model using custom data. The system offers two options in a dropdown menu:

- ***Train on RGB Data:*** To train the model on colored images (using the Red, Green, and Blue channels).
- ***Train on Grayscale Data:*** To train the model on grayscale images (using shades of gray only).

5.6.1 Train on RGB/Grayscale Data Interfaces

In this section, the application allows users to train, evaluate, and make predictions on any custom dataset by creating their own emotion recognition model — whether in grayscale or RGB — through three sequential stages: (train, test, and prediction)

The workflow is the same for both grayscale and RGB images; the only difference lies in the type of images processed.

- **Train Section:** In this stage, the application caters to training a new recognition model for emotions on the user's custom dataset. The interface allows the user to configure:
- **Dataset Folder:** Upload the labeled data set (ZIP format) arranged by class folders.
- Define some parameters such as number of epochs, image size, batch size.

The user hits the Start Training button to start the training process. The progress bar increases by 10% after every epoch, giving the visual feedback of the ongoing training process.

A warning message appears:

During training, do not refresh or navigate away until completion.

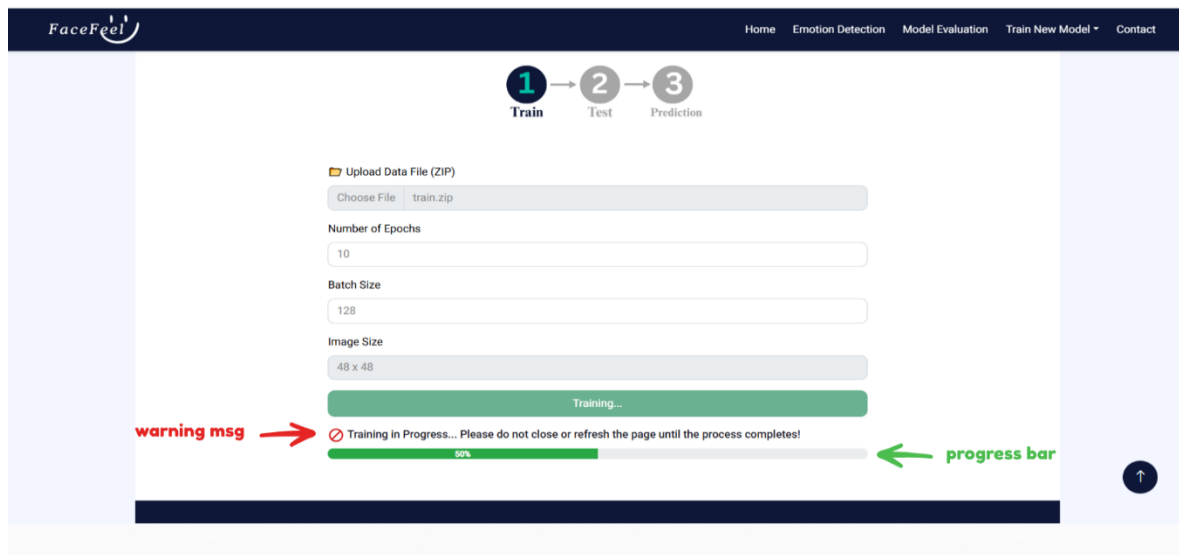


Figure 41: Model Training Progress with warning msg

After finishing the training stages, the application displays:

- Final Training Accuracy
- Loss and Accuracy Curves.
- Number of classes.
- Download Model button to download the trained model weights.
- **Go to Test** button that redirects the user directly to the evaluation phase interface to assess the trained model's performance on new, unseen data.

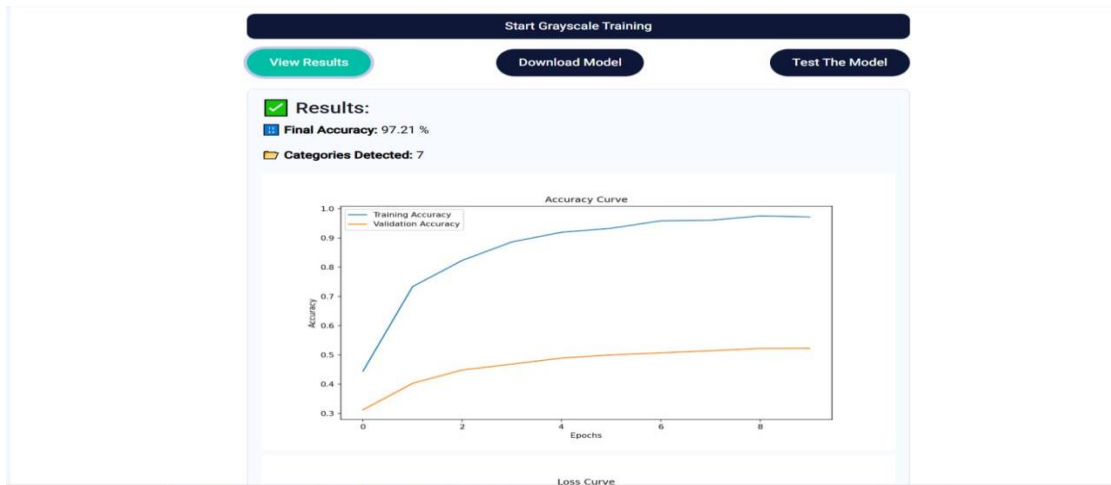


Figure 42: Training Results in Grayscale Model

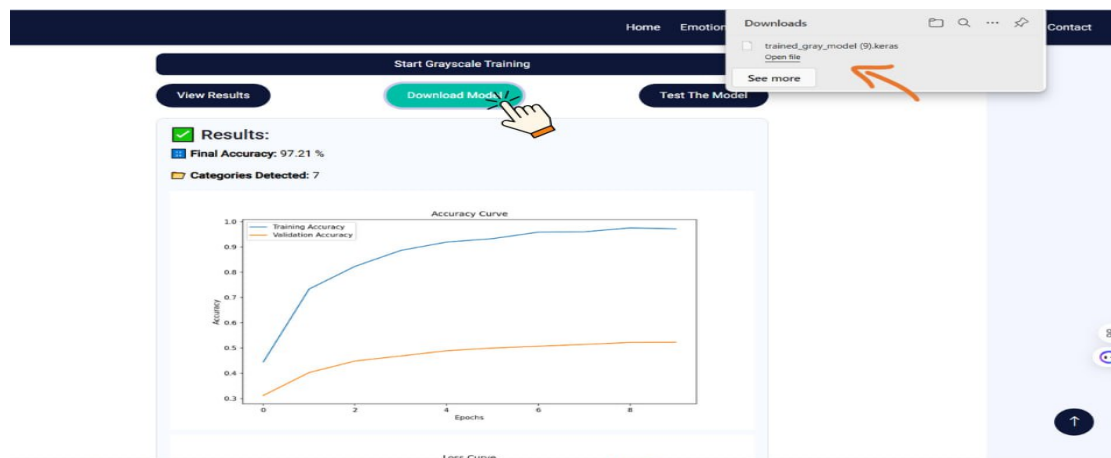


Figure 43: Downloading the Pre-trained Model

Testing Section:

After completing training, the user can check how well their model carries out classification by:

- Clicking Test Model to open up the Evaluation window.
- Uploading a labeled test dataset on (ZIP format).
- Clicking “Test Grayscale Model” to begin the testing.

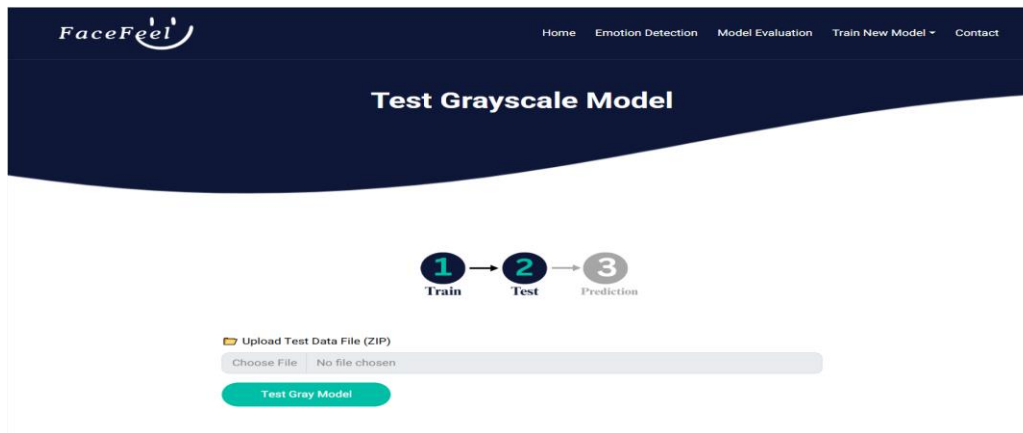


Figure 44 : Testing Grayscale Model Interface

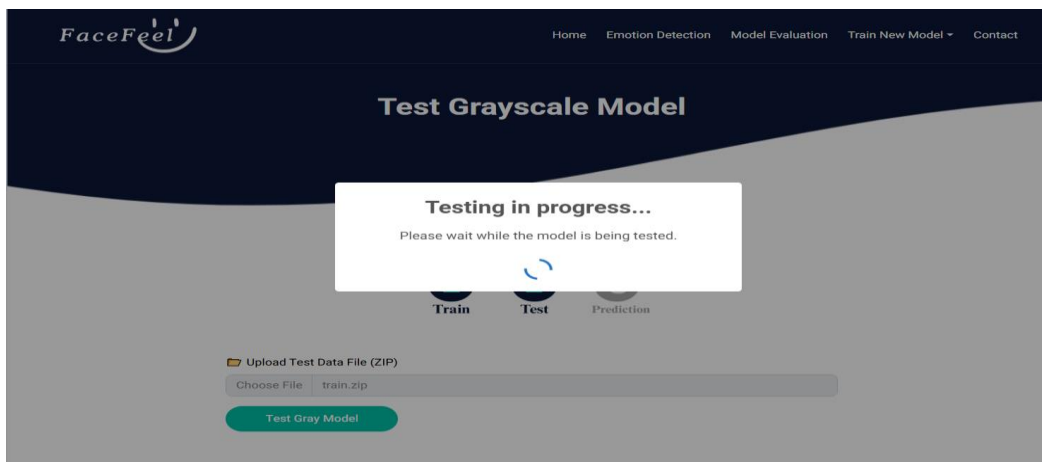


Figure 45: Model Being Tested' message window

The system will then display:

- Test Accuracy
- Classification Report (Precision, Recall, F1-Score)
- Confusion Matrix
- There is also a "Go to Prediction" button that takes you directly to real-time emotion detection.

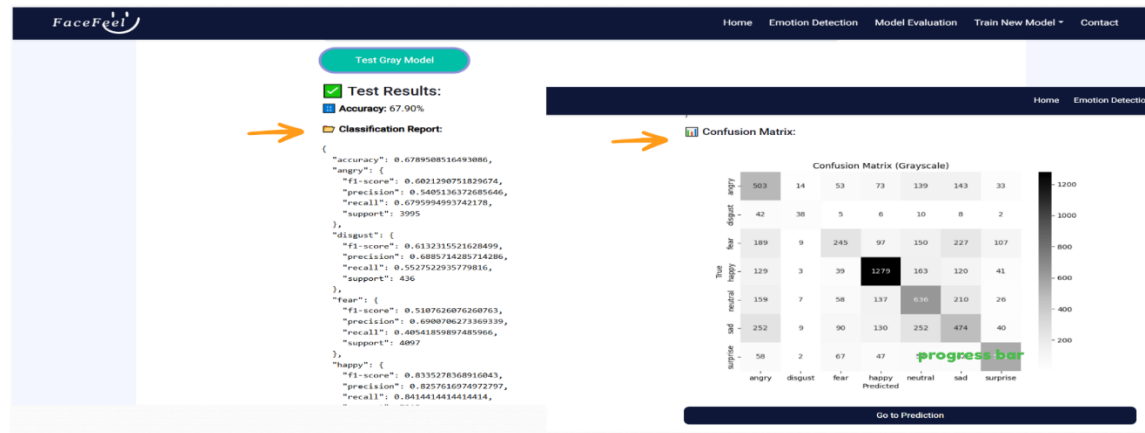


Figure 46 : Model Testing Results in Grayscale Model

Prediction Section :

In this phase, users can test their trained model on single or multiple images by:

- Uploading images (grayscale or RGB, just like the training input in size and preprocessing).
- Clicking Predict to obtain the prediction results.

For each image, the system displays:

- Prediction of the emotion label (happy, sad, angry, etc.)
- Corresponding emoji
- The confidence score for the result
- The emotion description text, which is a layman's term description of the detected emotional state
- This gives immediate and clear feedback regarding the emotions detected in the uploaded images.

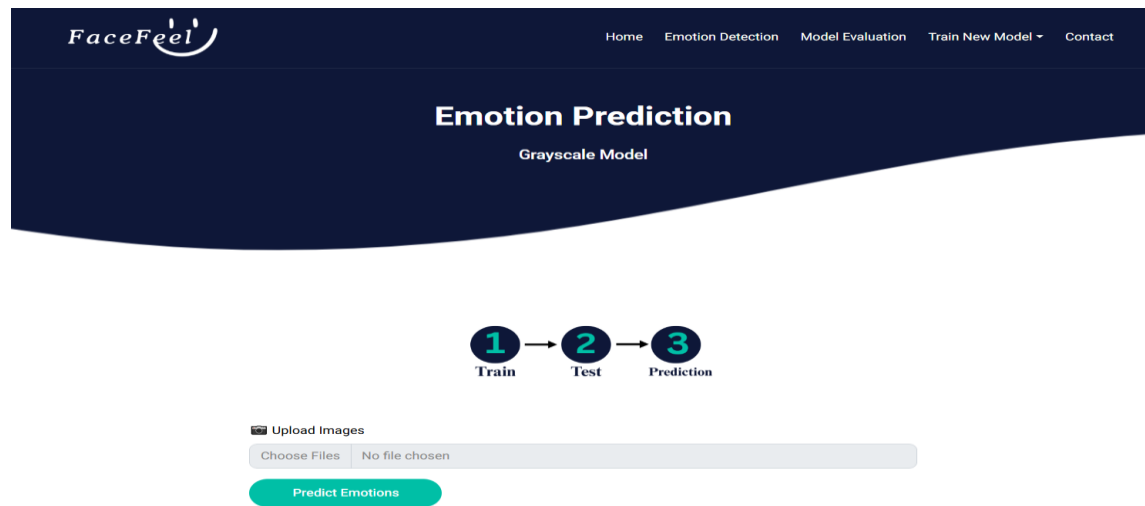


Figure 47 : Emotion Prediction Grayscale Interface

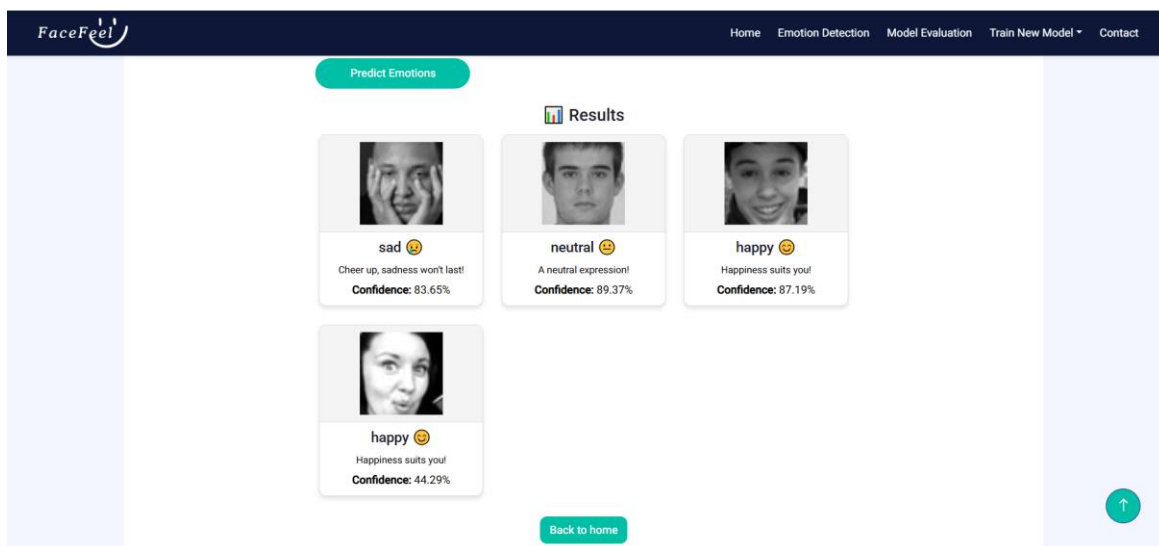


Figure 48: Emotion Prediction Result Grayscale Interface

5.7 Clarification of the Unified Training and Evaluation Process for Models

The explanation given above applies generally to both grayscale and RGB models, as the application's workflow for training, testing, and prediction is essentially the same for both.

However, the detailed results presented here correspond specifically to the grayscale model.

The RGB model follows the same steps, with the main difference being the input data format (color images with three channels instead of single-channel grayscale images).

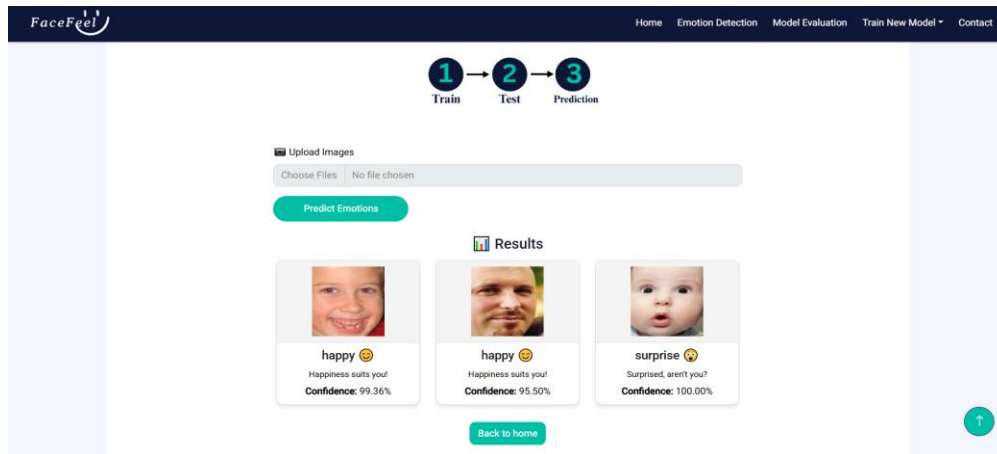


Figure 49 :Example of RGB Prediction in the RGB Prediction Interface

6 Conclusion

In conclusion, this chapter detailed the essential functionalities of the developed face emotion system, from data input and model training to evaluation and affective prediction. A complete examination of the user interfaces and their outputs has been carried out, with a particular concentration on the use of both pre-trained and fine-tuned models. As the system supports both grayscale and RGB images, it ensures unified and consistent workflow and output formats, leading to enhanced efficiency and effectiveness in facial emotion recognition. This implementation also points toward how the application can cater to varying real-world scenarios while providing a truly intuitive and flexible user interface.

General Conclusion

This thesis explored the design and the implementation of a facial emotion recognition system using Convolutional Neural Networks CNNs. The system was trained on grayscale facial images and achieved high performance in emotion classification tasks, as demonstrated through accuracy metrics, confusion matrix analysis, and classification reports.

Through a well-structured methodology that combined proper data preprocessing, model optimization, and real-time prediction mechanisms, the system was able to generalize well to unseen data. A user-friendly application was also developed to enable emotion detection on both pre-recorded images and real-time camera feeds, extending the practical utility of the proposed solution.

Despite the strong results, several challenges remain, such as dealing with variations in facial structure, lighting, and emotion ambiguity. Future work may involve incorporating multimodal emotion recognition, leveraging larger and more diverse datasets, and deploying the system on embedded platforms for wider accessibility.

In summary, this project contributes to the field of affective computing by offering an effective and interactive system for facial emotion recognition and paves the way for future innovations in intelligent human-computer interaction.

7Bibliographie

- [1] «Facial Recognition Technology: Federal Law Enforcement Agencies Should Have Better Awareness of Systems Used By Employees,» Government Accountability Office (GAO), 09 2023. [En ligne]. Available: <https://www.gao.gov/products/gao-24-107372>.
- [2] W. Richard E and G. Rafael C, Digital Image Processing, Pearson Education, 2018.
- [3] .Near East University ‘ECC419 Image Processing – Lecture Notes ‘Ş. Boran
- [4] J. Anil K, Fundamentals of Digital Image Processing, Prentice-Hall, 1989.
- [5] R. John C, Introduction to Image Processing and Analysis, CRC Press, 2016.
- [6] L. G et D. Y, «Deep Learning for Emotion Recognition: A Review,» *IEEE Transactions on Affective Computing*, 2020.
- [7] L. M, D. H et D. Z, «Facial Expression Recognition: A Comprehensive Review,» *IEEE Transactions on Affective Computing*, 2018.
- [8] J. M. E, E. M. M et N. S, «Speech Emotion Recognition: Features and Classification Models,» *IEEE Transactions on Affective Computing*, 2018.
- [9] M. M. Seif et B. M. Felip, «A Survey on Emotion Recognition from Text,» *ACM Computing Surveys*, 2017.
- [10] K. Y, M. C, S. M et S. L. J, «Emotion Recognition Based on Physiological Signals: A Review,» *IEEE Transactions on Affective Computing*, 2012.
- [11] R. Nawaz, S. Anwar et S. Zafeiriou, «Body Movement-Based Emotion Recognition Using Deep Learning: A Review,» *IEEE Transactions on Affective Computing*, 2021.
- [12] Z. Zeng, M. Pantic, G. Roisman et H. S. T, «Automatic Facial Expression Recognition: A Survey,» *Image and Vision Computing*, 2009.
- [13] C. A. Corneanu, M. O. Simón, J. F. Cohn et S. E. Guerrero, «A Survey on Facial Expression Recognition Systems,» *IEEE Transactions on Affective Computing*, 2016.
- [14] M. M. Tom , Machine Learning, McGraw-Hill, 1997.
- [15] . N. E. H. ADOUANE, «Traffic signs recognition for autonomous cars». 2019/2020.
- [16] Gareth James, Daniela Witten, Trevor Hastie et Robert Tibshirani, An Introduction to Statistical Learning: with Applications in R, Springer, 2013.
- [17] O. Richard, P. Duda, H. E et G. S. David, Pattern Classification, Wiley-Interscience, 2001.
- [18] Pang-Ning Tan, Michael Steinbach et Vipin Kumar, Introduction to Data Mining, Pearson, 2006.
- [20] Ian Goodfellow, Yoshua Bengio et Aaron Courville, Deep Learning, MIT Press, 2016.
- [21] S. Hochreiter et J. Schmidhuber, «Long Short-Term Memory,» *Neural Computation*, vol. Volume 9, n° %1Issue 8, p. 1735–1780, 1997.
- [22] R. Socher, C. C. Lin, C. Manning et A. Ng, Écrivains, *Parsing Natural Scenes and Natural Language with Recursive Neural Networks*. [Performance]. Proceedings of the 28th International Conference on Machine Learning (ICML), 2011.
- [23] G. E. Hinton, S. Osindero et Y. W. Teh, «A fast learning algorithm for deep belief nets,» *Neural Computation*, n° %17(18), p. 1527–1554, 2006.
- [24] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. Setio et F. Ciompi, «A survey on deep learning in medical image analysis,» *Medical image analysis*, n° %142, pp. 60-88, 2017.
- [25] François Chollet, «Chapter 6 – Deep learning for computer vision,» chez *Deep Learning with Python*, Manning Publications.
- [26] François Chollet, «Chapter 5 – Machine-learning fundamentals,» chez *Deep Learning with Python* ,

Manning Publications, 2018.

- [27] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media 2nd Edition, 2019.
- [28] Connor Shorten et K. Taghi M, Interviewees, *A survey on Image Data Augmentation for Deep Learning*. [Interview]. 2019.
- [29] S. Ruder, «An Overview of Gradient Descent Optimization Algorithms,» 2016.
- [30] L. S et D. W, « Facial Expression Recognition with CNNs: A Survey and Experiments,» 2020.
- [31] T. M, Q. A et A. S, Interviewees, *Deep Learning for Facial Emotion Recognition: A Review*. [Interview]. 2021.
- [32] P. J. Team, «JupyterLab: Next-generation user interface for Project Jupyter,» 2020. [En ligne]. Available: <https://jupyterlab.readthedocs.io/en/stable/>.
- [33] M. Corporation, «Visual Studio Code Documentation,» 2024. [En ligne].
- [34] "GeeksforGeeks," 16 05 2025. [Online]. Available: <https://www.geeksforgeeks.org/python-introduction-to-web-development-using-flask/>.

ملخص: تهدف هذه المذكرة إلى تصميم وتنفيذ نظام للتعرف على المشاعر البشرية اعتمادًا على تعابير الوجه، باستخدام تقنيات الذكاء الاصطناعي، وبشكل خاص الشبكات العصبية الالتفافية (CNN). يُعتبر التعرف على المشاعر مجالًا ناشئًا و هامًا ضمن الحوسبة العاطفية (Affective Computing) ، وله تطبيقات واسعة تشمل الصحة النفسية، التعليم، التسويق، أنظمة المراقبة، وتفاعل الإنسان مع الآلة.

تم في هذا العمل استخدام مجموعة بيانات للوجوه تتضمن مشاعر مختلفة (مثل الغضب، الفرح، الحزن، الدهشة...)، حيث خضعت الصور لعمليات معالجة أولية مثل تحويلها إلى تدرج رمادي وتعديل الحجم. ثم تم تدريب نموذج CNN باستخدام بيئة JupyterLab و Google Colab، مع استخدام أدوات مثل TensorFlow و Keras لتصميم النموذج وتقييمه. شمل النظام مراحل أساسية هي:

• اكتشاف الوجه (Face Detection)

• استخراج الملامح (Feature Extraction)

• تصنيف المشاعر (Emotion Classification)

وقد أظهرت النتائج دقة جيدة في التعرف على المشاعر، مما يؤكد فعالية النموذج المقترح. كما تم بناء واجهة تطبيق مبسطة لاختبار النموذج على صور حية أو مخزنة، مما يُقرب هذا المشروع من التطبيقات العملية الواقعية.

ورغم تحقيق نتائج مشجعة، إلا أن بعض التحديات مثل الإضاءة، اختلاف زوايا الوجه، وتشابه تعابير المشاعر لا تزال قائمة، مما يفتح المجال لأعمال مستقبلية أكثر تطورًا، مثل دمج إشارات متعددة (صوت، نص، تعابير وجه)، أو اعتماد نماذج أقوى مدعومة ببيانات أكثر تنوعًا.

الكلمات المفتاحية: الذكاء الاصطناعي، التعلم العميق، الشبكات العصبية الالتفافية (CNN)، التعرف على المشاعر، التعرف على الوجوه، تحليل المشاعر

Abstract:

This thesis aims to design and implement a system for recognizing human emotions based on facial expressions, using artificial intelligence techniques, particularly Convolutional Neural Networks (CNN). Emotion recognition is an emerging and important field within affective computing, with broad applications in mental health, education, marketing, surveillance systems, and human-computer interaction.

In this work, a facial image dataset containing various emotions (such as anger, happiness, sadness, surprise...) was used. The images underwent preprocessing steps such as grayscale conversion and resizing. The CNN model was then trained using environments like JupyterLab and Google Colab, with tools such as TensorFlow and Keras used for model design and evaluation.

The system consisted of the following main stages:

- Face Detection
- Feature Extraction
- Emotion Classification

The results showed good accuracy in emotion recognition, confirming the effectiveness of the proposed model. A simple application interface was also developed to test the model on both live and stored images, bringing the project closer to real-world applications.

Despite the promising results, some challenges remain, such as lighting conditions, facial angle variations, and similarities between emotional expressions. This opens the door for more advanced future work, such as integrating multiple modalities (voice, text, facial expression) or adopting more powerful models trained on more diverse datasets.