Carthage University
National Institute of Applied Sciences and Technologies

PROFESSIONAL PERSONAL PROJECT REPORT

Title : Facial Emotions Detection

Authors :

Mr. Abderrazek ABID   Mr. Amenallah BOUHACHEM
Mr. Walid BRINI   Mr. Helmi BALHOUDI

Supervisor :

Mr. Mourad MOUSSA

Jury :

Mr. Mohammed Ali HAMDI

Academic year : 2022 /2023

Appreciations and signatures of the supervisors

Acknowledgments

This work could only be carried out with the help and support of many individuals whom we love and respect profoundly. With these few words, we would like to take the opportunity to express our sincere appreciation and gratitude to all the academic staff of **INSAT** and the Department of Physical Engineering and Instrumentation, for their valuable efforts, help and support throughout the year.

In particular, we would like to pay tribute to Mr.**Mourad Moussa**, for his devotion, guidance, assistance, direction and especially his encouragement throughout this project. His relentless efforts, combined with our determination and hard work, allowed us to complete this project to the highest standards. He was constantly present when we needed him, prompt to answer any questions, and extremely valuable in explaining and tutoring us in unclear concepts.

We would also like to acknowledge the contributions of each of us, whose hard work and cooperation were essential to the success of the project. Our commitment to excellence and to teamwork was evident at all phases of the project.

In conclusion, we are very proud of what we have accomplished together and the progress we have made, and we are confident that this project will be a testament to the depth of dedication and proficiency of our team. We sincerely thank all those who have helped us, believed in us and supported us. Finally, we cannot forget to mention our dear colleagues who have been true supporters throughout the project and have been gracious enough to advise and encourage us.

We have learned a lot during this journey and, above all, we are aware that there is still much to learn, to know and to research. This project is merely a first step and there is still much to do in the future, while developing our skills and knowledge.

Abstract

This project developed an emotional face recognition system that uses machine learning and AI to accurately detect and interpret human emotions from facial expressions. The system was tested and found to be promising, with the potential to be used in a variety of applications.

Key Highlights:

- Theoretical study of ML & AI for emotional face recognition.
- Collection and preprocessing of relevant facial image dataset.
- Extraction of meaningful facial features using advanced algorithms.
- Training of machine learning models, including CNNs.
- Rigorous evaluation and comparison with existing approaches.
- Acknowledgement of limitations and scope for future enhancements.

Contents

Introduction 1

1 Theoretical Analysis 4

1 Machine learning and AI . . . 5
2 Computer Vision . . . 6
3 Facial Emotion Recognition . . . 6
4 Application Fields Of Facial Emotion Recognition . . . 8
5 Existing Approaches . . . 8
5.1 Feature-based approaches . . . 8
5.2 Facial Action Coding System (FACS) . . . 8
5.3 Landmarks . . . 9
6 Neural Network NN . . . 10
7 Convolutional neural network CNN . . . 12
7.1 Convolution layer . . . 12
7.2 Sub-sampling layer (pooling) . . . 13
8 Deep Convolutional neural networks DCNN . . . 13
9 Classification of Emotions . . . 14
9.1 Workflow . . . 14
9.2 Data Sets . . . 15
9.2.1 Train Data : . . . 15
9.2.2 Test Data : . . . 15
10 Problem Behind the Theoretical Analysis . . . 15
10.1 Poses variation . . . 15
10.2 lighting Change . . . 16

2 Needs analysis and specification **18**

1 Actors 18

2 Analyse of needs 19

2.1 Functional needs 19

2.2 Non Functional needs 19

3 Specification of needs 19

3.1 Modeling language 19

3.2 User case diagram 19

3.3 Sequence Diagram 20

3.4 Activity Diagram 21

3 Implementation **23**

1 Tools And Libraries Used 24

2 Training the model On the dataset FER-2013 25

2.1 Overview 25

2.2 Import & Manipulate the data set 26

2.3 Data Preprossessing 28

2.4 Creating the model and Compiling it 28

2.5 Data augmentation prior to training 31

2.6 Model Training 31

2.7 Results 32

2.8 Model Testing on Evaluation data 34

3 Interpretation 35

3.1 Visualizing the CNN 35

3.2 Limitations 36

Conclusion and perspectives **40**

List of Figures

1.1 Computer Vision . . . 6
1.2 Facial emotion recognition . . . 7
1.3 Action Units corresponding to different movements in face . . . 9
1.4 Landmarks on face . . . 10
1.5 Neural Network Architecture . . . 11
1.6 Diagram of the operation of a perceptron . . . 11
1.7 a typical CNN . . . 12
1.8 Convolution product . . . 12
1.9 Sub-Sampling Method . . . 13
1.10 Exemple of design of a DCNN . . . 14
1.11 WorkFlow of AI algorithm . . . 14
1.12 Data Distribution . . . 15
1.13 Change of Angles . . . 16
1.14 Lighting change . . . 16

2.1 User case Diagram . . . 20
2.2 Sequence diagram . . . 20
2.3 Activity diagram . . . 21

3.1 Dataset in CSV format . . . 26
3.2 Distribution of emotion categories in the dataset . . . 26
3.3 Some Images from the dataset . . . 27
3.4 Algorithm part 1 . . . 29
3.5 Algorithm part 2 . . . 30
3.6 Training the model and viewing its progress . . . 32

3.7 Accuracy evolution for train and validation data 32
3.8 Loss evolution for 100 epoch 33
3.9 Confusion Matrix 33
3.10 Confusion Matrix Heatmap 34
3.11 Testing the model on examples from the test set 34
3.12 Visualization of different convolutional layers 35
3.13 Accuracy evolution for 7 classes 36
3.14 Loss evolution for 7 classes 36

List of Tables

3.1 Parameters for Data Augmentation 31

List of acronyms

- **AI**. Artificial Intelligence
- **CV**. Computer Vision
- **FACS**. Facial Action Coding System
- **AU**. Action Units
- **ML**. Machine Learning
- **NN**. Neural Network
- **CNN**. Convolutional neural network
- **DCNN**. Deep Convolutional neural networks
- **UML**. Unified Modeling Language
- **CSV**. Comma-separated-values
- **LR**. learning rate
- **FER**. Facial Emotional Recognition

Introduction

Emotional face recognition is an intriguing field that combines computer vision and machine learning techniques to decipher human emotions through facial expressions. The ability to accurately detect and interpret emotions from facial cues has significant implications across various domains, including psychology, human-computer interaction, and social robotics. By enabling machines to understand and respond to human emotions, emotional face recognition opens up avenues for improved user experiences, personalized services, and enhanced human-machine interactions.

Recognizing emotions from facial expressions is a fundamental aspect of human communication and understanding. Humans are inherently skilled at interpreting subtle changes in facial muscle movements, which convey a wide range of emotions, including happiness, sadness, anger, fear, surprise, and disgust. However, developing automated systems that can accurately perceive and categorize these emotions remains a complex and challenging task.

This project aims to explore the realm of emotional face recognition by leveraging state-of-the-art computer vision techniques and machine learning algorithms. By analyzing facial images and extracting meaningful features, we seek to develop a robust system capable of accurately identifying and categorizing human emotions.

The potential applications of emotional face recognition are numerous. In the field of psychology, such systems can contribute to the assessment and diagnosis of emotional disorders, aiding therapists and clinicians in their understanding of patients' emotional states. In human-computer interaction, emotional face recognition can enhance user experiences by allowing machines to adapt their behavior or responses based on the user's emotional state, leading to more personalized and engaging interactions.

Moreover, emotional face recognition holds immense value in social robotics, where machines are designed to interact with humans in a more natural and empathetic manner. Robots equipped with emotional understanding can perceive and respond appropriately to human emotions, leading to improved companionship, caregiving, and assistance for individuals in need.

Through this project, we aim to contribute to the growing body of research in emotional face recognition, advancing the state of the art and uncovering new insights into the fascinating world of human emotions. By developing an accurate and reliable system, we aspire to bridge the gap between human emotional understanding and machines, unlocking novel opportunities for a wide range of applications.

About Emotion

Emotions can be analyzed from expressions and other parts of the face Body. We see changes when someone is very angry, sad, scared or surprised Recognize the expression of a specific facial image and try to distinguish emotions. Swings in mood by different facial images help understand how this works Some people feel that there is no non-verbal communication. this is sometimes debated Non-verbal emotional communication is more important and effective than verbal communication communicate. This applies to interpersonal communication where one or both Individuals cannot communicate verbally because they speak first, as in Not being able to talk to each other with young kids or deaf people or at all.

In this project, we will focus on the recognition and classification of emotions into seven distinct classes: **Anger, Disgust, Fear, Happiness, Sadness, Surprise, and Neutral**. Each of these classes represents a specific emotional state that can be expressed through facial expressions.

Anger reflects feelings of irritation, hostility, or frustration, often accompanied by facial cues such as narrowed eyebrows and a tense jawline. **Disgust** represents strong aversion or revulsion, often accompanied by facial expressions such as a wrinkled nose or a curled upper lip. **Fear** signifies a state of anxiety or apprehension, often exhibited through widened eyes and raised eyebrows.

Happiness encompasses feelings of joy, contentment, or pleasure, characterized by a smiling face with raised cheeks and crinkled eyes. **Sadness** reflects feelings of sorrow, melancholy, or grief, often displayed through a downturned mouth, drooping eyebrows, and tearful eyes. **Surprise** represents a sudden reaction to unexpected events, typically indicated by widened eyes and raised eyebrows.

Finally, the neutral class signifies a lack of any overt emotional expression, presenting a neutral or impassive facial appearance. By focusing on these seven emotion classes, we aim to develop a comprehensive emotional face recognition system capable of accurately identifying and categorizing a wide range of human emotions.

1

Theoretical Analysis

Introduction

The theoretical analysis chapter of this project aims to provide a comprehensive exploration of the underlying principles, theories, and concepts relevant to emotional face recognition. This analysis serves as a foundation for understanding the theoretical frameworks that guide the development and implementation of an accurate and efficient emotional face recognition system. By delving into the various aspects of facial expression recognition, computer vision, machine learning, datasets, and ethical considerations, this chapter aims to establish a strong theoretical basis for the subsequent stages of the project. Subsequently, we will delve into the theoretical aspects of machine learning and deep learning algorithms. Understanding the principles behind these algorithms, such as convolutional neural networks (CNNs) , will provide insights into how they learn to recognize patterns and make predictions based on training data. We will also explore the theoretical foundations of the datasets used for training and evaluation, including their properties, annotation processes, and evaluation metrics.

1 Machine learning and AI

Machine learning and artificial intelligence (AI) have emerged as transformative technologies with the potential to revolutionize various aspects of our lives. These fields encompass a wide range of algorithms, models, and techniques that enable computers to learn from data, make intelligent decisions, and perform tasks that traditionally required human intelligence. In the context of emotional face recognition, machine learning and AI play a crucial role in deciphering and interpreting human emotions from facial expressions.

The fusion of machine learning and AI techniques with emotional face recognition has the potential to impact various domains. In psychology, these technologies can aid therapists and researchers in understanding emotional disorders, facilitating more accurate assessments and diagnoses. In human-computer interaction, intelligent systems equipped with emotional understanding can provide more personalized and engaging experiences, adapting their behavior to match the user's emotional state. Furthermore, in social robotics, machines capable of perceiving and responding to human emotions can enhance companionship, caregiving, and assistance for individuals.

This project aims to explore the intersection of machine learning, AI, and emotional face recognition, with the goal of developing a robust system that can accurately detect and interpret human emotions from facial expressions. By harnessing the power of advanced algorithms and models, we seek to contribute to the growing body of knowledge in this field and unlock new possibilities for understanding and interacting with human emotions.

In the following sections, we will delve into the theoretical foundations of machine learning and AI, discuss the methodologies employed in our project, present the dataset used for training and evaluation, describe the models and algorithms implemented, and analyze the results obtained. Through this endeavor, we hope to advance the state of the art in emotional face recognition, paving the way for future advancements and applications in this exciting field.

2 Computer Vision

The term computer vision refers to the various techniques that allow computers to see and understand the content of images. It is a subcategory of Artificial Intelligence and Machine Learning.

The field of Computer Vision brings together multiple techniques from various fields of engineering or computer science. Generally speaking, the various methods aim to reproduce human vision.

Like any artificial intelligence technology, the basis of computer vision is data. The system must train itself to discern and recognize images on a large amount of data. Computer vision subfields include event detection, video tracking, object recognition, learning,indexing, motion estimation, 3D scene modeling, and image restoration. picture.

Figure 1.1: Computer Vision

3 Facial Emotion Recognition

Facial emotion recognition is a technology that analyzes feelings from various sources such as images and videos. It belongs to the family of technologies commonly referred to as "affective computing," a multidisciplinary field of research that investigates the ability of computers to recognize and interpret human emotions and emotional states, often built on top of artificial intelligence techniques. Facial expressions are a form of nonverbal communication that can provide cues about human emotions. Decoding such emotional expressions has been of research interest in psychology for decades, as well as in the field of human-computer interaction. More recently, the ubiquity of cameras and technological advances in biometric analysis, machine learning, and pattern recognition have played an important role in the development of FER technology. Emotion detection is based on analyzing the location of facial landmarks (e.g. nose tip, eyebrows). In addition, these changes in position are analyzed in the video to identify the contraction of a set of facial muscles. Based on algorithms, facial expressions can be broken down into basic emotions (e.g., anger, disgust, fear, joy, sadness, and surprise) or complex emotions (e.g., happy, sad, happy, surprised, happy, disgusted, sad, fearful, sad, angry), sad surprise). In other cases, facial expressions may be related to physical or psychological states such as tiredness or boredom.

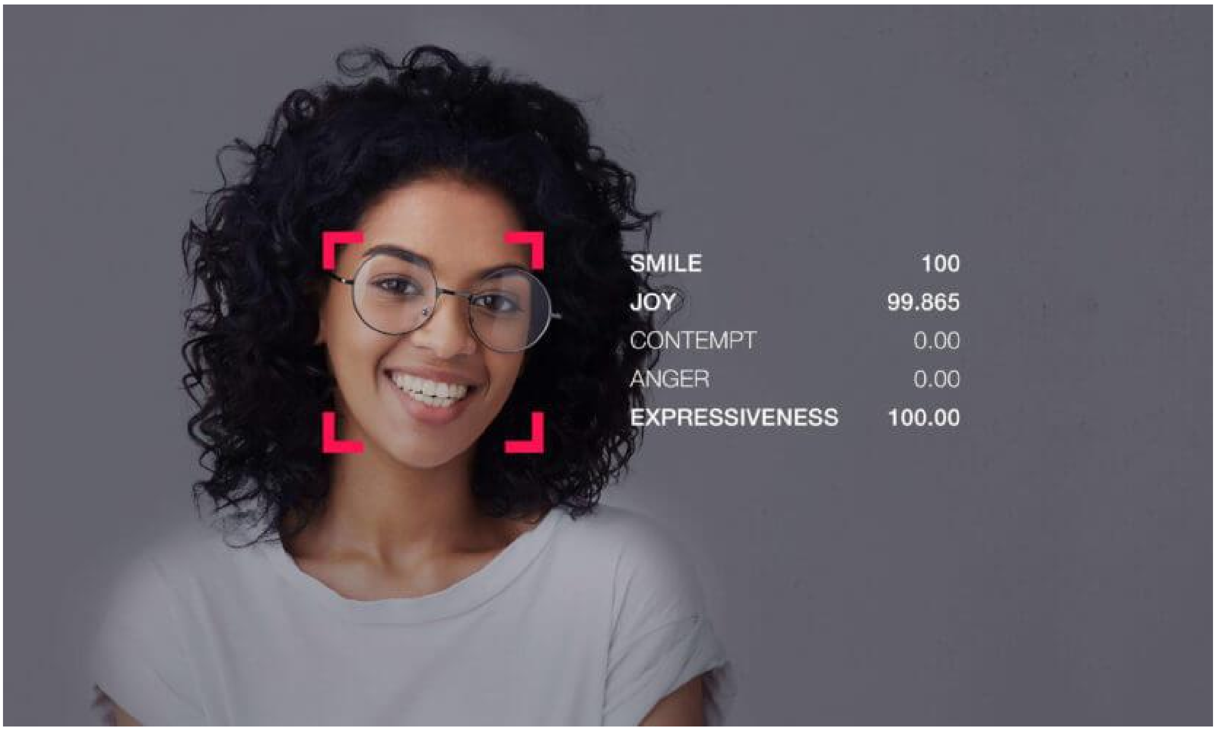

Figure 1.2: Facial emotion recognition

4 Application Fields Of Facial Emotion Recognition

Facial emotion recognition has applications in various industries, such as retail. “Based on our current customer and partner engagement, marketing/advertising is one of the most promising use cases. Other industries that could benefit from this technology include: Teleconsultations, health predictions: AI models used in social services, hospitals and medical facilities can spot signs of stroke and epilepsy. It can also be used as a tool in occupational therapy to predict health status and identify signs of symptoms of depression, anxiety, and/or non-motor phases of cognitive decline. Police emotion analysis: In terms of security, facial emotion detection can provide real-time information on crowd emotions, emergency control, and risk management. Facial emotion recognition can identify interesting people/suspicious characters and provide early warning signs, which helps to detect associations of aggressive groups and their potential for violent behavior.

5 Existing Approaches

5.1 Feature-based approaches

In the context of machine learning and pattern recognition, feature-based methods are methods for extracting relevant features from data to represent and characterize the underlying patterns or structures. These methods are commonly used in a variety of fields, including image recognition, natural language processing, and speech recognition.

With feature-based methods, the main goal is to identify a set of informative features that can effectively capture the essential characteristics of the data. These features are selected or developed based on prior knowledge of the problem domain or through exploratory analysis of the data. After features are extracted, they serve as input to a learning algorithm that builds a model based on these features.

5.2 Facial Action Coding System (FACS)

The Facial Action Coding System (FACS) is a comprehensive systematic approach to objectively describe facial expressions. It was developed by psychologists Paul Ekman and Wallace V. Friesen in the 1970s as a tool for analyzing and classifying human facial movements.

FACS breaks down facial expressions into specific units, so-called Action Units (AUs). Each AU represents a specific facial muscle or muscle group involved in producing a specific movement or expression. 46 AUs were identified in the FACS, each assigned a unique numerical code.

By using FACS, researchers and psychologists can analyze and describe facial expressions in a standardized manner. FACS can be used to study emotional expression, non-verbal communication, facial expressions, and other facial behavior. It is used in a wide variety of fields, including psychology, neuroscience, computer science, and animation.

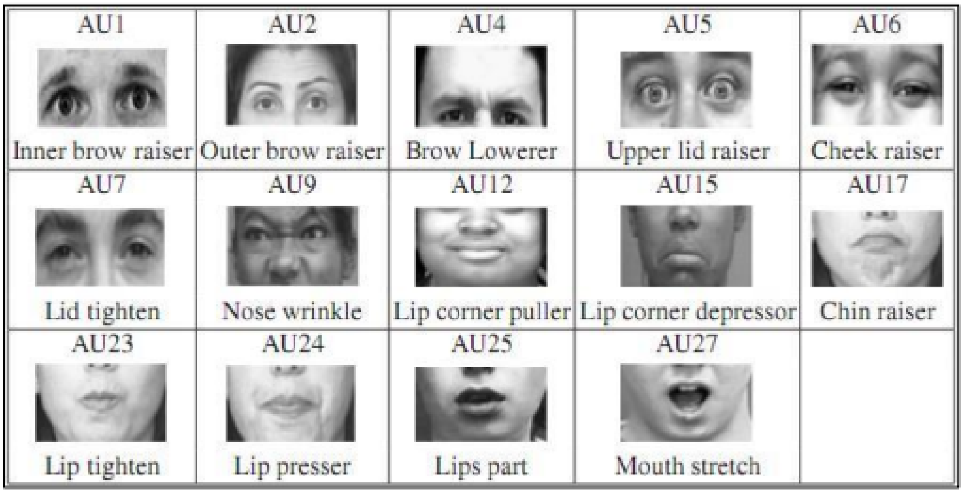

Figure 1.3: Action Units corresponding to different movements in face

5.3 Landmarks

Landmarks on the face are very crucial and can be used for face detection and recognition. The same landmarks can also be used in the case of expressions. The Dlib library has a 68 facial landmark detector which gives the position of 68 landmarks on the face.

Figure 2 shows all the 68 landmarks on face. Using dlib library we can extract the coordinates(x,y) of each of the facial points. These 68 points can be divided into specific areas like left eye, right eye, left eyebrow, right eyebrow, mouth, nose and jaw.

Figure 1.4: Landmarks on face

6 Neural Network NN

The artificial neural network is a machine learning algorithm introduced by Rumelhart, Hinton and Williams in 1985 based on the model of biological neural networks in which several neurons work together to generate information from several received information. The fully connected neural network consists of at least three layers as shown in the figure. The first layer represents the data layer, the last layer represents the output layer and the intermediate layers also called hidden layer is made up of several perceptions which take as inputs the outputs of the layer which precedes it.

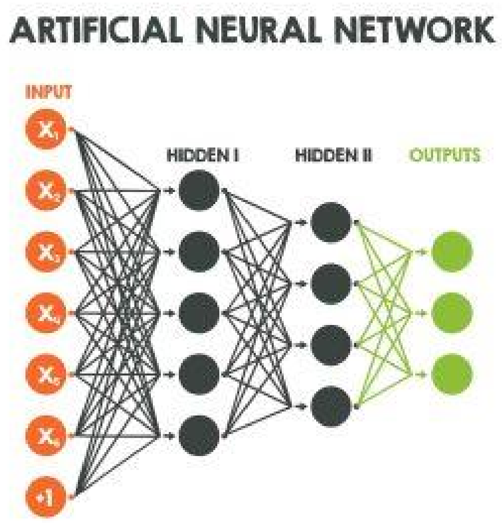

Figure 1.5: Neural Network Architecture

Each perceptron transforms N inputs to one output according to the relation:

$$f\left(\sum_{i=1}^{N} WiXi + b\right)$$

The entries Xi are multiplied by the weights Wi, then summed with a bias b to end up as parameters to an activation function. The weights and biases are model variables that are updated to improve the precision of the network as for the activation function it is a nonlinear function to represent complex relations.

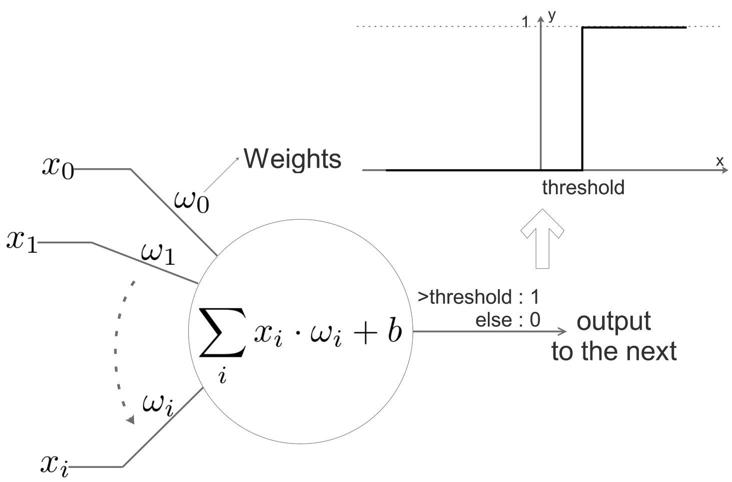

Figure 1.6: Diagram of the operation of a perceptron

7 Convolutional neural network CNN

Convolutional neural network is a variation of neural networks that is frequently used in video and image recognition. It is built by stacking pairs of convolution layer and sub-sampling layer to extract attributes, then a fully connected network to make a prediction.

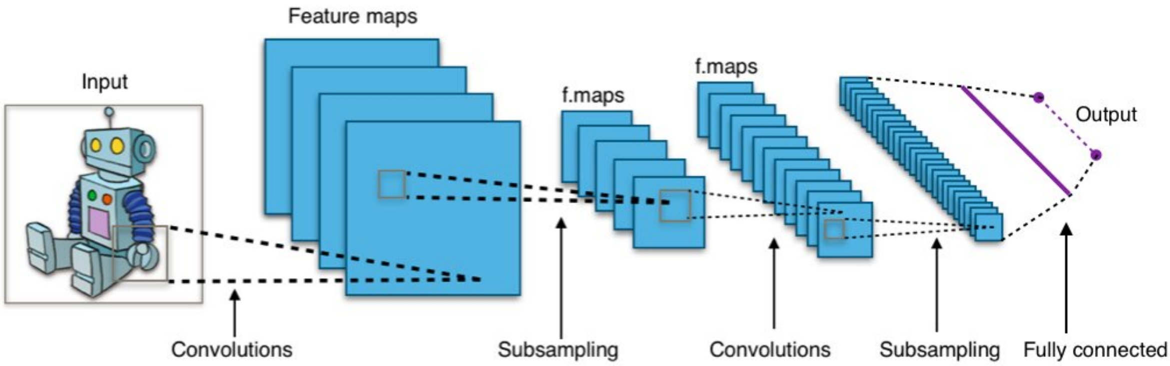

Figure 1.7: a typical CNN

7.1 Convolution layer

Used to detect the attributes of an image by applying successive transformations to it with a set of filters. These filters are dragged over the entire image calculating the convolution in each step as shown below before applying the activation function to each pixel

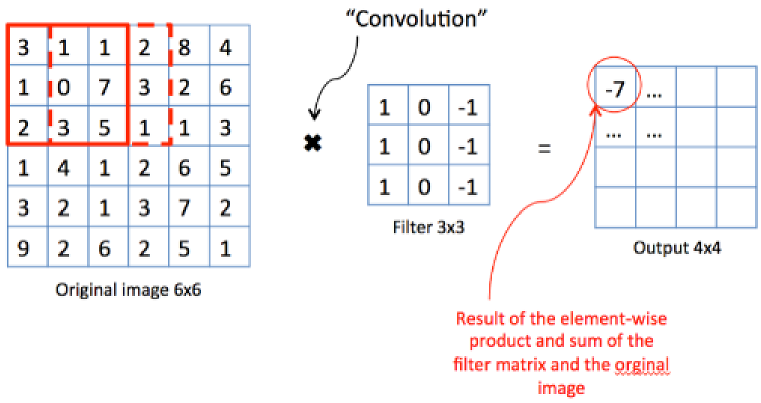

Figure 1.8: Convolution product

7.2 Sub-sampling layer (pooling)

Is a layer that always sits after the convolution layer, used for dimensionality reduction without changing the variance of the data. This reduces parameters and computation time.

The pooling operation consists of dividing the images into sub-matrices of equal sizes then reconstructing an image by choosing the max of the pixels of each sub-matrix in the event of max pooling, or of taking the average of the pixels of each sub-matrix in average pooling.

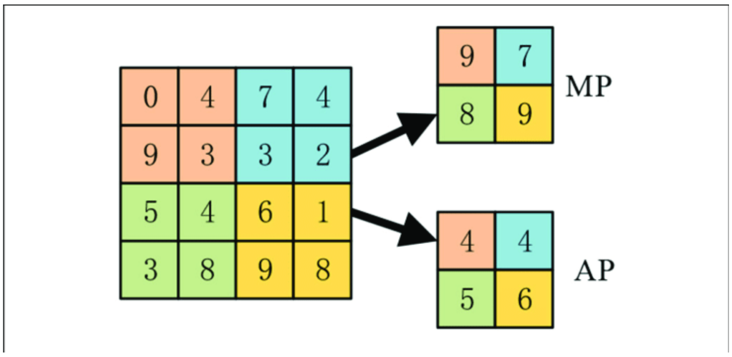

Figure 1.9: Sub-Sampling Method

8 Deep Convolutional neural networks DCNN

Deep Convolutional Neural Networks (DCNNs) have emerged as a powerful class of neural networks specifically designed for processing and analyzing visual data, such as images and videos. By incorporating convolutional layers, these networks can automatically learn hierarchical representations of visual features, enabling them to achieve state-of-the-art performance in various computer vision tasks.

The architecture of a DCNN consists of multiple layers, typically including convolutional layers, pooling layers, and fully connected layers. Convolutional layers are responsible for capturing local patterns and features in the input data through the application of filters or kernels. Pooling layers reduce the spatial dimensions of the feature maps, effectively downsampling the information. Finally, fully connected layers, similar to those in traditional neural networks, perform high-level feature integration and decision-making.

Next in our project, The network has learned rich feature representations for a wide range of images. The input image size for the network is 48 by 48. The DCNN network is divided into 23 stages. The first stage consists of a 3x3 convolution kernel with a stride. Additionally, stages 2 through 22 are repeatedly stacked. Furthermore, stage 23 consists of a 1x1 convolution layer for the output .

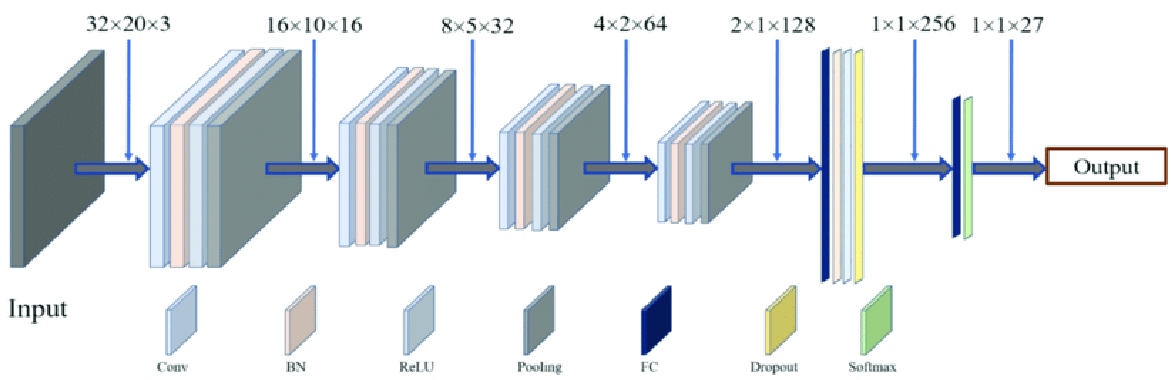

Figure 1.10: Exemple of design of a DCNN

9 Classification of Emotions

9.1 Workflow

Classification is a supervised learning problem used to predict discrete values in a finite interval. The creation of this model is done with an iterative work process schematized below:

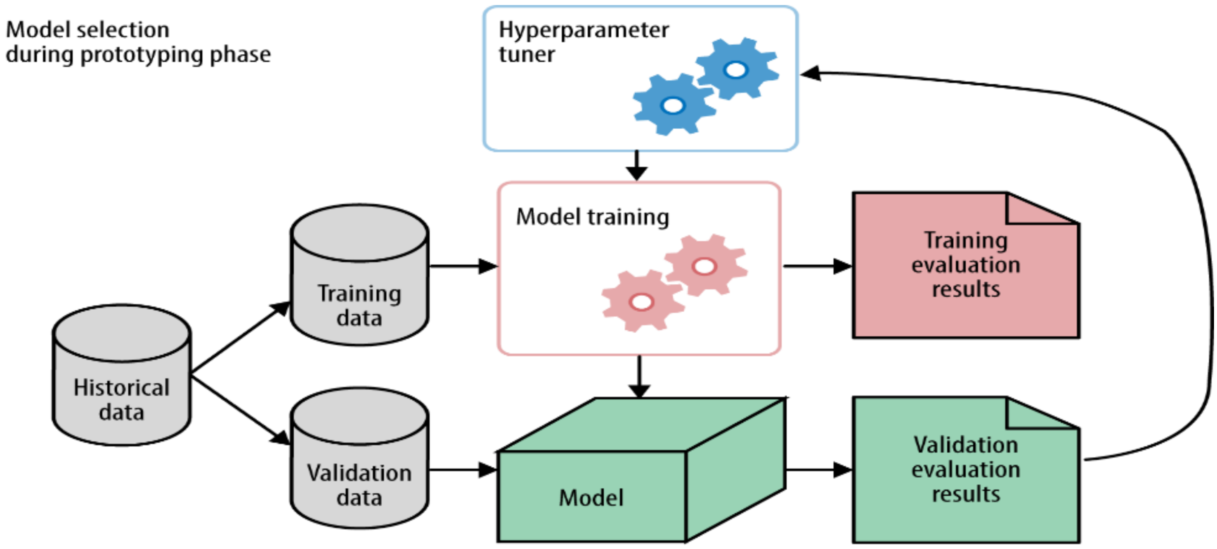

Figure 1.11: WorkFlow of AI algorithm

9.2 Data Sets

In order to successfully classify, one must first prepare a large data set suitable for training the model. To properly evaluate the model, we divide this data into validation and test data as shown in the following figure:

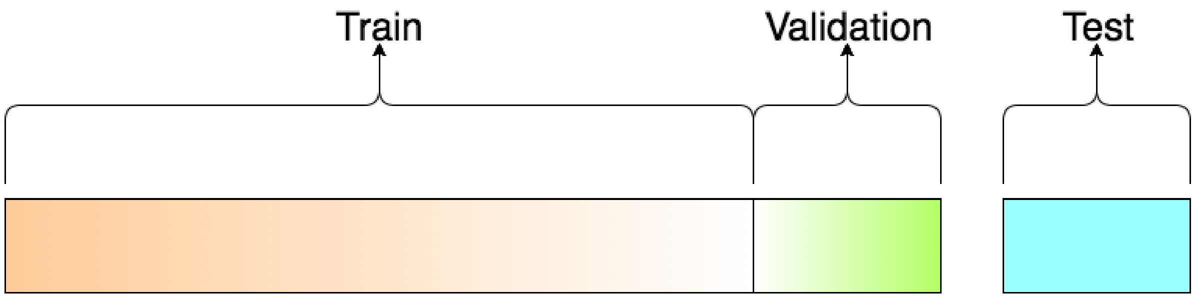

Figure 1.12: Data Distribution

9.2.1 Train Data :

Our database is composed of 35887 images. Indeed, our training dataset is composed of 32299 images (90%).

9.2.2 Test Data :

To properly evaluate our model, we took 3588 images (10%).

10 Problem Behind the Theoretical Analysis

The problem of facial emotion recognition can be formulated as follows: given one or multiple images of a face, the task is to find or verify the emotion by comparing their face to a set of face images stored in a database. Facial emotion recognition poses numerous challenges as faces are deformable 3D objects. Such systems need to overcome the following problems:

10.1 Poses variation

Changes in orientation and changes in the angle of inclination of the face lead to many appearance medications in the collected images. Causing deformations that vary the overall shape of the face.

Figure 1.13: Change of Angles

10.2 lighting Change

The intensity and direction of lighting during the shot greatly affects the appearance of the face in the image. So we must especially avoid collecting views at different times, inside and outside. Given a particular shape of the human face, these changes in lighting can reveal shadows that highlight or hide certain facial features.

Figure 1.14: Lighting change

Conclusion

In conclusion, the theoretical analysis chapter has provided a thorough exploration of the fundamental principles and theories that underpin emotional face recognition. By examining the theoretical frameworks of facial expression recognition, computer vision, machine learning, datasets, and ethical considerations, we have established a strong theoretical foundation for the subsequent stages of the project.

The analysis has revealed the intricate relationship between facial expressions and emotions, shedding light on the psychological and physiological factors influencing facial expressions. Furthermore, the exploration of computer vision and image processing techniques has demonstrated their significance in extracting meaningful features from facial images.

2

Needs analysis and specification

Introduction

The requirements analysis phase is a critical step in any project development as it lays the foundation for the functionality and capabilities of the application. This chapter focuses on the importance of conducting a thorough requirements analysis for our project and outlines the key aspects that will be addressed. By identifying the actors involved, defining functional and non-functional requirements, and modeling using UML diagrams, our goal is to create a clear understanding of system requirements and expectations. This chapter is an important guide for ensuring that your application meets its intended goals and provides a high-quality user experience.

1 Actors

We have identified the actor who will act on our application which is:
User : Person responsible for managing the application

2 Analyse of needs

In this section, we will outline the functional, non-functional, and domain requirements in detail.

2.1 Functional needs

The application must be able to: - Preprocess and extract feature from input image. - Predict the emotion in the input image.

2.2 Non Functional needs

These needs that the application must meet are implicitly required, namely: - Python program: based on Tensorflow, numpy and pandas libraries. - Speed: the system response time must be less than two seconds.

3 Specification of needs

In this part we will present the interactions between the actors and the system through UML diagrams such as the use case diagram and the sequence diagram in order to better understand the functional aspect of the project.

3.1 Modeling language

For modeling, we have chosen as UML "Unified Modeling Language" which is a general modeling language which is intended to define a standard way of visualizing the way a system has been designed.

3.2 User case diagram

The use case diagram is a simple representation of the interaction of actors with the system showing the different use cases in which the user is involved. This diagram can identify the actions performed on the system and the reactions of the latter and also its limits.

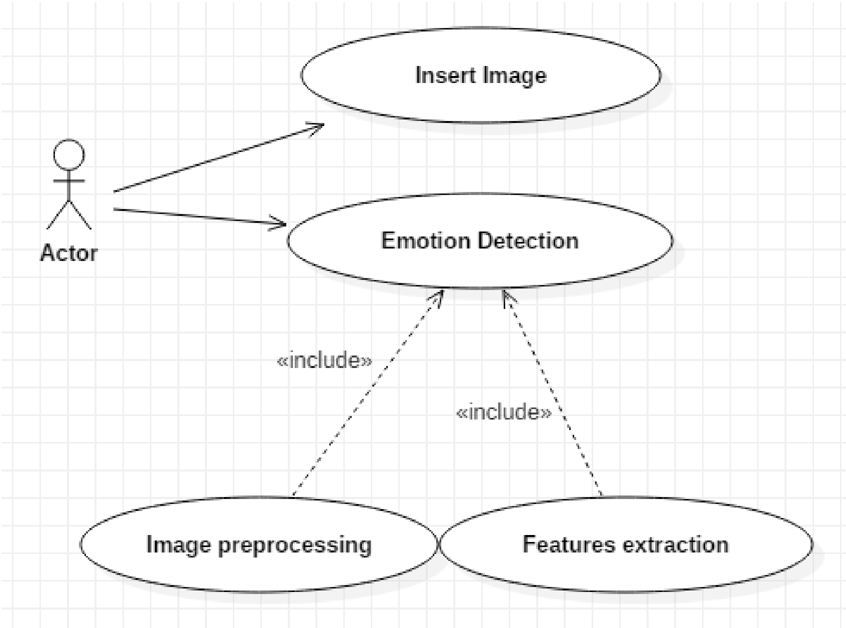

Figure 2.1: User case Diagram

3.3 Sequence Diagram

In order to show the interactions between the actors and the application, we will present a sequence diagram to better describe the operations carried out and the sequence of the messages exchanged between the objects necessary for the execution of the functionality

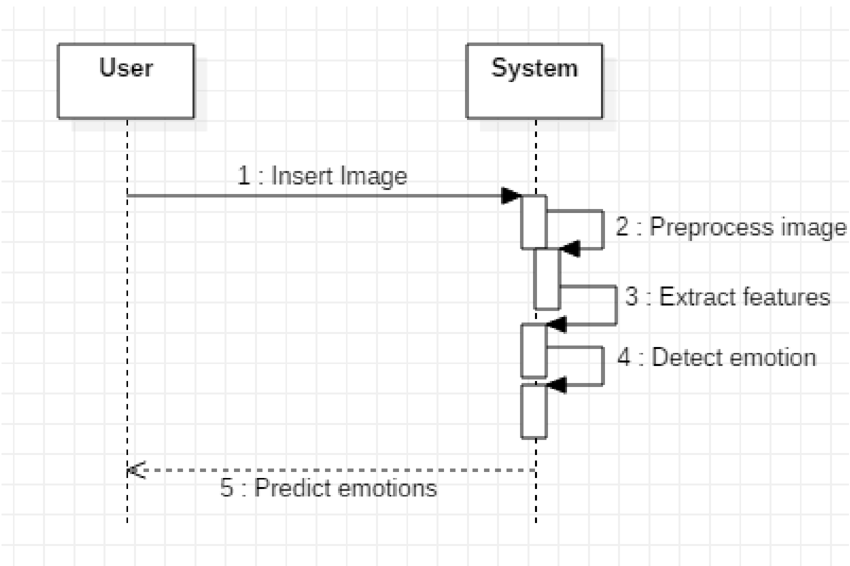

Figure 2.2: Sequence diagram

3.4 Activity Diagram

An activity diagram is a graphical representation that shows the flow of activities and operations in a system or a specific process. A behavioral diagram in Unified Modeling Language (UML) that illustrates the sequence and dependencies of tasks, actions, and decision points in a given workflow.

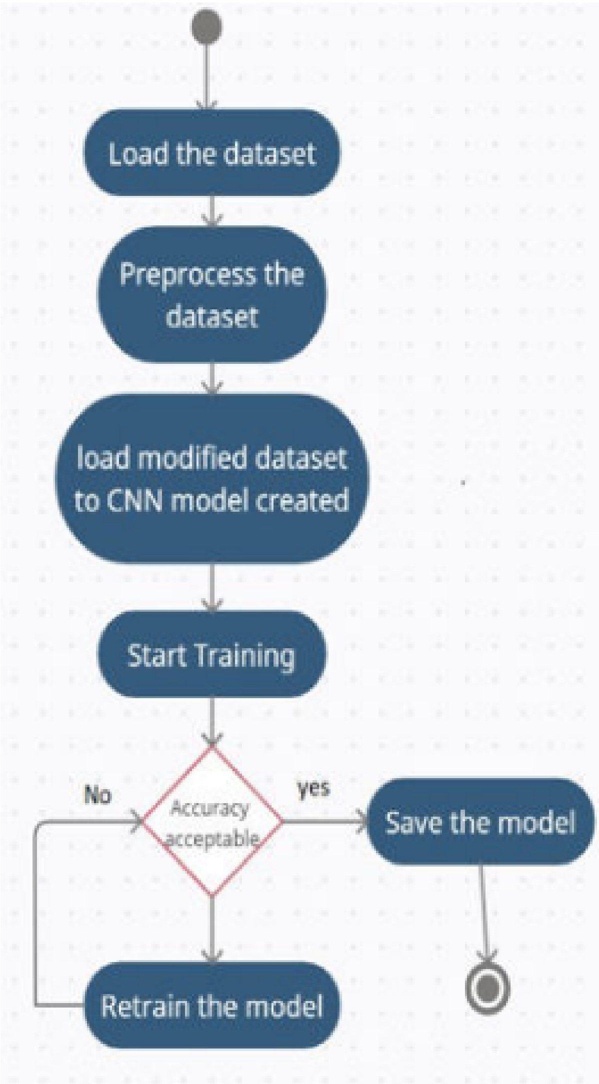

Figure 2.3: Activity diagram

Conclusion

The needs analysis phase acts as a bridge between the project's initial conception and the subsequent implementation, providing a clear roadmap for development and ensuring that the final product meets the identified needs. The insights gained from this chapter will guide us as we proceed to the implementation phase, enabling us to create a system that fulfills the expectations of its users and delivers a seamless and satisfying experience.

3

Implementation

Introduction

After an in-depth theoretical study of emotional facial recognition, we are now starting the implementation phase of the project. This stage marks an important milestone on the way from the conceptual and theoretical realm to the practical realm of turning ideas into reality. In this section, we dive into the details of the implementation process and discuss the tools, methods, and techniques used to develop our expressive facial recognition system.

The aim of this implementation phase is to apply the theoretical knowledge gained during the course to create a powerful and efficient system capable of accurately recognizing and classifying emotions based on facial expressions. We outline the steps for collecting and preprocessing data, selecting appropriate algorithms and models, integrating relevant libraries and frameworks, and fine-tuning system parameters for optimal performance.

1 Tools And Libraries Used

OpenCV

OpenCV is an open-source library for computer vision and image processing. It provides a wide range of functions for image transformation, including converting images to grayscale. OpenCV is written in C++, but it also provides bindings for Python. It is a complete package that can be used with other libraries to form a pipeline for any image extraction or detection framework. OpenCV is a powerful tool for image transformation and processing. It is used by a wide range of developers, including researchers, engineers, and artists.

Tensoflow

TensorFlow is an open-source machine learning framework developed by Google. It provides a comprehensive set of tools and libraries that enable developers to build and deploy machine learning models efficiently. TensorFlow is designed to handle both deep learning and traditional machine learning tasks. It allows users to create computational graphs that represent the flow of data through a series of mathematical operations, making it easier to define and train complex models.

Python

Python, a powerful scripting language, proves highly valuable in addressing statistical challenges associated with machine learning algorithms. It offers a multitude of utility functions that aid in preprocessing tasks. Python's processing speed is swift, and it boasts broad platform support. Integration with C++ and other image libraries is straightforward, facilitated by its seamless compatibility. Moreover, python provides built-in functions and libraries for efficient data storage and manipulation across diverse data types. The pandas and numpy frameworks, integral to python, empower data manipulation as per specific requirements. Notably, numpy arrays allow the creation of feature-rich datasets with the flexibility of handling n-dimensional data.

Scikit-learn

Scikit-learn, the machine learning library in python, encompasses matplotlib, numpy, and an extensive collection of machine learning algorithms. It boasts a user-friendly and intuitive API, simplifying the process of building and deploying models. With its comprehensive set of functions, scikit-learn facilitates data analysis and visualization tasks. Leveraging its feature reduction, feature importance, and feature selection functions, scikit-learn enables the creation of robust feature sets. Furthermore, the library offers a wide range of algorithms suitable for classification and regression problems, along with their respective sub-types.

Jupyter Notebook

Jupyter Notebook serves as the integrated development environment (IDE) that brings together python and all the libraries utilized in our implementation. It offers an interactive environment where computations can be executed, although more intricate computations may require additional processing time. The instant display of plots and images enhances the overall user experience. Jupyter Notebook acts as a convenient hub, meeting various project requirements, and simplifies the integration of libraries such as Dlib, OpenCV, and Scikit-learn.

2 Training the model On the dataset FER-2013

2.1 Overview

The FER-2013 dataset comprises 28,000 labeled images in the training set, 3,500 labeled images in the development set, and 3,500 images in the test set. Each image in FER-2013 is labeled with one of seven emotions: happy, sad, angry, afraid, surprise, disgust, and neutral. Among these emotions, happy is the most prevalent, establishing a baseline accuracy of 24.4% for random guessing. The FER-2013 images consist of grayscale headshots, both posed and unposed, with dimensions of 48x48 pixels. This dataset was created by collecting the outcomes of a Google image search for each emotion, including synonyms.

2.2 Import & Manipulate the data set

Since the format of the data is in zip format we have import to utilize the zipfile library which will help us extract the dataset. Once extracted we can have a view of our dataset. In our case the format is CSV that's why we use the pandas readcsv function to return a data frame that we can process later. The picture below showcases the data format and the first 5 rows.

```
(35887, 3)
```

	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training

Figure 3.1: Dataset in CSV format

Additionally, it is important to note that the dataset contains 35,887 rows and 3 columns. Each row includes the emotion label, pixel values, and usage (training/testing) for a particular image. The FER-2013 dataset consists of 7 classes of emotions. Furthermore, the pixel values are represented in grayscale, which explains why the pixel column is one-dimensional.

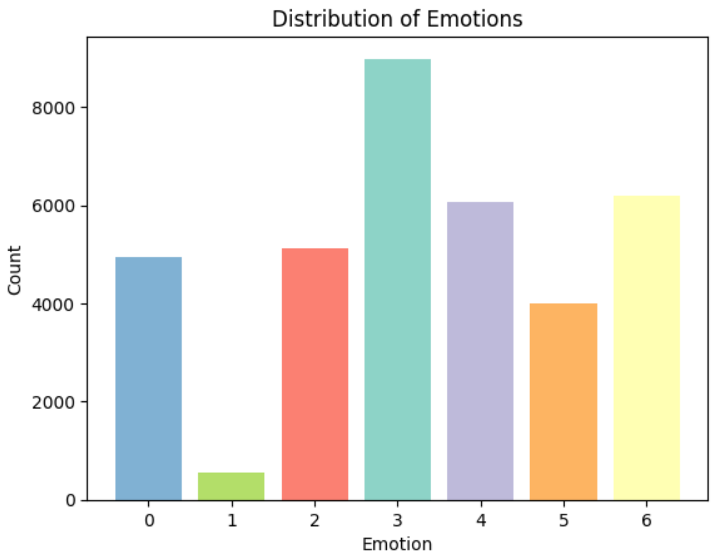

Figure 3.2: Distribution of emotion categories in the dataset

One way to get a good sens of our data is to iterate over all the CSV file which won't be practical. That's why we view the data distribution using the pyplot bar chart which will help us understand the count of every class.

We can also view the data using pyplot layout feature which will return different images for every category. The figure below showcases random images that belongs to it's according category.

Figure 3.3: Some Images from the dataset

2.3 Data Preprossessing

After analyzing the data distribution, we have made the decision to narrow our focus to three specific classes for classification: happiness, sadness, and neutral. Therefore, in the initial phase, our goal will be to predict these three classes exclusively. Consequently, we will only be interested in the classes labeled as [3, 4, 6]. In addition to that we need to split the data set into training and validation set. In our case 10% of the training Data Set represents the validation data which is sufficient to evaluate the model's on data that is different from the training data. LabelEncoder object is created and assigned to the variable le. The LabelEncoder is a utility class from the scikit-learn library that is commonly used for encoding categorical labels into numerical values.

One more data preprossessing is compulsory which consists of Normalization. In our case X_train and X_valid are being normalized before being used as input data for a neural network. Normalization is an important preprocessing step when working with neural networks because they are sensitive to unnormalized data. The normalization is achieved by dividing the pixel values in X_train and X_valid by 255. The pixel values in most image datasets range from 0 to 255, where 0 represents black and 255 represents white. Dividing each pixel value by 255 scales the values to a range between 0 and 1.

2.4 Creating the model and Compiling it

The model is composed of multiple layers that employ various convolutional and max pooling techniques. The initial layers are responsible for extracting features from the input data. Towards the end of the model, there are fully connected layers that generate the output necessary for predicting the corresponding classes. These fully connected layers utilize the extracted features to make accurate predictions. This architecture employs techniques such as ELU activation, he_normal kernel initialization, batch normalization, and dropout regularization to improve generalization and network performance.

In order to view the model architecture we can use the tensor flow function plot_model which will result in the corresponding image below.

When we compile a model in Keras, we specify the loss function, optimizer, and metrics that will be used during the training process. Compiling the model prepares it for training by configuring its learning process.

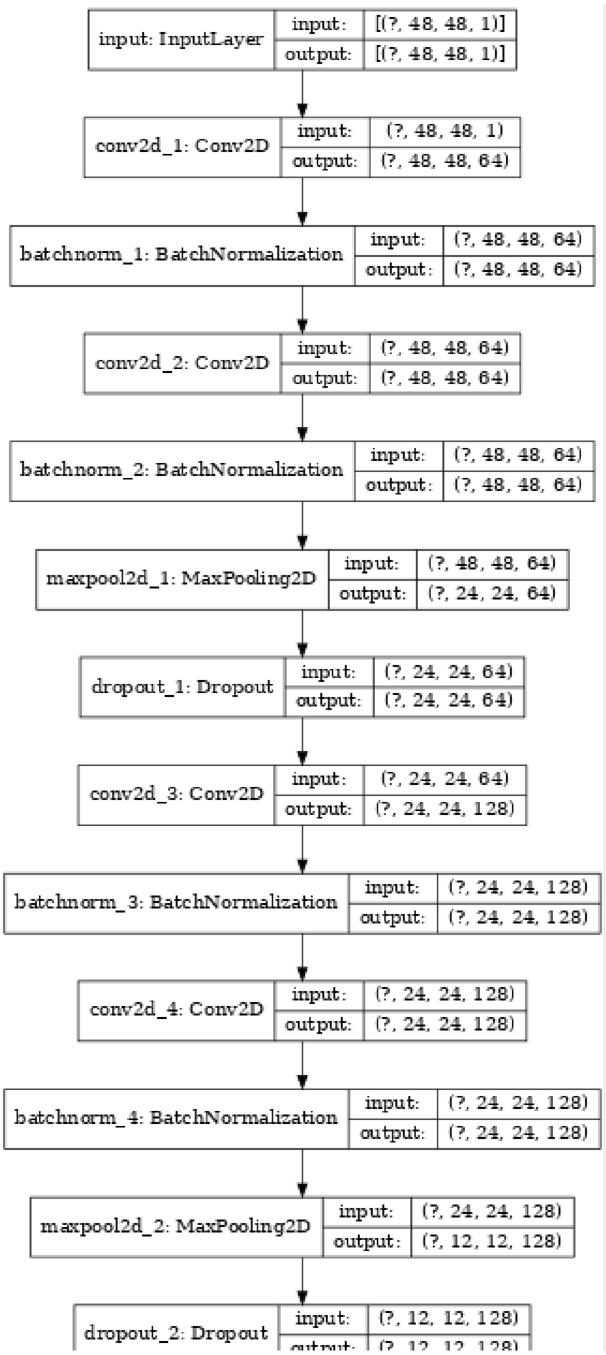

Figure 3.4: Algorithm part 1

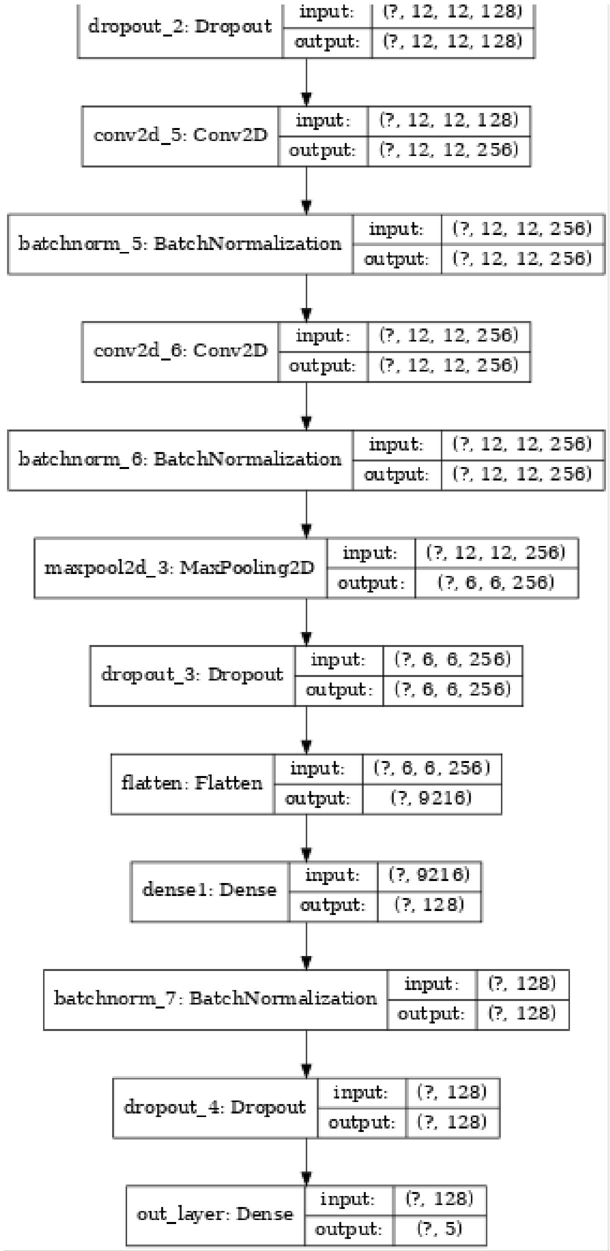

Figure 3.5: Algorithm part 2

2.5 Data augmentation prior to training

We use the ImageDataGenerator to perform data augmentation on the training data. In order to increase the size of the training dataset by applying various transformations to the existing images. It helps to improve the model's ability to generalize and handle variations in the data.

The parameters are defined below :

Parameter	Value
rotation_range	15
width_shift_range	0.15
height_shift_range	0.15
shear_range	0.15
zoom_range	0.15
horizontal_flip	True

Table 3.1: Parameters for Data Augmentation

2.6 Model Training

An essential feature during training is the callback function, which can be included using the model.fit method. By specifying a set of callback functions, we can effectively prevent overfitting and halt the training process based on specific metrics or patterns. These callback functions provide valuable functionality, enabling us to monitor and control the training process for optimal performance.

In our case we used EarlyStopping which is used to prevent overfitting and stop the training process early if the model's performance on the validation data stops improving. It monitors a specified metric, in this case, 'val_accuracy', which is the validation accuracy. If the validation accuracy doesn't improve by a minimum delta of 0.00005 for a certain number of epochs (patience), the training is stopped. In addition to that we call ReduceLROnPlateau a function that adjusts the learning rate (LR) dynamically during training to optimize the model's performance.

We run the model for 100 epochs and for a batch_size equal to 32 and we keep tracking the model history after every epoch that is done by saving every metric in a history array. We can also visualize the model metrics during training using the tensor board.

```
Epoch 1/100
<ipython-input-28-22665cd605d0>:10: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  history = model.fit_generator(
598/598 [==============================] - 26s 33ms/step - loss: 1.2927 - accuracy: 0.4019 - val_loss: 1.0232 - val_accuracy: 0.4927 - lr: 0.0010
Epoch 2/100
598/598 [==============================] - 20s 33ms/step - loss: 1.0261 - accuracy: 0.4801 - val_loss: 0.9558 - val_accuracy: 0.5252 - lr: 0.0010
Epoch 3/100
598/598 [==============================] - 20s 34ms/step - loss: 0.9201 - accuracy: 0.5510 - val_loss: 0.8037 - val_accuracy: 0.6215 - lr: 0.0010
Epoch 4/100
598/598 [==============================] - 20s 34ms/step - loss: 0.8214 - accuracy: 0.6104 - val_loss: 0.7395 - val_accuracy: 0.6530 - lr: 0.0010
```

Figure 3.6: Training the model and viewing its progress

2.7 Results

After training we can visualize the model accuracy and loss for both train and validation set.

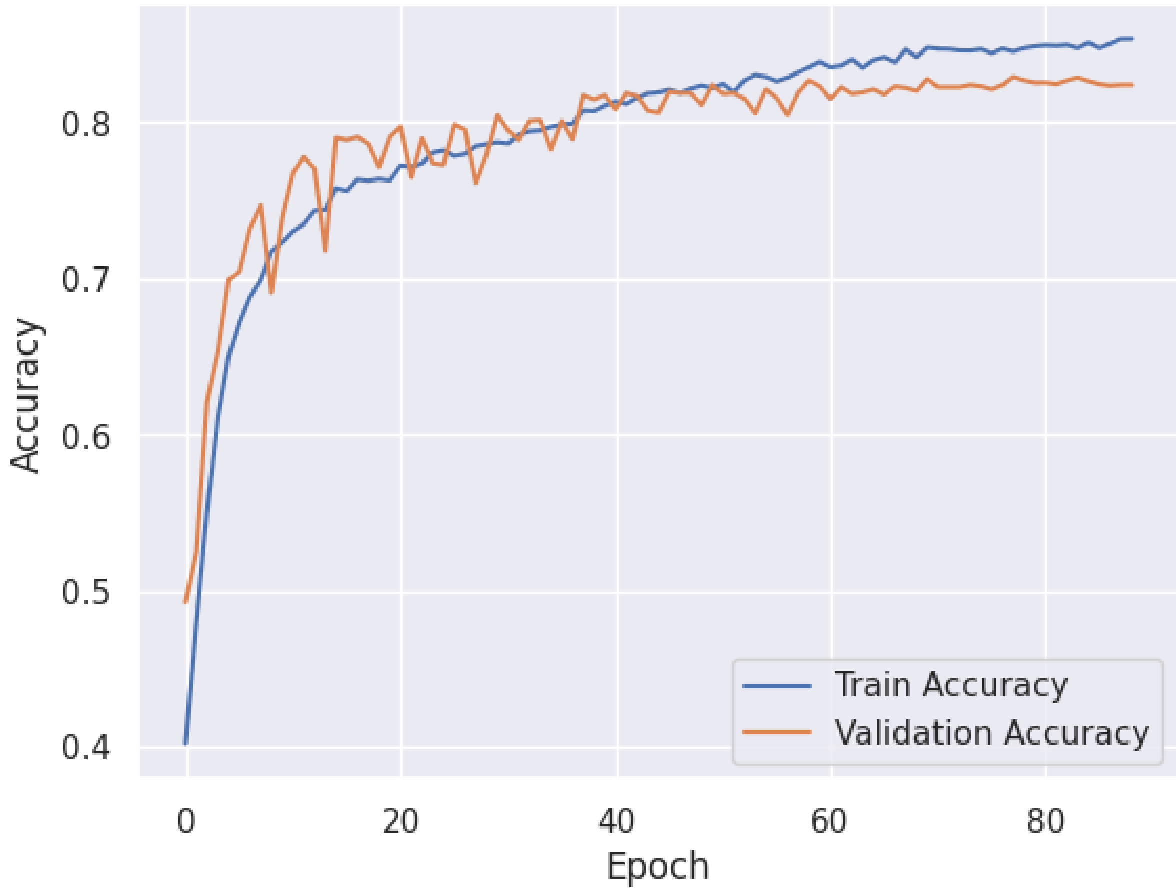

Figure 3.7: Accuracy evolution for train and validation data

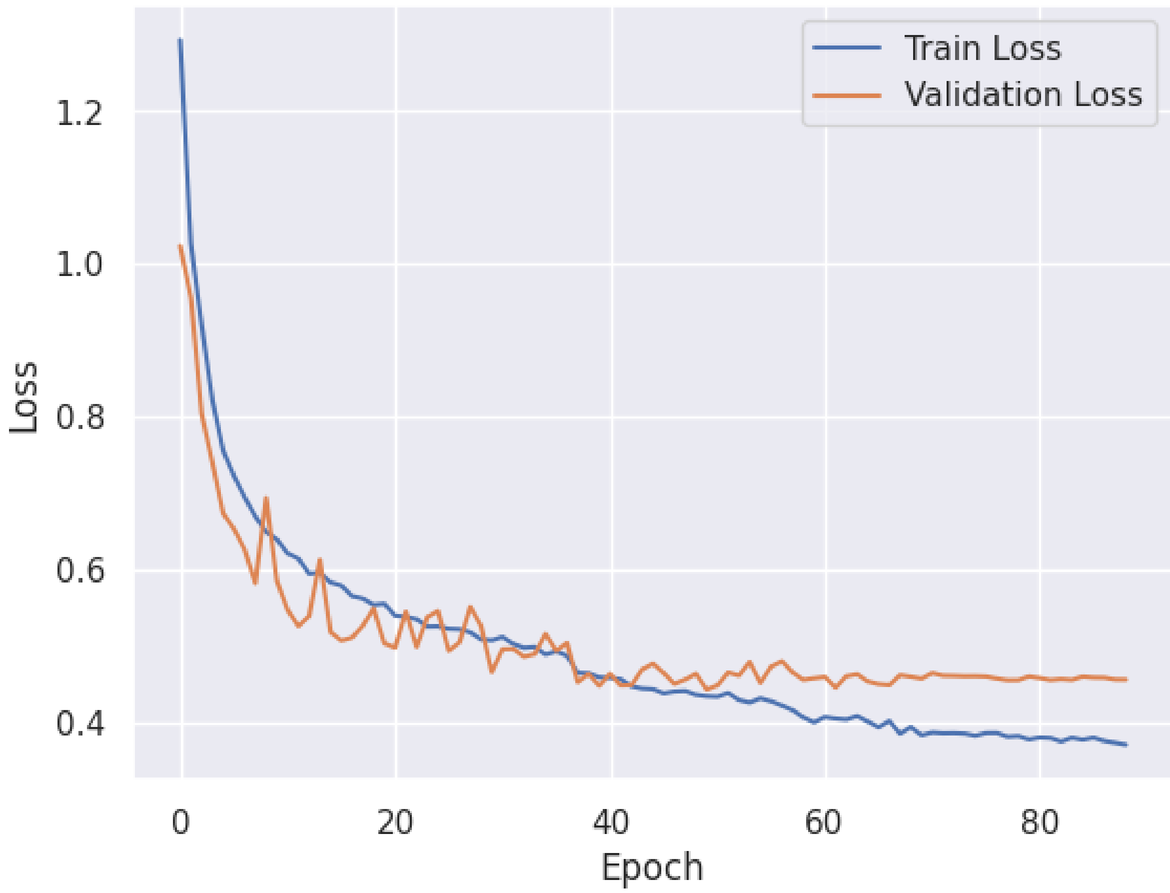

Figure 3.8: Loss evolution for 100 epoch

The training set achieves a final accuracy of 0.8537, while the validation set attains an accuracy of 0.8242. These metrics provide a general assessment of the model's performance. However, to gain deeper insights into the model's performance, it is crucial to leverage additional tools such as the confusion matrix. By utilizing the confusion matrix, we can obtain a more detailed understanding of how the model performs in terms of correctly classifying each individual class and identifying any patterns or areas that require further improvement.

	precision	recall	f1-score	support
0	0.94	0.92	0.93	899
1	0.79	0.74	0.76	608
2	0.72	0.78	0.75	620
accuracy			0.83	2127
macro avg	0.82	0.82	0.81	2127
weighted avg	0.83	0.83	0.83	2127

Figure 3.9: Confusion Matrix

A confusion matrix is a table that is used to evaluate the performance of a classification model. It provides a detailed breakdown of the predicted and actual class labels for a given data set. In our case it reveals that our model performs well in classifying the "happy" emotion, but it's performance is relatively lower for the other two classes.

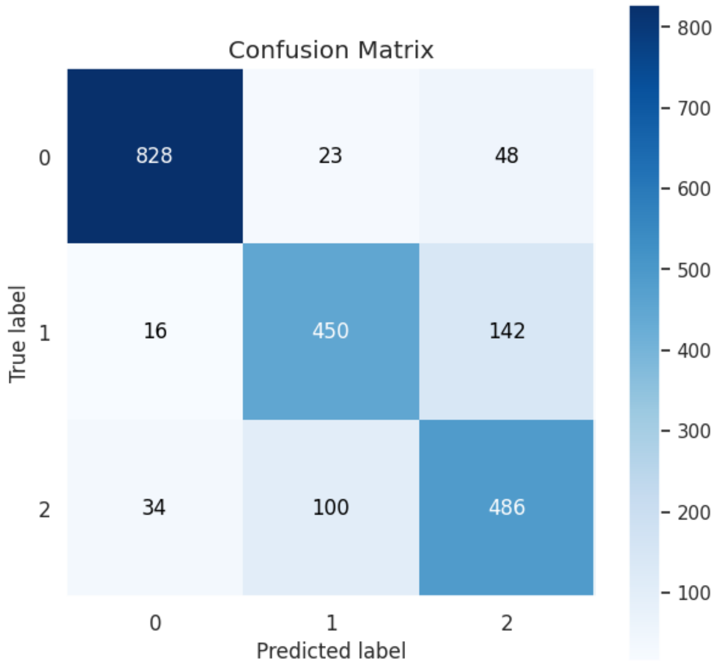

Figure 3.10: Confusion Matrix Heatmap

2.8 Model Testing on Evaluation data

The next step is to test the model on the evaluation dataset for that we pick random images from every class and run inference. We can clearly see that the model miss predicted 4 out of 18 images which also proves that the model isn't 100% accurate but still can give some good results.

Figure 3.11: Testing the model on examples from the test set

3 Interpretation

3.1 Visualizing the CNN

The output of the feature maps is visualized below :

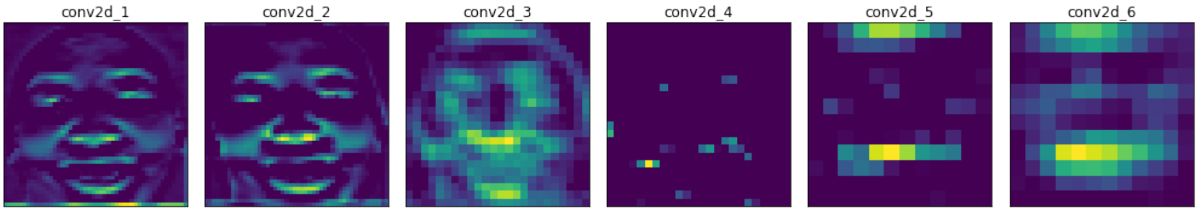

Figure 3.12: Visualization of different convolutional layers

We can observe that the first convolution are working on feature extraction such as collecting different shared features such as extracting the edges and other common feature for the face. At the final layers we can see that our model focuses on important aspects of the image lips, eyes and eyebrows.

The analysis of the epoch history reveals a gradual improvement in accuracy, reaching a commendable +83% accuracy on both the training and validation sets. However, as the training progresses, a phenomenon known as overfitting becomes apparent. Overfitting occurs when the model becomes excessively tuned to the training data, resulting in a decrease in performance on unseen data. This observation suggests that the model has learned the specific patterns and details of the training set too well, hindering its ability to generalize to new data. To address this issue, techniques such as regularization, dropout, or early stopping can be employed to prevent overfitting and improve the model's overall performance on unseen data.

The confusion matrix reveals that our model performs well in classifying the "happy" emotion, but its performance is relatively lower for the other two classes. This could be attributed to the limited amount of data available for these classes. However, upon reviewing the images, it became apparent that some instances within these classes posed challenges even for human observers to accurately determine whether the person appeared sad or neutral. Facial expressions can vary significantly among individuals, and what might be perceived.

3.2 Limitations

In a second phase we train the model on the 7 classes 0:'anger', 1:'disgust', 2:'fear', 3:'happiness', 4: 'sadness', 5: 'surprise', 6: 'neutral' We also keep track of the model's metrics. The figures below showcase the model's metrics when trained for 7 classes.

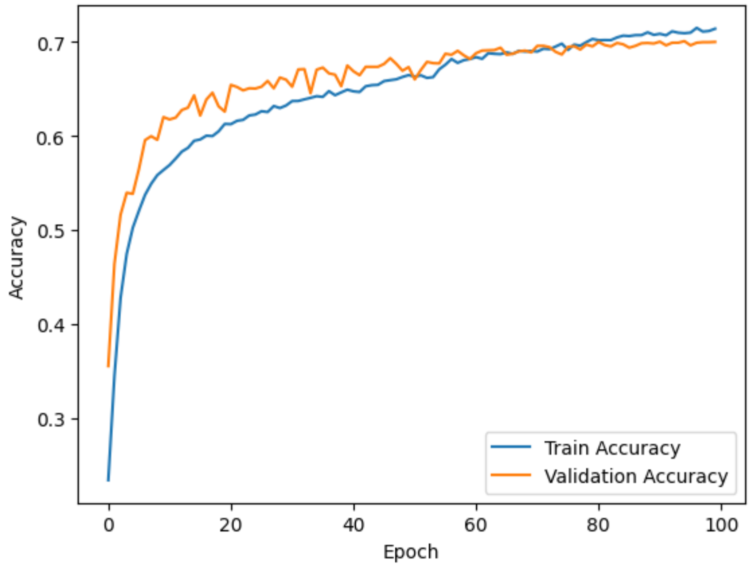

Figure 3.13: Accuracy evolution for 7 classes

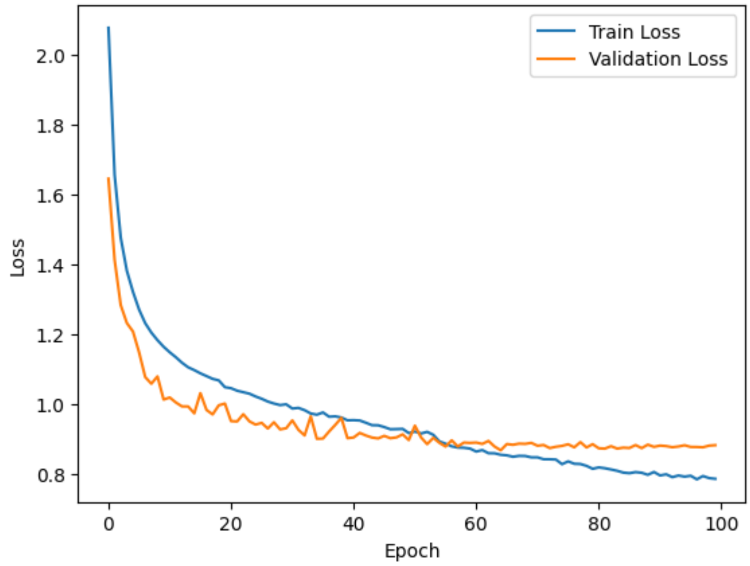

Figure 3.14: Loss evolution for 7 classes

For the same training parameters (batch_size=32 , Number of epochs = 100) We can observe a lower accuracy which is equal to 0.7002 for the validation set, which was expected while increasing the model's output classes to 7.

In order to adapt to the increase in the number of output classes and improve the accuracy on the validation set, several strategies can be considered:

- Adjust the model architecture: With a higher number of output classes, it may be necessary to modify the model architecture to accommodate the increased complexity. This could involve adding more convolutional layers, increasing the number of filters, or adjusting the size of the layers to capture more intricate patterns and features specific to the new classes.
- Increase the model capacity: If the model is not sufficiently complex to handle the expanded classes, it might be necessary to increase its capacity by adding more layers or increasing the number of parameters. This can help the model learn more intricate representations and improve its ability to differentiate between the new classes.
- Collect more data: If the decrease in accuracy is due to a lack of data for the new classes, collecting additional data specific to those classes can help improve the model's performance. More diverse and representative data will allow the model to learn a more comprehensive range of patterns and improve its generalization capabilities.
- Fine-tune hyperparameters: Adjusting hyperparameters such as learning rate, batch size, optimizer, or regularization techniques can also have an impact on the model's performance. Fine-tuning these hyperparameters specifically for the new multi-class scenario may help achieve better results.

Conclusion

In conclusion, the implementation phase of our project represents an important transition from theory to practice. At this stage, we manage to translate the theoretical insights from our research into a tangible emotional face recognition system. By applying different tools, methods, and techniques, we have developed an efficient and practical solution capable of accurately detecting and classifying emotions based on facial expressions.

Throughout the implementation, we encountered challenges and made informed decisions to effectively address them. We collected and preprocessed necessary data, selected appropriate algorithms and models, and integrated related libraries and frameworks. By fine-tuning system parameters, we strive to achieve optimal performance and accuracy.

Overall, the implementation phase has been instrumental in bringing our project to life and realizing the practical implications of emotional face recognition. It sets the stage for further advancements in this field and opens up new avenues for research, development, and applications that can benefit various industries and domains.

Conclusion and perspectives

Finally, the project looks at emotional facial recognition and uses machine learning and artificial intelligence (AI) techniques to accurately identify and interpret human emotions based on facial expressions. By exploring advanced algorithms and models, we hope to contribute to the growing field of emotional facial recognition and open up new possibilities for understanding and processing human emotions.

Throughout the project, we conducted an in-depth theoretical study of machine learning and artificial intelligence to understand their importance and role in emotional facial recognition. By harnessing the power of these technologies, we were able to develop a robust system that can analyze facial images, extract meaningful features, and accurately classify emotional states.

The project's approach includes collecting and preprocessing suitable datasets, selecting suitable features, and training machine learning models. We use techniques such as deep learning, leveraging convolutional neural networks (CNNs) and other advanced architectures to achieve powerful emotional facial recognition.

The results of our project demonstrate the effectiveness of the developed system in accurately detecting and interpreting human emotions based on facial expressions. Through rigorous evaluation and comparison with existing methods, we discover promising performance, paving the way for potential practical applications in various domains such as psychology, human-computer interaction, social robotics

Despite the project's success, it's important to recognize its limitations. The accuracy of emotional face recognition is affected by many factors such as lighting conditions, image quality, and individual differences in facial expressions. Further improvements can be made by examining larger and more diverse datasets, fine-tuning model architectures, and incorporating temporal information for a more comprehensive understanding of sentiment.

In summary, our project demonstrates the potential and importance of machine learning and artificial intelligence in the field of emotional facial recognition. By accurately recognizing and interpreting human emotions, we can pave the way for future machines to have a deeper understanding of human emotions, leading to improved interactions, improved user experiences, and more sensitive and responsive technological environments.

Bibliography

[1] Khadija LEKDIOUI. (2018) . Universite de Technologie de Belfort Montbeliard : Reconnaissance d'etats emotionnels par analyse visuelle du visage et apprentissage machine.

[2] N. Sebe. (2007) . Faculty of Science, University of Amsterdam, The Netherlands : Authentic facial expression analysis.

[3] GAURAV SHARMA, Kaggle . [https://www.kaggle.com/code/gauravsharma99/visualizing-the-cnn/]

[4] Wafa Mellouk, (2020) . Laboratoire d'automatique de Tlemcen (LAT), Tlemcen university, BP 320, Chetouane Tlemcen 1300, Algeria : Facial emotion recognition using deep learning: review and insights.

[5] Nitisha Raut , (2018) . San Jose State University :Facial Emotion Recognition Using Machine Learning.

[6] Konstantina VEMOU, (2021) . facial emotion recognition. [https://edps.europa.eu/data-protection/our-work/publications/techdispatch/techdispatch-12021-facial-emotion-recognition_en]

[7] Ned Hill, (2022) . How AI and Computer Vision Shape Our World. [Spiceworks]