

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
UNIVERSITY MOHAMED BOUDIAF - M'SILA

FACULTY OF TECHNOLOGY
DEPARTEMENT OF ELECTRONICS

N° :



FIELD : SCIENCE AND
TECHNOLOGY

FILIERE : ELECTRONICS

OPTION :

**Brief submitted for obtaining
Academic Master's degree**

By:

**Benhamida Hani Karim
Bendaas Mohamed Chahem**

a Master's Thesis Titled

***Face expression recognition with
Convolutional neural networks
(CNN)***

Defended before the jury composed of:

Mm: Fodil Malika	University Mohamed Boudiaf M'sila	President
Mr: Mezaache Hatem	University Mohamed Boudiaf M'sila	Protractor
Pr: Bouchama Idriss	University Mohamed Boudiaf M'sila	Examiner

Academic year : 2022 / 2023

Acknowledgments :

First of all, we want to thank the Almighty God for giving us the strength and the will to complete this dissertation and for the help he gave us during my years of study.

I would like to express my gratitude to my director, Mr. Mezaache Hatem, I thank him for having supervised, guided, helped and advised me throughout the period of this study, as well as to the president and members of the jury. who have done the honor of accepting, reviewing and evaluating our work.

I extend my sincere thanks to all the professors, speakers and all the people who by their words, their writings, their advice and their criticisms have guided my reflections and have agreed to meet with me and answer my questions during my research.

I would also like to thank the teachers of the Electronics Department of Mohammed Boudiaf University in M'sila.

I would like to sincerely thank the members of our two families, who always encouraged and supported, without forgetting my friends who have always been there for me, their unconditional support and encouragement have been a great help. Finally, we thank all those who have contributed, directly or indirectly to the development of this thesis, we express to all of them our deep gratitude.

Dedication :

dedicate this modest work to:

To my parents, my pride and happiness, who always supported me and who contributes effectively to success. No tribute could be paid to the height of love Ofwhich they never cease to fill me. May God grant them good health and long life. To my dear sisters especially 'Ines', for their moral supportthroughout my life, without forgetting my parents.To all my friends and comrades from the 'ESEM Master2" group.

To my partner Bendaas Mohamed Chahem, as well as to his family.To the whole Benhamida and Ghadbane family And to all those who have loved me and wish me happiness and success, I say Thank you.

HANI

To my dear parents and all bendaas family who made me what I am today and who have Be careful to guide my steps throughout my life by theirhelpers, May God protect you.

To my brothers sami and houmam and sister. To all my friends, to my partner Benhamida Hani KarimTo all the promo 'ESEM master 2 '
And to all those who have contributed directly or indirectly to make this project possible, I say thank you.

Mohamed

Abstract :

Face expression recognition plays a vital role in various fields, including human-computer interaction, affective computing, and social robotics. Traditional methods for face expression recognition heavily rely on handcrafted features and shallow classifiers, limiting their ability to capture intricate facial dynamics. In recent years, deep learning techniques have emerged as a powerful tool for tackling complex vision tasks, offering promising advancements in face expression recognition.

This thesis focuses on the development and evaluation of a deep learning-based approach for face expression recognition. The primary objective is to leverage the expressive capacity of deep neural networks to accurately detect and classify facial expressions from static images or video sequences. The proposed methodology involves several key stages, including data pre-processing, feature extraction, network architecture design, model training, and performance evaluation.

To facilitate this research, a comprehensive dataset of labeled facial expressions is collected, comprising diverse individuals from different demographic groups. The dataset is carefully curated and annotated to ensure robustness and generalizability of the trained models. Preprocessing techniques such as face alignment, image augmentation, and illumination normalization are applied to enhance the quality and consistency of the data.

For feature extraction, various deep learning architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are investigated to capture both spatial and temporal dependencies in facial expressions. The chosen architecture is optimized to balance model complexity and performance. Transfer learning techniques are explored to leverage pre-trained models and accelerate the training process.

Extensive experiments are conducted to evaluate the proposed methodology on the collected dataset and compare its performance against existing state-of-the-art methods. Various evaluation metrics, including accuracy, precision, recall, and F1-score, are employed to

assess the robustness and generalization capabilities of the developed models. The results demonstrate the effectiveness and superiority of the deep learning-based approach in face expression recognition.

Furthermore, the thesis explores the impact of different factors, such as pose variations, occlusions, and individual differences, on the performance of the proposed models. Mitigation strategies and adaptation techniques are investigated to enhance the models' robustness and adaptability in real-world scenarios.

The outcomes of this research contribute to the advancement of face expression recognition techniques, providing valuable insights into the potential applications of deep learning in this domain. The proposed methodology showcases the ability of deep neural networks to accurately detect and classify facial expressions, thus opening avenues for further research and practical implementations in fields like human-computer interaction, affective

Table of contents:

1.1 ACKNOWLEDGEMENTS	II
1.2 Dedication	III
1.3 Abstract.....	V
1.4 Table of contents.....	VII
1.5 Figures list	XI
1.6 Table list.....	XIII
1.7 General Introduction	XIV
Chapter01(Face Detection).....	1
Introduction	2
2. Digital images	2
3. Images types	3
3.1 Based on coloring	3
3.1.1 binary image	3
3.1.2 Grayscale image.....	4
3.1.3 Truecolor image	4
2.2- Based on the nature	5
2.2.1 Vector graphics(images)	5
2.2.2 matrix graphics(images).....	6
4. image characteristics.....	6
4.1 bit in depth.....	6
4.2 Imagesizeand resolution.....	7

4.3 RGB channels and space colors	7
5. Image processing operations	8
5. Methodology of face detection.....	9
6. Challenges in face detection.....	10
7. Contrast.....	11
8. Machine learning.....	14
8.1 Definition.....	14
8.2 Learning Types of machine Algorithms.....	15
8.2.1 Supervised learning.....	15
8.2.2 Unsupervised learning.....	16
9 Deep learning	18
9.1 Definition.....	18
9.2 Deep learning position	18
10.1 Face detection methods.....	19
10.2 Face Detection Using Haar Cascade Classifier.....	20
10.3 Face Detection Using Local Binary Pattern Cascade.....	22
10.3.1 Classifier (LBP)	22
10.3.2 Principles of local binary patterns	22
10.4 Support Vector Machine(SVM)	24
10.5 Types of SVM.....	24
10.5.1 Linear case.....	24
10.5.2 Non linear case.....	25
II. Conclusion	27

Chapter 02 (artificial intelligence)

I- Introduction.....	29
Types of deep learning	29
1.1 Recurrent Neural Networks(RNNs).....	29

1.2 Long Short-Term Memory (LSTM) Networks	29
1.3 Generative adversarial networks (GANs).....	29
1.4 Autoencoders	29
1.5 Convolutional Neural Networks (CNNs)	29
2. Convolutional Neural Networks layers.....	29
2.1 Convolutional layer.....	30
2.2 pooling layer	31
2.3 Fully connected layer.....	31
3 Convolution operation.....	31
4. Activation functions	34
4.1 sigmoid	34
4.2 tanh.....	35
4.3 Leaky ReLU	36
Parametric Linear Units.....	37
Facial expression recognition	37
Features extracting images	38
preprocessing.....	38
Feature extraction-classification	38
Conclusion.....	40

Chapter 03: simulation and results

I. Introduction	42
Data-set description.....	42
1.2 Evaluation metrics.....	43
Results and Discussion.....	44
Adam Results.....	44
SGD Results	46
RMSprop RESULTS.....	48
Confusion matrix	50
II. Conclusion	51
Final Conclusion.....	52

Parametric Linear Units	37
Facial expression recognition	37
Features extracting images	38
preprocessing	38
Feature extraction-classification	38
Conclusion.....	40

Chapter 03: simulationand results

I.Introduction	42
Data-set description.....	42
1.2 Evaluation metrics.....	43
Results and Discussion.....	44
Adam Results.....	44
SGD Results	46
RMSprop RESULTS.....	48
Confusion matrix	50
III. Conclusion	51
Final Conclusion.....	52

Figures list :

Figure1 process of face detection.....	2
Figure1.2 binary pictures.....	4
Figure1.3 lina grayscale.....	4
Figure 1.4 grayscale in 8-bit	4
Figure 1.5 lina RGB	5
Figure 1.6 Table ofsome pixel colors values	5
Figure 1.7 A zoom from a part a vector image after being scaled	5
Figure 1.8 A zoom for a part matrix image	6
Figure 1.9 Different pixel resolutions for logo university of Mohamed Boudiaf.....	7
Figure 1.10 Different between RGB application	7
Figure 1.11 RGB image color ordering	8
Figure 1.12 zoomed image of person face... ..	8
Figure 1.13 the representation of pixel values of zoomed part of RGB	9
Figure 1.14 Encryptionofthe symbol of university of Mohamed boudiaf	9
Figure 1.15 blockdiagram of the propos method	10
Figure 1.16 categories challenges	11
Figure 1.17 the difference between low and highcontrast	12
Figure 1.18 the difference in contrast between a low and highcontrast.....	13
Figure 1.19 blue plot indicating input image are indicating CLAHE intensities.....	14
Figur1.20 Machine learning explanation.....	15
Figure 1.21 Unsupervised machine learning	18
Figure 1.22 the difference between AI vs Machine learning vs Deep learning	18
Figure 1.23 system overview	19
Figure1.24: features in Haar cascade classifier.....	20
Figure1.25 application features Haar cascade in face person	21
Figure1.26: per-processed image divided into 64 regions	22
Figure1.27 The original lbp operation.....	23

Figure 1.28 Circularly neighbor-sets for three different values of P and R.....23

Figure1.29 SVM classifier.....24

Figure1.30 Linear case classifications.....25

Figure1.31 Nonlinear example.....26

Figure 2.1 one convolution layer30

Figure 2.2 convolutionlayer followed by pooling layer30

Figure 2.3 the convolutional layers followed by a fully connected layer31

Figure 2.4 an 4*4 gray-scale image31

Figure 2.5 a kernal of size 2*2... ..31

Figure 2.6 examples of RGB image32

Figure 2.7 illustrating the first 5 steps of convolution operation32

Figure 2.8 the final feature map after the complete convolution operation32

Figure 2.8 the computations performed.....33

Figure 2.10 sigmoid34

Figure 2.11 tant34

Figure 2.12 ReLU35

Figure 2.13 Leaky ReLU.....35

Figure 2.14Noisy ReLU36

Figure 2.15: facial expression.....37

Figure2.16 : proposed system38

Figure 2.17 : operations.....39

Figure 2.18: max pooling operation40

Figure 3.1:dataset images content42

Figure 3.2 loss / acc with accuracy 64,54 %45

Figure 3.3 loss / acc with accuracy 38,53 %47

Figure 3.4 loss / acc with accuracy 65,14 %49

Figure 3.5 Confusion matrix result50

List of Tables

Table3.1 the results of Adam optimizer	44
Table3.2 the results of sgd optimizer	46
Table3.3 the results of RMSprop optimizer	48
Table 3.4 classification rapport.....	50

General Introduction :

Facial expressions are fundamental components of human communication, conveying a wealth of emotional information and social cues. Accurate recognition and interpretation of facial expressions play a crucial role in various domains, including human-computer interaction, affective computing, psychology, and social robotics. Traditionally, face expression recognition has relied on manual feature engineering and shallow classifiers, which often struggle to capture the complex and subtle dynamics of facial expressions. However, recent advancements in deep learning techniques have revolutionized the field by offering powerful tools to automatically learn expressive features and classify facial expressions with remarkable accuracy.

Deep learning, a subfield of machine learning, encompasses a family of algorithms and architectures inspired by the structure and function of the human brain. It excels in automatically discovering intricate patterns and representations from raw data, making it an ideal candidate for face expression recognition. By leveraging deep neural networks, which consist of multiple layers of interconnected artificial neurons, deep learning models can learn hierarchical representations of facial expressions, capturing both local facial features and global contextual information.

The primary objective of this thesis is to investigate and develop a deep learning-based approach for face expression recognition. By utilizing the expressive capacity of deep neural networks, the thesis aims to overcome the limitations of traditional methods and achieve state-of-the-art performance in accurately detecting and classifying facial expressions. The proposed approach involves several key stages, including data preprocessing, feature extraction, network architecture design, model training, and performance evaluation.

Data preprocessing is an essential step to ensure the quality and consistency of the input data. Techniques such as face detection and alignment are employed to isolate and normalize facial regions, reducing variations due to head pose and image alignment. Additional preprocessing steps, such as illumination normalization and image augmentation, are applied to enhance the robustness and generalization capabilities of the models.

Feature extraction plays a critical role in capturing discriminative information from facial expressions. Deep learning architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are investigated to automatically extract expressive features from facial images or video sequences. CNNs excel in spatial feature extraction, capturing local patterns and facial landmarks, while RNNs are well-suited for modeling temporal dependencies and capturing dynamic changes in expressions over time.

The chosen network architecture is designed and optimized to strike a balance between model complexity and performance. Transfer learning techniques are explored to leverage pre-trained models on large-scale datasets, such as Image Net, to accelerate the training process and enhance the generalization capabilities of the models. Fine-tuning strategies are

employed to adapt the pre-trained models to the specific task of face expression recognition.

The developed models are thoroughly evaluated using a comprehensive dataset of labeled facial expressions, encompassing diverse individuals and a wide range of expressions. Various evaluation metrics, including accuracy, precision, recall, and F1-score, are utilized to assess the models' performance and compare them against existing state-of-the-art methods. The results highlight the efficacy and superiority of the deep learning-based approach in accurately recognizing facial expressions.

Additionally, this thesis investigates the impact of different factors, such as pose variations, occlusions, and individual differences, on the performance of the developed models. Strategies for handling these challenges, such as data augmentation, robust feature extraction, and ensemble methods, are explored to enhance the models' robustness and adaptability in real-world scenarios.

The outcomes of this research have significant implications for various fields, including human-computer interaction, affective computing, psychology, and social robotics. Accurate and real-time face expression recognition can enable natural and intuitive human-computer interfaces, personalized emotion-driven systems, and socially-aware robots. By leveraging the power of deep learning, this thesis aims to contribute to the advancement of face expression recognition techniques and pave the way for innovative applications in these domains.

In this Thesis we will discuss :

- **Chapter one** : What's is face recognition in general
- **Chapters two** : Face expression using convolutional neural network
- **Chapter three** : Simulation and results

Chapter 1 :

Face Recognition

“Face recognition unveils a world where identities are unlocked by a
glance”

I- Introduction :

Facial recognition is a biometric technology that uses artificial intelligence (AI) and computer algorithms to analyze and identify human faces. This technology works by capturing an image or video of a person's face, analyzing the facial features, and comparing them to a database of known faces to determine a match. Facial recognition technology is widely used in various industries, including security, law enforcement, marketing, and social media. In the security industry, facial recognition is used for access control, surveillance, and identification purposes. Law enforcement agencies use facial recognition to help identify suspects or missing persons. In marketing, facial recognition is used to analyze consumer behavior and preferences. On social media platforms, facial recognition is used for photo tagging and to suggest friends. The technology behind facial recognition is constantly evolving, and AI has played a significant role in advancing this technology. By using deep learning algorithms, facial recognition systems can now detect and analyze more complex facial features and accurately identify individuals even in challenging lighting conditions or when the image is blurry. While facial recognition technology offers many benefits, it has also raised concerns about privacy and potential misuse of personal data. It is important for companies and organizations to ensure that they are transparent about their use of this technology and comply with relevant regulations and standards to protect the privacy and security of individuals.

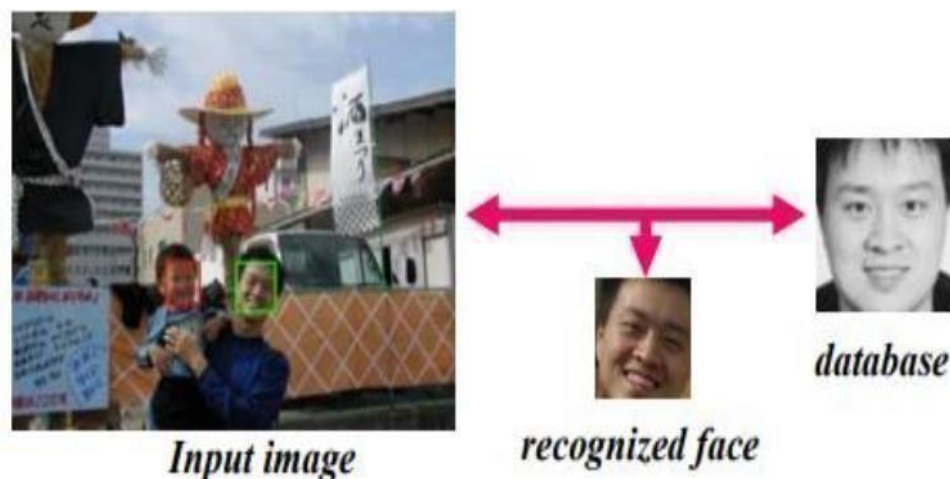


Figure 1.1 : process of face detection

1. Digital Images :

Digital images are everywhere, especially in our days, from the digital camera to the smart phone camera, sending and receiving images is an everyday task. In this chapter we are going to see what is a digital image and the types of digital images to get a basic understanding about it, then we are going to

see some image processing operations in order to get a basic understanding on how to process and perform some operations on these digital images. The images can convey the essence of a topic more effectively than words and description as the English idiom sentence says” an image is worth a thousand words”, the essential component of an image is the pixel, an image is a matrix of pixels. Mathematically an image is two-dimensional function of integer coordinates $N \times N$, these coordinates refers to an image element values (pixels) P, such that

$$I(u, v) \in P \text{ and } u, v \in N$$

A digital image can be represented as an array of values contains the value of each pixel, with a length equals to the dimensions of the image (array length = width*height)

2. Images types :

2.1 Based on coloring :

Based on coloring images are divided into three types.[1]

2.1.1 Binary image:

A binary image is an image that each pixel value is either 1 (white) or 0 (black) shows a binary image on the left, and on the right a representation of a zoom of a small part of the image to the pixel level to get a good idea how the image is composed.

Digital images are everywhere, especially in our days, from the digital camera to the smart phone camera, sending and receiving images is an everyday task. In this chapter we are going to see what is a digital image and the types of digital images to get a basic understanding about it, then we are going to see some image processing operations in order to get a basic understanding on how to process and perform some operations on these digital images The images can convey the essence of a topic more effectively than words and description as the English idiom sentence says” an image is worth a thousand words”, the essential component of an image is the pixel, an image is a matrix of pixels. Mathematically an image is two-dimensional function of integer coordinates $N \times N$, these coordinates refers to an image element values (pixels), such that

$$I(u, v) \in P \text{ and } u, v \in N$$

A digital image can be represented as an array of values contains the value of each pixel, with a length equals to the dimensions of the image (array length = width*height).



Figure 1.2:binary pictures

0	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

2.1.2 Grayscale image :

In Our work, we have studied the of 8-bit plan, so the value of each pixel is between 0 and 255 (totalwhite to total black), meaning 256 shades or degree of gray color, that is $2^8 = 256$. [1]

an image of Lena in grayscale while shows all possible 256bins



Figure 1.3: Lena grayscale



Figure 1.4:grayscale in 8_bit

2.1.3 True color image :

Unlike the grayscale image space color, the RGB image codes each pixel by three values, red,value and the blue value. [1]

. shows Lena in RGB mode, while Picture. shows a table of some popular colors used in digital images andtheir pixelrange.



Figure 1.5:Lena RGB

Color name	RGB triplet	Color
Red	(255, 0, 0)	
Lime	(0, 255, 0)	
Blue	(0, 0, 255)	
White	(255, 255, 255)	
Black	(0, 0, 0)	
Gray	(128, 128, 128)	
Fuchsia	(255, 0, 255)	
Yellow	(255, 255, 0)	
Aqua	(0, 255, 255)	
Silver	(192, 192, 192)	
Maroon	(128, 0, 0)	
Olive	(128, 128, 0)	
Green	(0, 128, 0)	
Teal	(0, 128, 128)	
Navy	(0, 0, 128)	
Purple	(128, 0, 128)	

Figure1.6: table of some pixel colors values

2.1 Based on the nature:

2.2.1 Vector graphics (images)

This type of images can be represented mathematically (straight lines, circles, points, ...) because it is composed of geometric shapes, as shown in picture, which is a vector image with extension [s1] you can scale it without losing any information, thus, it very useful when you are working on applications that frequently require a deep manipulation and adjusting the object content. A simple example of vector graphics that is used in most people's daily life is the graphic shapes in Word, no matter how much you scale its size, it will keep its details.



Figure1.7: A zoom for a part of a vectorimage after being scaled

This type of graphics has its own negative points, for instance some manipulations such as color changes are not easy and difficult on an area of an object, on a single object, or on a group of objects as

shown in the zoomed part on the right of picture., but in the positive way a vector file is much more compact than a bitmap file. Its size relies on the content of the image, not to its resolution, however, it took a high resource on the digital image (memory) to displayed anyobject.

2.2.2 Matrix graphics (images)

The matrix representation is the most popular type used in our days, and the most image encryption algorithms working on this type, this latter represents by a set of points(pixels), each one has its intensity (values), and also has specific spatial frequency information. In the literature, we found many formats of these images

- BMP: Windows Bit Map.
- TIFF: Tagged Image File Format.
- JPEG: Joint Photographic Expert Group.
- GIF: Graphics Interchange Format.
- PNG: Portable Network Graphic.

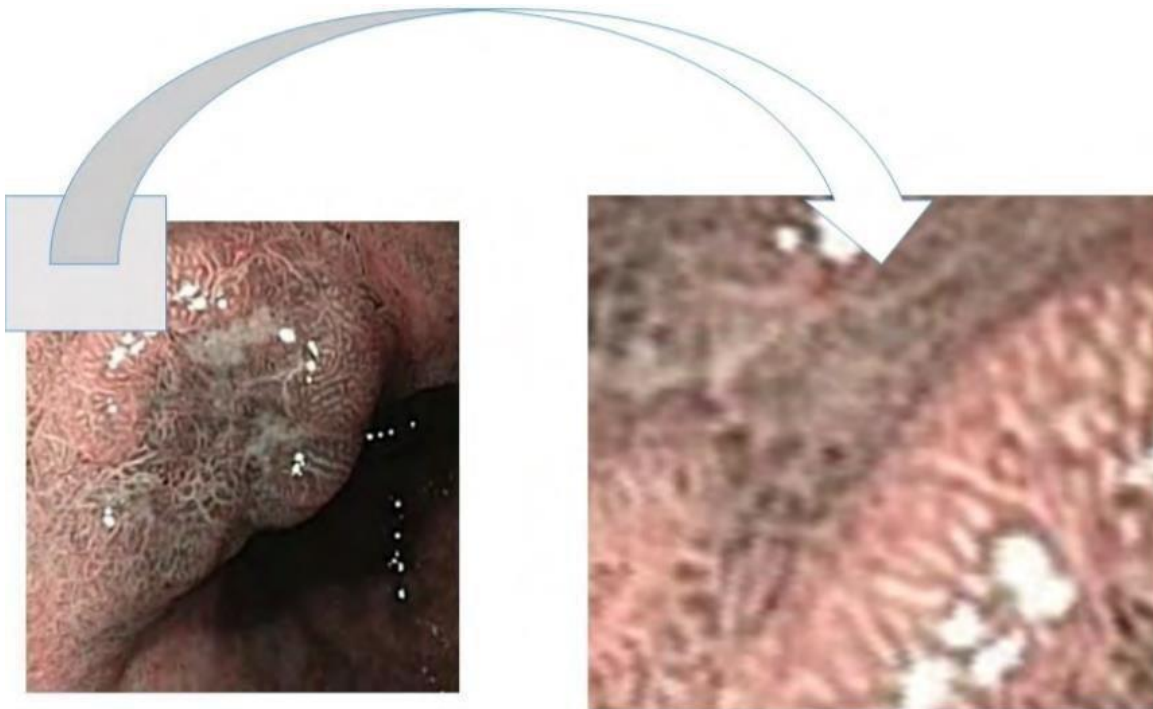


Figure1.8:A zoom for a part matrix image

3. Image characteristics :

3.1 Bit depth:

Bit depth is determined by the number of bits to represent each pixel in the image which translate to color or a grayscale value, so in binary images we have either black or white, because all binary images are bit depth, here we are going to work with are 8-bit depth, in other words, we have 255 possible value (in RGB images, each pixel is coded with 24 bit depth, so a total of 2^{24} -bits possible value.

3.2 Image size and resolution :

The size of an image is computed directly from the number of pixels along the width M (number of columns) and height N (number of rows) of the image matrix I . [2]In general, the resolution means the amount of number pixelused to represent the details, edge, foreground and background of image. [3]

The pixel density, it can be measured using pixels per inch (PPI), dots per inch (DPI), or pixels per centimeter (PPCM). Those are common terms used to express measurements of the resolution for digital images.

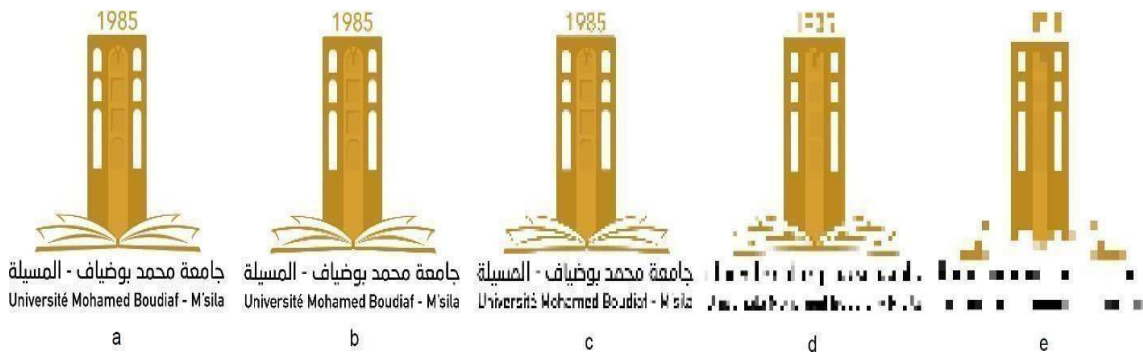


Figure1.9:Different Pixel Resolutions for logo of university of Mohamed boudiaf M’sila.

3.3 RGB channels and space color :

The basic space color of image refers to the RGB space, which each pixel represents by the combination of three channels: R, G and B. The 0) depicts an image with RGB space and . RGB **image color ordering** depicts the three channel separately

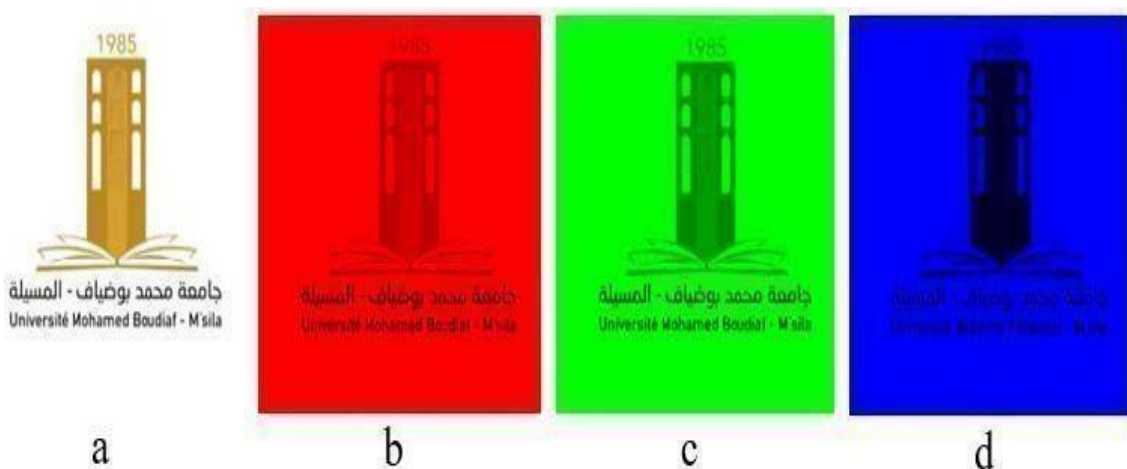


Figure1.10:(a)Original image, (b), (c), and (d) components Red, Green and Blue, respectively

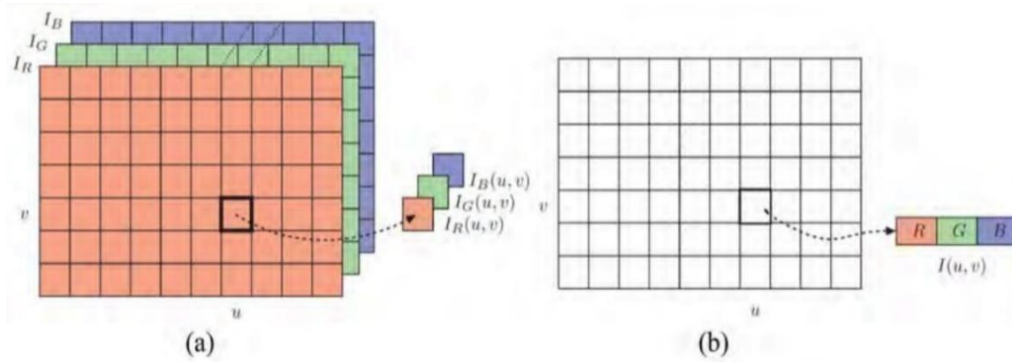


Figure 1.11 RGB image color ordering

Figure 1.11 RGB **image color ordering** (b) shows an example matrix representation of an image, the pixel I at the position (u,v) is composed of three values, one for R (red), and one for G (green), and the last for B (blue), so for example:

if $I_{(u,v)} = [67,67,67]$ then: we can get the three channels as:

$$I_{R(u,v)} = [67,0,0], I_{G(u,v)} = [0,67,0], I_{B(u,v)} = [0,0,67]$$

4. Image processing operations :

In real time application, the image may manipulate by different rules of matrix operations like addition, subtraction, multiplication and division. On other hand, we can find different format of matrix as binary image (in a case of pixels are 0 or 1), a gray image (in a case of one matrix), color image, RGB image (in a case of Multi-matrices) and image palette or multi-color (tensor matrix) . In addition, the image can be manipulated by logic operator and filters in order to extract or enhance its quality regarding and also very crucial for image security applications. For example, image encryption is a manipulation from a plain image to a random image or unclear image that cannot lead us visually to the original ones.

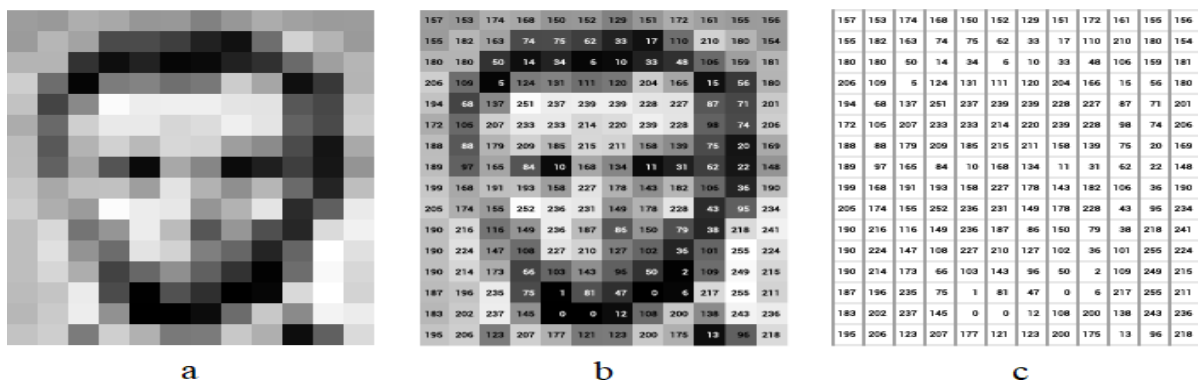


Figure 1.12: (a) zoomed image of a person's face, (b) same image with the value of each pixel, (c) the matrix of pixels values of the image (a)

with the value of each pixel on top of it, (c) shows each pixel value of the image (a) organized in a matrix, naturally with the same dimensions of the original image.

In case of an RGB image, they are going to split the image into three matrices, red, green and blue like as is explained in previous section.

Figure 1.13 the representation of pixel values of a zoomed part of an RGB image shows an example of pixel values representation of small zone of RGB image .

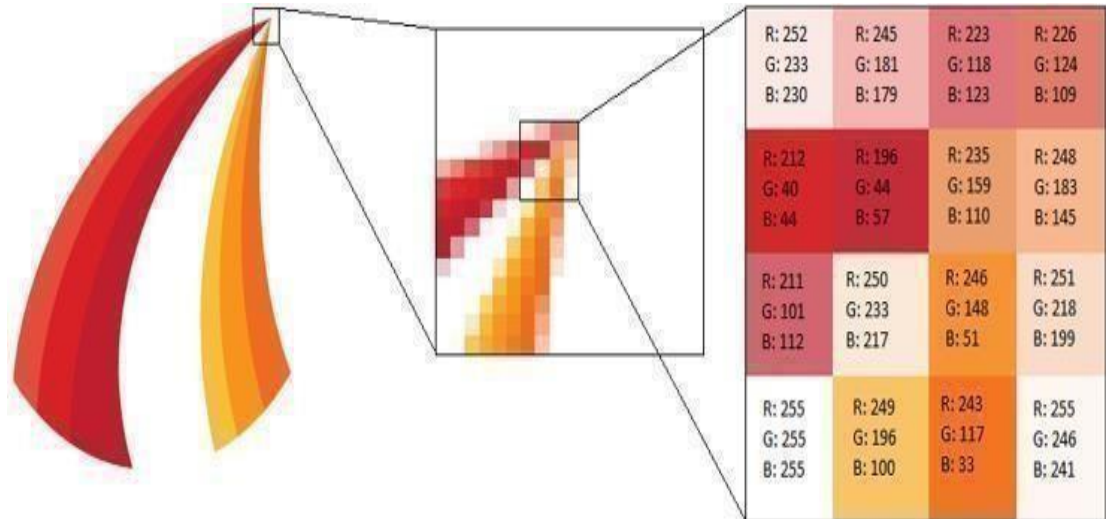


Figure 1.13 the representation of pixel values of a zoomed part of an RGB image

Figure 1.14: Encryption of the symbol of university of Mohamed Boudiaf -M'sila shows an example of three images. The first image is the original, the second seems fully noisy, which we have done a number of manipulation of pixels values through a specific function, the third image is the recovered image after having applying the inverse function [29].

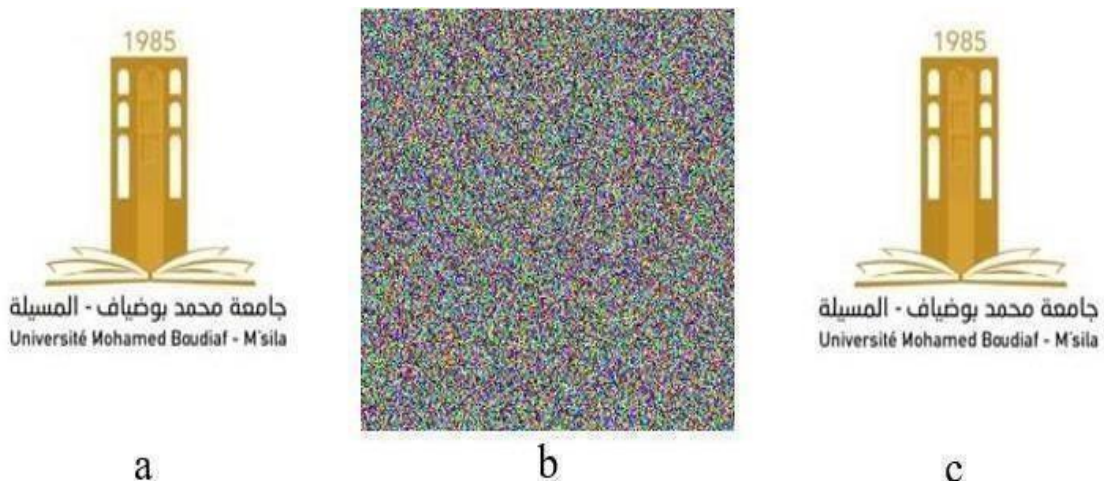


Figure 1.14: Encryption of the symbol of university of Mohamed Boudiaf -M'sila

we explain some basic proprieties about digital image and how to process it, it's very important to well cover the image propriety in order to well protect it and to develop an efficient encrypted image

5. Methodology of face detection :

The following figure represents the steps involved in the proposed approach in a sequence

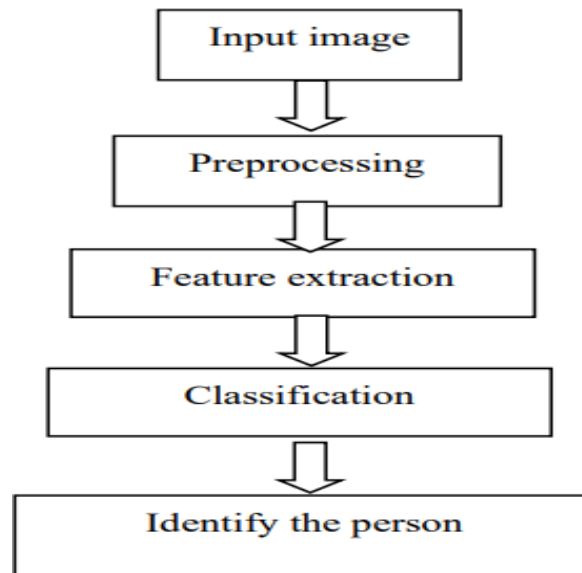


Fig 1.15:Block diagram of the propose method.

6. Challenges in face detection :

Challenges in face detection, are the reasons which reduce the accuracy and detection rate of face detection. These challenges are complex background, too many faces in images, odd expressions, illuminations, less resolution, face occlusion, skin color, distance and orientation etc. (Figure16)

- Odd expressions Human face in an image may have odd expressions unlike normal, which is challenge for face detection.

- Face occlusion Face occlusion is hiding face by any object. It may be glasses, scarf, hand, hairs, hats and another object etc. It also reduces the face detection rate.

- Illuminations Lighting effects may not be uniform in the image. Some part of the imagemay have veryhigh illumination and other mayhave verylow illumination.

-Complex background Complex background means a lot of objects presents in the image, which reduces the accuracy and rate of face detection.

- Too many faces in the image It means image contains too many human faces, which is challenge for face detection.
- Less resolution Resolution of image may be very poor, which is also challenging for face detection.
- Skin color Skin-color changes with geographical locations. Skin color of Chinese is different from African and skin-color of African is different from American and so on. Changing skin-color is also challenging for face detection.
- Distance Too much distance between camera and human face may reduce the detection rate of human faces in image.
- Orientation Face orientation is the pose of face with an angle. It also reduces the accuracy and detection rate of face detection



Figure1.16: categories challenges

7. Contrast:

contrast is an important factor in face recognition. High contrast images make it easier for the face recognition system to detect and extract facial features, such as eyes, nose, and mouth, and to distinguish them from the surrounding background. Low contrast images, on the other hand, can make it more difficult for the system to accurately detect and extract facial features, leading to lower recognition accuracy. Contrast in face recognition refers to the difference in brightness or intensity between the facial features and the surrounding background in an image. High contrast images have a greater difference in brightness between the facial features and the background, while low contrast images have a smaller difference. Contrast plays an important role in face recognition, as it can affect the ability of the recognition system to

accurately detect, extract, and match facial features. Facial features, such as eyes, nose, and mouth, are important for face recognition, as they contain unique characteristics that can be used to distinguish one individual from another. In high contrast images, these facial features are more distinguishable from the surrounding background, making it easier for the face recognition system to detect and extract them. In low contrast images, the facial features can blend into the background, making it more difficult for the system to accurately detect and extract them.

This image below explain the difference between low contrast (left) and high contrast (right) :

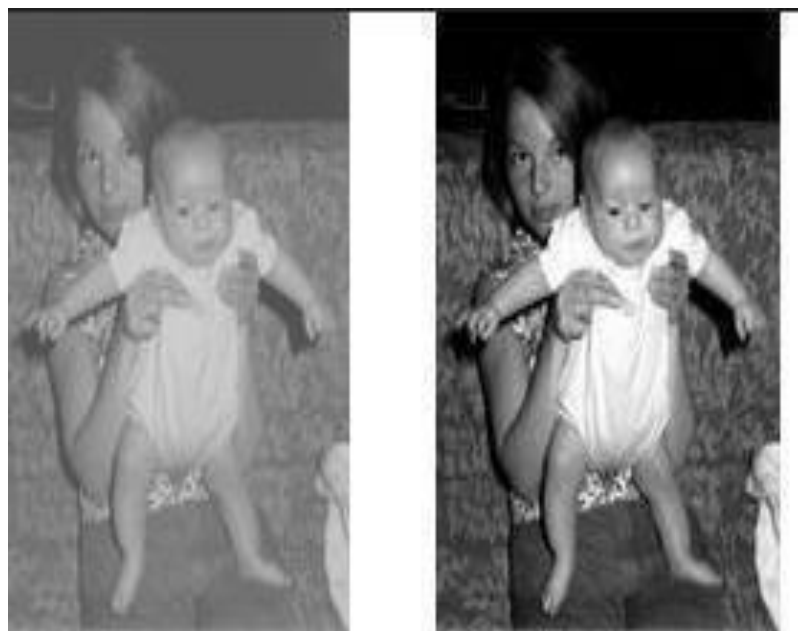


Figure1.17: the difference between low contrast (left) and high contrast (right)

In the high contrast image on the left, the facial features are clearly visible and distinguishable from the background, making it easier for the face recognition system to accurately detect and extract them. In the low contrast image on the right, the facial features are less distinguishable from the background, making it more difficult for the system to accurately detect and extract them.

Image processing techniques, such as Histogram Equalization (HE), Adaptive Histogram Equalization (AHE), and Contrast-Limited Adaptive Histogram Equalization (CLAHE), can be used to enhance the contrast of facial features in low contrast images. These techniques adjust the pixel intensity distribution of an image to improve the contrast and make the facial features more distinguishable. The figure below illustrates the difference in contrast between a low contrast image (left) and a low contrast image after applying CLAHE (right):



Figure 1.18: the difference in contrast between a low contrast and high contrast

In the low contrast image on the left, the facial features are not clearly visible and are difficult to distinguish from the background. However, after applying CLAHE to the image on the right, the facial features are more distinguishable and can be more easily detected and extracted by the face recognition system

In summary, contrast is an important factor in face recognition, as it can affect the accuracy and robustness of the recognition system. High contrast images make it easier for the system to accurately detect and extract facial features, while low contrast images can make it more difficult. Image processing techniques, such as HE, AHE, and CLAHE, can be used to enhance the contrast of facial features in low contrast images and improve the accuracy of the recognition system.

$$\text{Contrast} = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}}$$

Where I_{\max} is the highest and I_{\min} is the lowest luminance. In a digital image, 'luminance' is a value that goes from 0 (black) to a maximum value depending on color depth. In case of typical 8-bit images i.e. grayscale, the value is $2^8 - 1 = 255$, since this is the number of combinations, one can achieve with 8 bits sequences, assuming 0-1 values for each. To perform CLAHE we need to take the input image I , number of bins n , minimum intensity \min , maximum intensity \max , window size ht_w , wd_w and clip limit clip as input parameters and return an output image with new intensities after CLAHE.

The following plot shows the input and output intensities of our input image

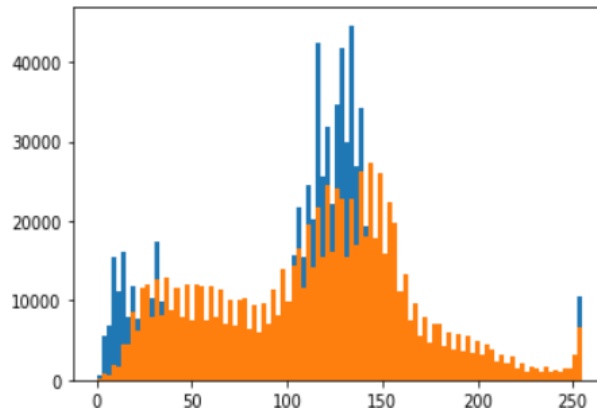
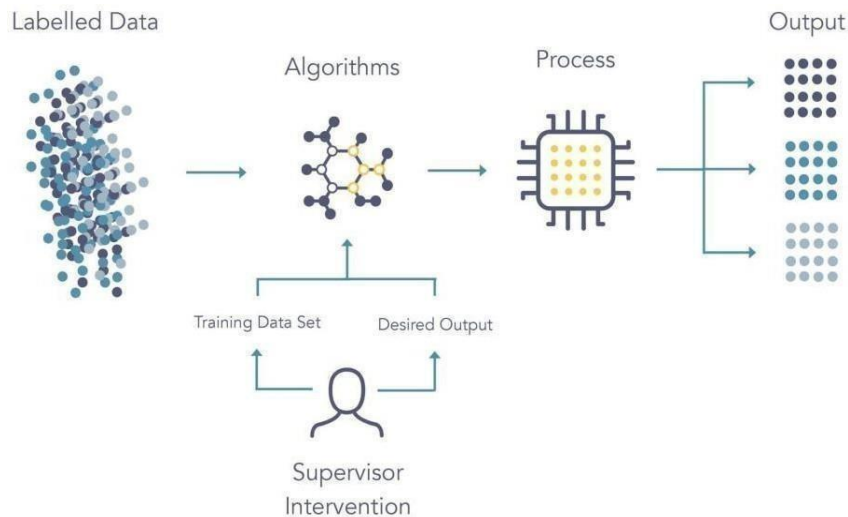


Figure1.19:Blue plot indicating input image intensities and orange are indicatingCLAHE

8. Machine learning

8.1 definition :

Machine learning is a sub-field of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers or machines to learn and make predictions or decisions without being explicitly programmed. It involves the study of statistical models and algorithms that allow computer systems to automatically improve their performance on a specific task by learning from data and experience. In machine learning, a model is trained on a set of data called the training data, which consists of input examples and their corresponding correct output or target values. The model learns patterns and relationships within the data and uses this knowledge to make predictions or decisions on new, unseen data. The process of training a machine learning model involves alliteratively adjusting its internal parameters based on the training data to minimize the difference between the predicted outputs and the actual outputs. This optimization process is typically guided by a specific objective or loss function, which quantifies the model's performance. Machine learning techniques can be broadly categorized into supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the model learns from labeled examples, where both input and output are known. In unsupervised learning, the model learns patterns and structures in unlabeled data. Reinforcement learning involves training an agent to interact with an environment and learn optimal actions based on rewards or punishments. Machine learning has applications in various domains, including image and speech recognition, natural language processing, recommendation systems, fraud detection, autonomous vehicles, and many others. It has become an essential tool for extracting insights and making predictions from large and complex datasets



Figur1.20 Machine learning explanation

8.2 Learning Types of machine Algorithms :

8.2.1 Supervised learning :

Supervised learning is a machine learning technique where an algorithm learns from labeled training data to make predictions or decisions. It involves two main components: input features (X) and corresponding output labels or target values (y). The algorithm learns to map the input features to the output labels by finding patterns and relationships in the training data.

Here's a step-by-step explanation of supervised learning:

Step 1: Data-set Preparation

In supervised learning, we start with a labeled data-set that contains examples with input features and their corresponding output labels. The input features represent the characteristics or attributes of the data points, while the output labels indicate the desired outcome or prediction. For example, consider a dataset of housing prices, where the input features could be the size, number of bedrooms, and location of houses, and the output labels could be the corresponding prices

Step 2: Model Training

We select a suitable supervised learning algorithm based on the nature of the problem and the available data. Common algorithms include linear regression, decision trees, random forests, support vector machines (SVM), and neural networks.

The algorithm is trained by feeding it the labeled training data. During training, the algorithm learns the underlying patterns and relationships between the input features and output labels.

Step 3: Model Evaluation

Once the model is trained, we evaluate its performance using a separate set of data called the test set. The test set contains examples that the model has not seen during training. By evaluating the model on unseen data, we can assess how well it generalizes to new, unseen examples. Common evaluation metrics for supervised learning models include accuracy, precision, recall, score, and mean squared error (MSE), depending on the nature of the problem (classification or regression).

Step 4: Prediction

After evaluating the model's performance, we can use it to make predictions on new, unseen data. We input the features of the new data point into the trained model, and it outputs a predicted label or value based on its learned patterns from the training data.

For example, if we have a new house with its size, number of bedrooms, and location, we can input these features into the trained housing price prediction model, and it will output an estimated price for the house. Supervised learning is widely used in various applications, such as image classification, sentiment analysis, spam filtering, credit risk assessment, medical diagnosis, and many others. By learning from labeled data, supervised learning algorithms can make predictions or decisions based on the patterns and relationships discovered in the training phase.

8.2.2 Unsupervised learning:

Unsupervised learning is a machine learning approach where the algorithm learns from unlabeled data without any specific output labels or target values. The goal is to discover hidden patterns, structures, or relationships in the data without prior knowledge or guidance. Here's a step-by-step explanation of unsupervised learning:

In unsupervised learning, we start with a dataset that contains only input features (X) without any corresponding output labels. The input features represent the characteristics or attributes of the data points.

For example, consider a dataset of customer purchasing behavior, where the input features could be the types of products purchased, the frequency of purchases, and the total amount spent by each customer.

Step 2: Model Training

We select an appropriate unsupervised learning algorithm based on the specific problem and the characteristics of the data. Common unsupervised learning algorithms include clustering and dimensional reduction techniques.

Clustering: Clustering algorithms group similar data points together based on their similarities or distances in the feature space. Examples of clustering algorithms include k-means clustering, hierarchical clustering, and DBSCAN. The algorithm automatically discovers clusters in the data, grouping together data points that are more similar to each other

Dimensional Reduction: Dimensional reduction techniques aim to reduce the number of input features while preserving the important information or structure in the data. These techniques help to deal with high-dimensional data and eliminate irrelevant or redundant features. Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) are commonly used dimensional reduction techniques.

Step 3: Model Evaluation (optional)

Unlike supervised learning, where we evaluate the model's performance using labeled data, evaluating unsupervised learning models can be challenging since there are no explicit labels. Instead, evaluation in unsupervised learning often involves qualitative assessment or domain-specific criteria. For clustering algorithms, we can visually inspect the clusters and assess their coherence and separation. In dimensional reduction, we can examine how well the reduced-dimensional representation preserves the important structure of the original data.

Step 4: Inference and Insight Generation

Once the unsupervised learning model is trained, we can use it to gain insights and extract useful information from the data. In clustering, we can assign new, unlabeled data points to the discovered clusters, enabling us to group similar data points together and identify patterns or segments within the data. In dimensional reduction, we can visualize the reduced-dimensional representation of the data, which can help us understand the underlying structure or relationships between data points. Unsupervised learning finds applications in various domains, such as customer segmentation, anomaly detection, recommendation systems, and data visualization. It allows us to uncover hidden patterns or structures in the data and gain a deeper understanding of the underlying information without relying on explicit labels or targets

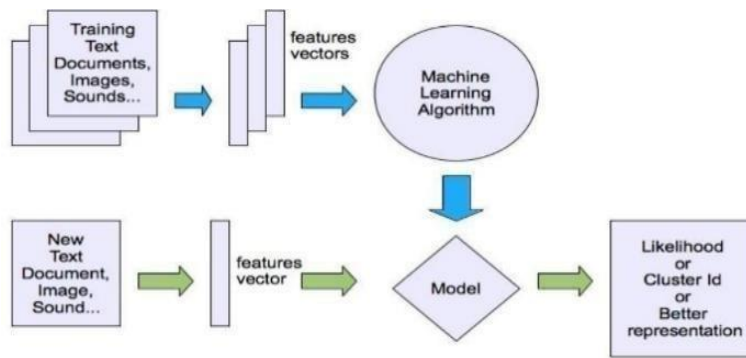


Figure 1.21 Unsupervised machine learning

9 Deep learning

9.1 Definition

DL is a class of ML algorithms that use multiple layers to extract functions of a progressively higher level from an elementary entry. By example, in image processing, the lower layers can define edges, while that the higher layers can define human concepts such as numbers, letters or faces. DL is based on what are called, by analogy, "artificial neural networks", made up of thousands of "neuron" units that each perform small and simple operations

9.2 Deep learning position in AI

Currently, there are three widely used terms - artificial intelligence (AI), machine learning (ML), and deep learning (DL) - that are sometimes used interchangeably to describe intelligent systems or software. In Figure 2, we provide a visual representation of the relationship between deep learning, machine learning, and artificial intelligence. According to the figure, deep learning is a subset of machine learning, which in turn falls within the broader field of artificial intelligence. In general, artificial intelligence involves imbuing machines or systems with human-like behavior and intelligence [103]. On the other hand, machine learning is a methodology that enables systems to learn from data or experiences [97], thereby automating the construction of analytical models. Deep learning, as a specific approach within machine learning, involves learning from data using multi-layer neural networks and complex processing techniques. The term "deep" in deep learning refers to the idea of processing data through multiple levels or stages to construct a model driven by data.

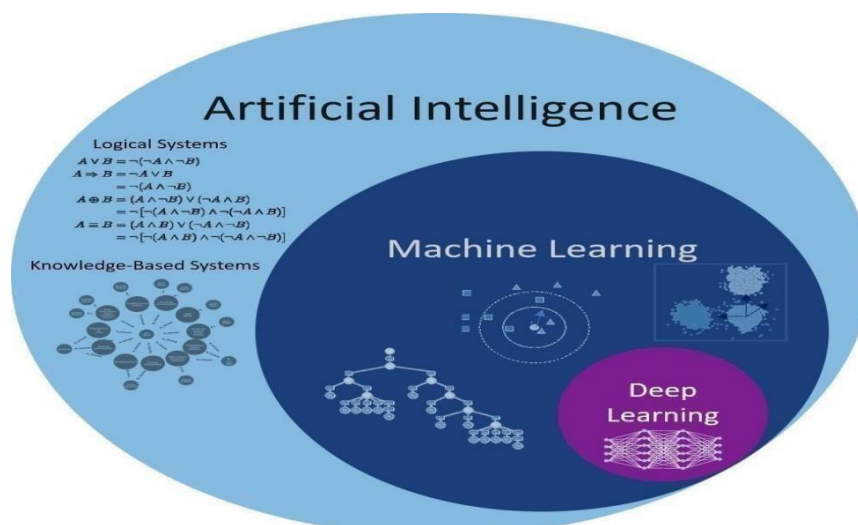


Figure 1.22 the difference between AI vs Machine learning vs Deep learning

10 Face Detection:

In the beginning of the 1970's, face recognition was treated as a 2D patten recognition

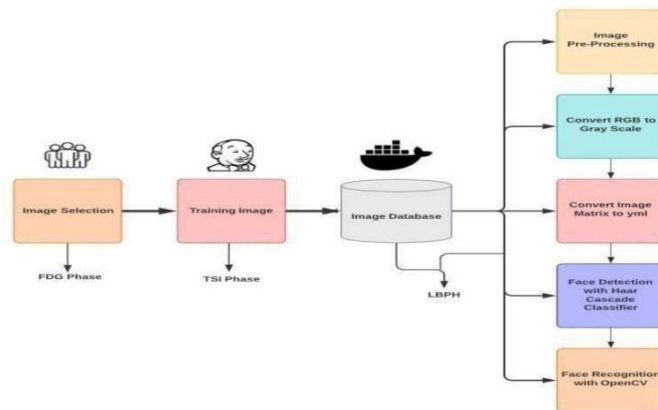


Figure1.23: System overview

Problem The distances between important points where used to recognize known faces, e.g. measuring the distance between the eyes or other important points or measuring different angles of facial components But it is necessary that the face recognition systems to be fully auto

Face recognition is such a challenging yet interesting problem that it has attracted researchers who have different backgrounds: psychology, pattern recognition, neural networks, computer vision, and computer graphics. The following methods are used to face recognition

10.1 Face detection methods :

1. Face Detection Using Haar Cascade Classifier
2. Face Detection Using LocalBinaryPattern Cascade Classifier
3. Support Vector Machine

10.2 Face Detection Using Haar Cascade Classifier:

Object Detection using Haar feature-based cascade classifiers is an effective method proposed by Paul Viola and Michael Jones in the 2001 paper, "Rapid Object Detection using a Boosted Cascade of Simple Features". It is a machine learning based approach in which a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.

There are some features in Haar cascade classifier

- Edge feature
- Line feature
- Four rectangle feature

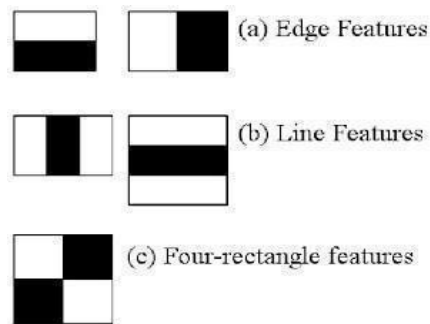


Figure1.24: features in Haar cascade classifier

Now all possible sizes and locations of each kernel are used to calculate plenty of features. For each feature calculation, we need to find the sum of the pixels under the white and black rectangles. To solve this, they introduced the integral images. It simplifies calculation of the sum of the pixels, how large may be the number of pixels, to an operation involving just four pixels. But among all these features we calculated, most of them are irrelevant. For example, consider the image below. Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or another place is irrelevant. So how do we select the best features out of 160000+ features? It is achieved by **Adaboost**.

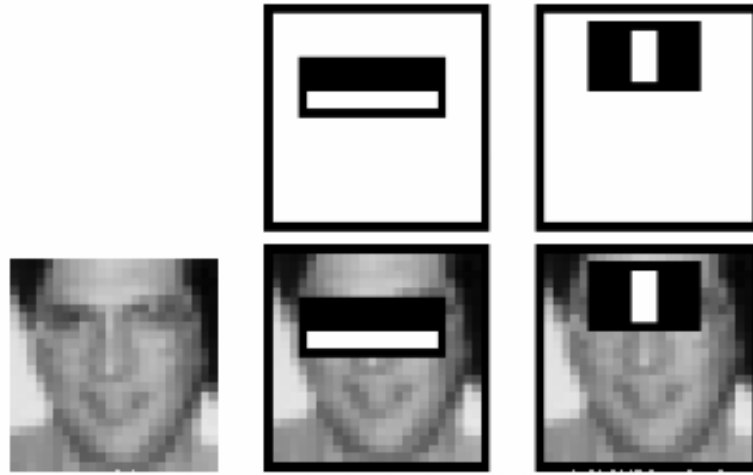


Figure1.25 application features Haar cascade in face person

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. But obviously, there will be errors or miss classifications. We select the features with minimum error rate, which means they are the features that best classifies the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of classified images are increased. Then again same process is done. New error rates are calculated. Also new weights. The process is continued until required accuracy or error rate is achieved or required number of features are found). Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features. (Imagine a reduction from 160000+ features to 6000 features. That is a big gain). So now you take an image. Take each 24x24 window. Apply 6000 features to it. Check if it is face or not. Wow.. Wow.. Isn't it a little inefficient and time consuming? Yes, it is. Authors have a good solution for that.

In an image, most of the image region is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot. Don't process it again. Instead focus on region where there can be a face. This way, we can find more time to check a possible face region. For this they introduced the concept of **Cascade of Classifiers**. Instead of applying all the 6000 features on a window, group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region. How is the plan !!! Authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in first five stages. (Two features in the above image is actually obtained as the best two features from Adaboost). According to authors, on an average, 10 features

out of 6000+ are evaluated per sub-window. So this is a simple intuitive explanation of how Viola-Jones face detection works

10.3 Face Detection Using Local Binary Pattern Cascade

10.3.1 Classifier (LBP):

the Local Binary Pattern (LBP) method. This relative new approach was introduced in 1996 by Ojala et al. [5]. With LBP it is possible to describe the texture and shape of a digital image. This is done by dividing an image into several small regions from which the features are extracted

These features consist of binary patterns that describe the surroundings of pixels in the regions. The obtained features from the regions are concatenated into a single feature histogram, which forms a representation of the image. Images can then be compared by measuring the similarity (distance) between their histograms. According to several studies [2, 3, 4] face recognition using the LBP method provides very good results, both in terms of speed and discrimination performance. Because of the way the texture and shape of images is described, the method seems to be quite robust against face images with different facial expressions, different lightening conditions, image rotation and aging of persons.



Figure1.26: per-processed image divided into 64 regions

10.3.2 Principles of local binary patterns :

The original LBP operator was introduced by Ojala et al. This operator works with the eight neighbors of a pixel, using the value of this center pixel as a threshold. If a neighbor pixel has a higher gray value than the center pixel (or the same gray value) than a one is assigned to that pixel, else it gets a zero. The LBP code for the center pixel is then produced by concatenating the eight ones or zeros to a binary code

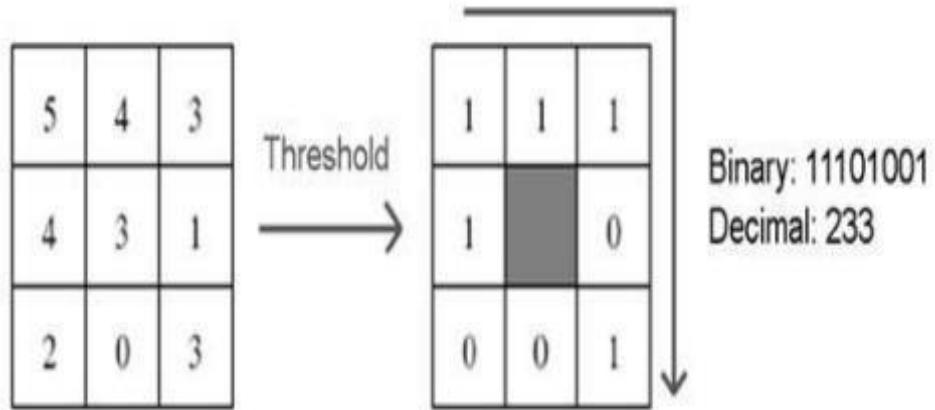


Figure 1.27 The original lbp operation

Later the LBP operator was extended to use neighborhoods of different sizes. In this case a circle is made with radius R from the center pixel. P sampling points on the edge of this circle are taken and compared with the value of the center pixel. To get the values of all sampling points in the neighborhood for any radius and any number of pixels, (bi linear) interpolation is necessary. For neighborhoods the notation (P, R) is used. Figure 1.4 illustrates three neighbor-sets for different values of P and R

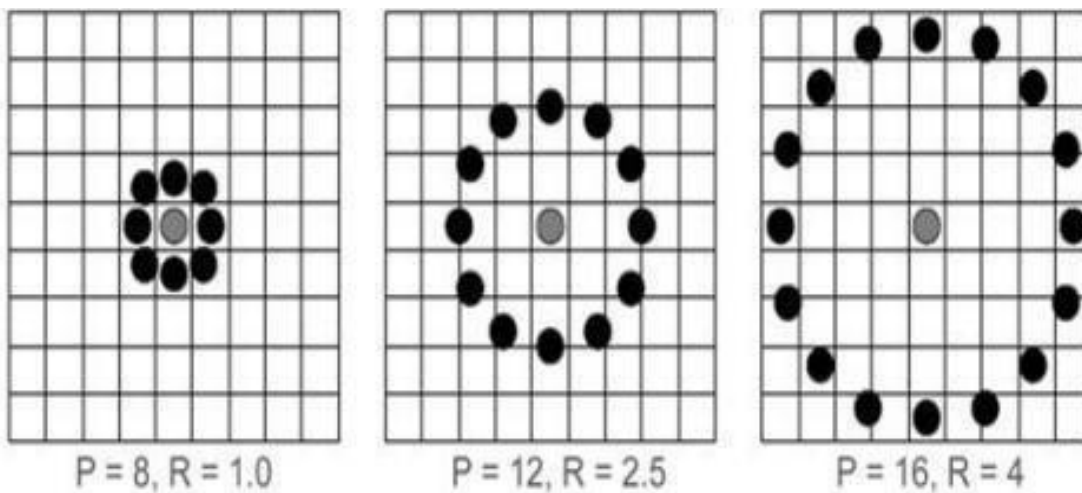


Figure 1.28 Circularly neighbor-sets for three different values of P

10.4 Support Vector Machine(SVM):

Support Vector Machines (SVMs) have been recently proposed by Vapnik and his co-workers [17] as a very effective method for general purpose pattern recognition. Intuitively, given a set of points belonging to two classes, a SVM finds the hyperplane that separates the largest possible fraction of points of the same class on the same side, while maximizing the distance from either class to the hyperplane. According to Vapnik [17], this hyperplane is called Optimal Separating Hyperplane (OSH) which minimizes the risk of miss classifying not only the examples in the training set but also the unseen examples of the test set. The application of SVMs to computer vision problem have been proposed recently.

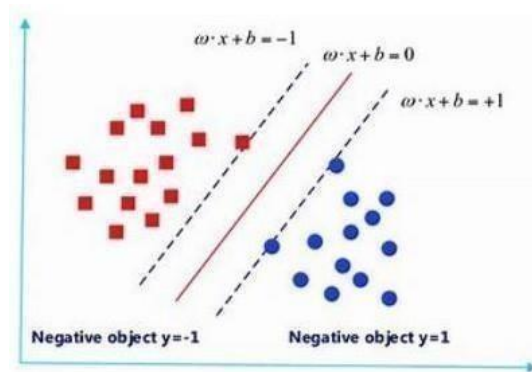


Figure1.29 SVM classifier

10.5 Types of SVM :

10.5.1 linear case

Consider a set of vectors $\{x_i\}$, $x_i \in \mathbb{R}^n$, $1 \leq i \leq l$, representing input samples and set of labels $\{y_i\}$, $y_i \in \{\pm 1\}$, that divide input samples into two classes, positive and negative. If the two classes are linearly separable, there exists a separating hyper plane (w, b) defining the function :

$$f(x) = \langle w \cdot x \rangle + b, \quad (1)$$

and $\text{sgn}(f(x))$ shows on which side of the hyper plane x rests, in other words - the class of x . Vector w of the separating

hyper plane can be expressed as a linear combination of x_i (often called a dual representation of w) with weights α_i

$$W = \sum_{1 < i < l} \alpha_i y_i x_i \quad (2)$$

The dual representation of the decision function $f(x)$ is then:

$$F(x) = \sum_{1 \leq i \leq l} a_i y_i \langle x_i, x \rangle + b \quad (3)$$

Training a linear SVM means finding the embedding strengths $\{a_i\}$ and offset b such that hyperplane (w, b) separates positive samples from negative ones with a maximal margin. Notice that not all input vectors $\{x_i\}$ might be used in the dual representation of w ; those vectors x_i that have weight $a_i > 0$ and form are called support vectors

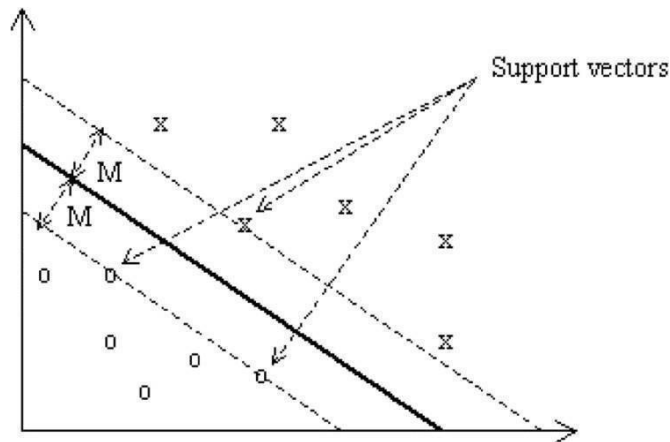


Figure 1.30 Linear case classifications

10.6 Non linear case :

In real-life problems it is rarely the case that positive and negative samples are linearly separable.

Non-linear support

vector classifiers map input space X into a feature space F via a usually non-linear map $\phi : X \rightarrow F, x \mapsto \phi(x)$ and solve the linear separation problem in the feature space by finding weights a_i of the dual expression of the separating hyperplane's vector w :

$$w = \sum_{1 \leq i \leq l} a_i y_i \phi(x_i) \quad (4)$$

while the decision function $f(x)$ takes the form:

$$F(x) = \sum_{1 \leq i \leq l} a_i y_i \langle \phi(x_i), x \rangle + b \quad (5)$$

Usually F is a high-dimensional space where images of training samples are highly separable, but working directly in such a space would be computationally expensive. However we can choose a space F which is induced by kernel K , defined by a kernel function $K(x, y)$ that computes the dot product in F , $K(x, y) = \langle \phi(x) \cdot \phi(y) \rangle$. The decision function (5) can then be computed by just using the kernel function and it can also be shown that finding the maximum margin separating hyperplane is equivalent to solving the following optimization problem

$$\text{MAX} \left[\sum_{1 < i < l} a_i - \frac{1}{2} \sum_{1 < i < l} a_i a_j y_i y_j k(x_i, x_j) \right] \quad (6)$$

$$\text{Subject to } 0 < a_i < c, 1 < i < l \sum_{1 < i < l} a_i y_i = 0$$

where positive C is a parameter showing the trade-off between margin maximization and training error minimization.

This knowing the kernel function K we avoid working directly in feature space F . After solving (6), offset b can be chosen so that the margins between the hyper plane and the two classes of sample images are equal.

We then have our decision function :

$$\text{Sgn}(f(x)) = \text{sgn} \left[\sum_{1 < i < l} a_i y_i k(x_i, x) + b \right] \quad (7)$$

Commonly used kernels include polynomial kernels $K(x, y) = (x + y)^d$ and the Gaussian kernel $K(x, y) = \exp(-\|x - y\|^2 / \sigma^2)$.

In our implementation we use the Gaussian kernel, however one of the interesting points for further research is approaches for choosing an optimal kernel for the given input data.

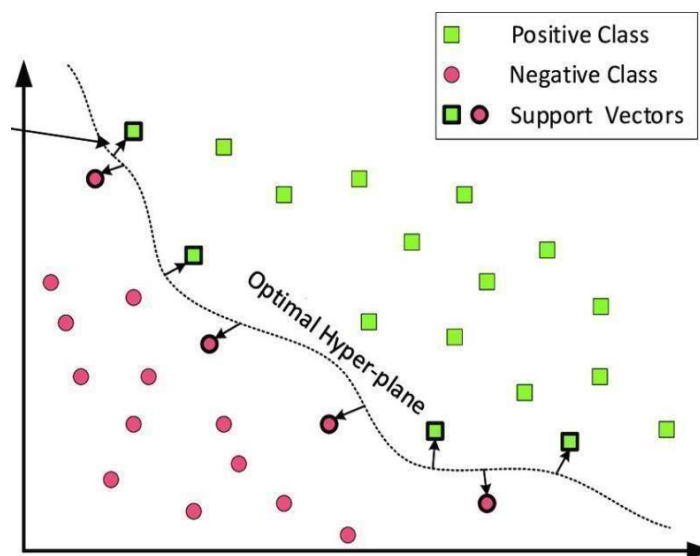


Figure1.31 Nonlinear example

Conclusion

In this chapter , we talked about images and digital images including their application and modalities , we also clarified the use of machine learning and deep learning methods in image and face recognition , including their applications and techniques , we talked to about face detection methods and techniques .

CHAPTER 02 :

Facial expression recognition using Convolutional neural
network
(CNN)

"Artificial intelligence is the most profound technology of our era"

Andrew Ng

1 Types of deep learning :

There is several types architectures we use in deep learning this is some of them RNNs are designed to handle sequential data, such as time series data or natural language. They have feedback connections that allow information to flow in cycles, enabling them to capture temporal dependencies and context. RNNs are commonly used for tasks like speech recognition, machine translation, and sentiment analysis.

1.1 Long Short-Term Memory (LSTM) Networks:

LSTMs are a specialized type of RNN that address the vanishing gradient problem, which can occur when training deep neural networks. LSTMs use memory cells and gating mechanisms to selectively retain and update information over long sequences, making them particularly effective for tasks that require modeling long-term dependencies, such as language modeling and speech recognition.

1.2 Generative Adversarial Networks (GANs):

GANs consist of two neural networks, a generator and a discriminator, which are trained in a competitive manner. The generator network generates synthetic data, while the discriminator network tries to distinguish between real and fake data. GANs have been successful in generating realistic images, video synthesis, and data augmentation.

1.3 Auto encoders:

Auto encoders are neural networks designed for unsupervised learning and data compression. They consist of an encoder network that maps the input data to a lower-dimensional latent space representation, and a decoder network that reconstructs the original data from the latent representation. Auto encoders can be used for tasks like data denoising, dimensionality reduction, and anomaly detection.

2. Convolutional Neural Networks (CNNs):

These networks are well-suited for processing and analyzing visual data, such as images and videos. CNNs use convolutional layers to detect patterns and features in different spatial scales, making them ideal for tasks like image classification, object detection, and image segmentation.

2.1 Convolutional Neural Networks layers :

2.1.1 Convolutional layer :

The convolution operation is one of the fundamental building blocks of a convolutional neural

filter is small spatially (along width and height), but extends through the full depth of the input volume. Typical filter sizes might have size 3x3, 5x5, 7x7. The third dimension of the filter corresponds to the number of channels in the input. The grayscale image depth is 1 and the color image has 3 (RGB) color channels

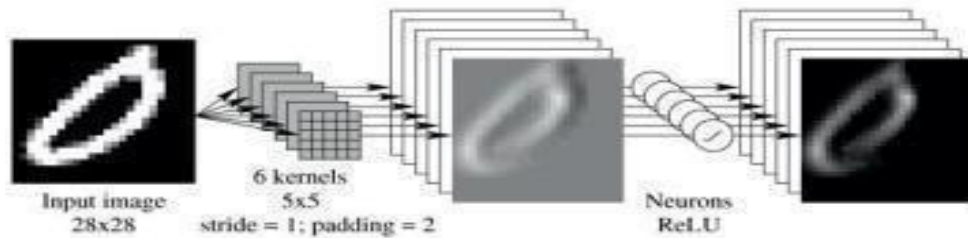


Figure 2.1:one Convolutional layer

During the forward propagation, each filter performs convolution on the input volume across the width and height and compute the dot products between the entries of the filter and the input at any position, this operation is followed by a nonlinear activation function (sigmoid, tanh, ReLU etc.), the resulting outputs are called feature maps. The feature map (also known as an activation map), gives the responses of the filter at every spatial position. An example of convolution layer followed by nonlinear activation is shown in Fig. 1. We stack these activation maps along the depth dimension and produce the output volume. The output volume depends on three hyper parameters: depth, stride and padding

- ◆ The depth of the output volume represents the number of filters that are used in the convolution operation. Each filter is learning something different in the input, edges, blobs, colors.
- ◆ The stride is the number of steps that we slide the filter in the input. When the stride is 1 then we move the filters one pixel at a time. When the stride is 2 then the filters jump 2 pixels at a time as we slide them around. This will produce smaller output volumes spatially.
- ◆ Padding allows controlling the output size. Applying convolution to an input, reduce the output size that leads to losing information. To avoid that, we pad the input volume with zeros around the border. Two common choices are valid convolution and the same convolution. The valid convolution means no padding, the same convolution means that the output size remains the same as the input size.

the output size is calculated in the following way:

$$(n + 2p - f) / s + 1$$

Where n is the number of filters, p is the amount of padding, f is the filter size and s is the stride.

2.1.2 Pooling layer :

CNNs often use pooling layer operation after convolution layers, its function is to reduce the dimension, also referred as sub-sampling or down sampling. Hyper parameters of pooling layer represent the filter size and strides. Most commonly used pooling layer is with filter size 2 and with stride 2. Two common types of pooling layers are max pooling and average pooling, where the maximum and average value is taken, respectively. Max pooling is used often than average pooling. Pooling layer does not have parameters to learn. The intuition of what max pooling is doing is that the large number means that there may be detected a feature. An example of convolution layer followed by pooling layer is shown

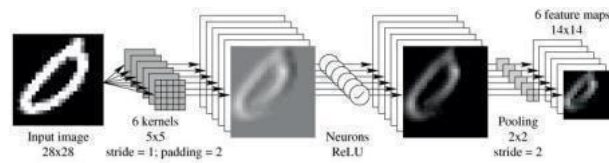


Figure2.2: Convolution layer followed by pooling layer

2.1.3 Fully Connected layer

After several convolution and pooling layers, the CNN generally ends with several fully connected layers. The tensor that we have at the output of these layers is transformed into a vector and then we add several neural network layers. The fully connected layers typically are the last few layers of the architecture as shown in the Fig. 3, the dropout [16] regularization technique can be applied in the fully connected layers to prevent over-fitting. The final fully connected layer in the architecture contains the same amount of output neurons as the number of classes to be recognized.

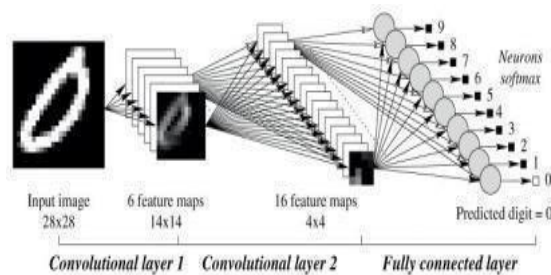


Fig2.3 Two convolutional layers followed by a fully connected layer3.

3. Convolution Operation:

The convolution operation is a fundamental mathematical operation used in various fields, including signal processing and image processing. It involves combining two functions to produce a third function

that represents how one function is modified by the other. Before we go any deeper, let us first understand the input format to CNN. Unlike other classical neural networks (where the input is in a vector format), in CNN the input is a multi-channelled image (e.g. for RGB image as it is 3 channelled and for Gray-Scale image, it is single channelled). Now, to understand the convolution operation, if we take a grayscale image of 4×4 dimension, shown in Fig.5 and an 2×2 kernel with randomly initialized weights as shown

1	0	-2	1
-1	0	1	2
0	2	1	0
1	0	0	1

0	1
-1	2

Figure 2.4: An 4×4 Gray-Scale image **Figure 2.5: A kernel of size 2×2**

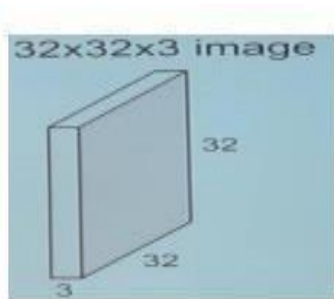


Figure 2.6: Example of a RGB image

Now, in convolution operation, we take the 2×2 kernel and slide it over all the complete 4×4 image horizontally as well as vertically and along the way we take the dot product between kernel and input image by multiplying the corresponding values of them and sum up all values to generate one scale value in the output feature map. This process continues until the kernel can no longer slide further. To understand the thing more clearly, let's do some initial computations performed at each step graphically as shown in Fig. 7, where the 2×2 kernel (shown in light blue color) is multiplied with the same sized region (shown in yellow color) within the 4×4 input image and the resulting values are summed up to obtain a corresponding entry (shown in deep blue) in the output feature map at each convolution step.

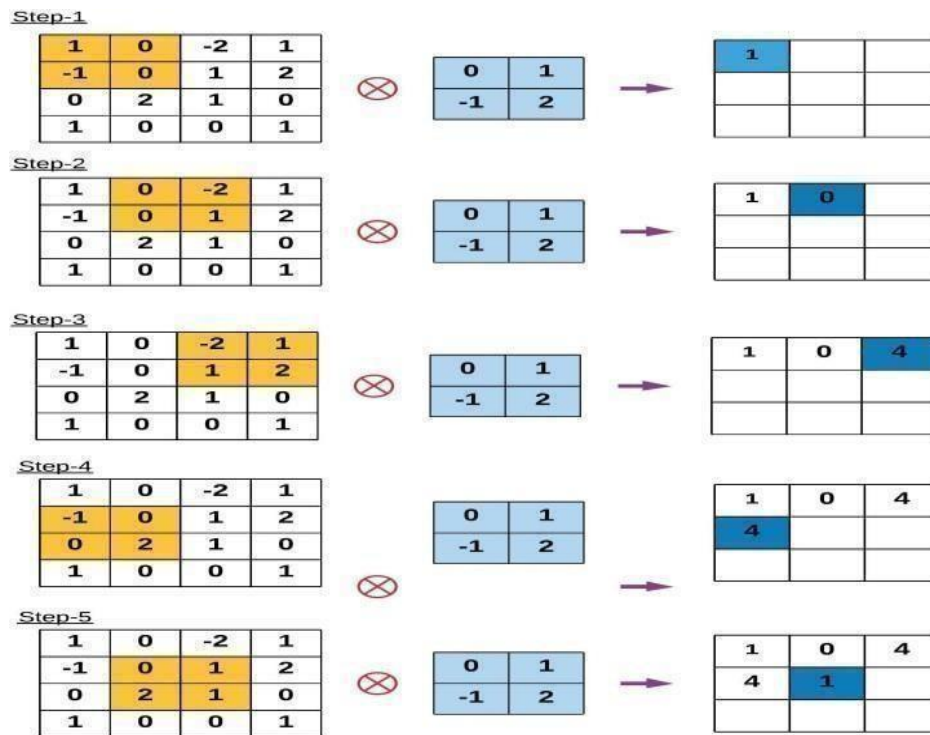


Figure 2.7: Illustrating the first 5 steps of convolution operation

After performing the complete convolution operation, the final output feature map is shown

1	0	4
4	1	1
1	1	2

Figure2.8: The final feature map after the complete convolution operation

In the above example, we apply the convolution operation with no padding to the input image and with stride (i.e. the taken step size along the horizontal or vertical position) of 1 to the kernel. But we can use other stride value (rather than 1) in convolution operation. The noticeable thing is if we increase the stride of the convolution operation, it resulted in lower-dimensional feature map. The padding is important to give border size information of the input image more importance, otherwise without using any padding the border side features are gets washed away too quickly. The padding is also used to increase the input image size, as a result the output feature map size also get increased. The Fig.41 gives an example by showing the convolution operation with Zero-padding and 3 stride value. The formula to find the output feature map size after convolution operation as below:

$$h' = \frac{h-f+p}{s} + 1$$

$$w' = \frac{w-f+p}{s} + 1$$

Where h' denotes the height of the output feature map, w' denotes the width of the output feature map, h denotes the height of the input image, w denotes the width of the input image, f is the filter size, p denotes the padding of convolution operation and s denotes the stride of convolution operation.

Step-1

0	0	0	0	0	0
0	1	0	-2	1	0
0	-1	0	1	2	0
0	0	2	1	0	0
0	1	0	0	1	0
0	0	0	0	0	0

⊗

1	0	1
0	-1	2
-2	1	0

→

-2	

Step-2 :

0	0	0	0	0	0
0	1	0	-2	1	0
0	-1	0	1	2	0
0	0	2	1	0	0
0	1	0	0	1	0
0	0	0	0	0	0

⊗

1	0	1
0	-1	2
-2	1	0

→

-2	-1

Step-3 :

0	0	0	0	0	0
0	1	0	-2	1	0
0	-1	0	1	2	0
0	0	2	1	0	0
0	1	0	0	1	0
0	0	0	0	0	0

⊗

1	0	1
0	-1	2
-2	1	0

→

-2	-1
1	

Step-4 :

0	0	0	0	0	0
0	1	0	-2	1	0
0	-1	0	1	2	0
0	0	2	1	0	0
0	1	0	0	1	0
0	0	0	0	0	0

⊗

1	0	1
0	-1	2
-2	1	0

→

-2	-1
1	0

Figure 2.9: The computations performed at each step

4. Activation Functions:

The main task of any activation function in any neural network based model is to map the input to the output, where the input value is obtained by calculating the weighted sum of neuron's input and further adding bias with it (if there is a bias). In other words, the activation function decides whether a neuron will fire or not for a given input by producing the corresponding output. In CNN architecture, after each learnable layers (layers with weights, i.e. convolutional and FC layers) non-linear activation layers are used. The non-linearity behavior of those layers enables the CNN model to learn more complex things and manage to map the inputs to outputs non-linearly. The important feature of an activation function is that it should be differentiable in order to enable error back propagation to train the model. The most commonly used activation functions in deep neural networks (including CNN) are described below.

4.1 Sigmoid:

The sigmoid activation function takes real numbers as its input and bind the output in the range of $[0,1]$. The curve of the sigmoid function is of 'S' shaped. The mathematical representation of sigmoid is:

$$F(x) = 1 / (1 + e^{-x})$$

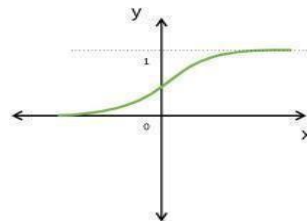


Figure2.10: sigmoid

4.2 Tanh:

The Tanh activation function is used to bind the input values (real numbers) within the range of $[-1,1]$. The mathematical representation of Tanh is:

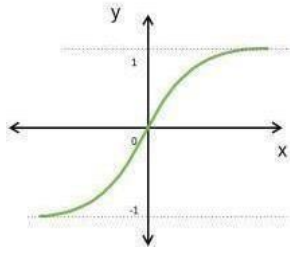


Figure2.11:Tant

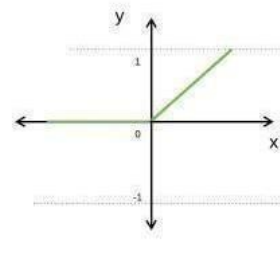


Figure2.12:ReLU

The Rectifier Linear Unit (ReLU)[25] is the most commonly used activation function in Convolutional Neural Networks. It is used to convert all the input values to positive numbers. The advantage of ReLU is that it requires very minimal computation load compared to others. The mathematical representation of ReLU is:

$$f(x)\text{ReLU} = \max(0, x)$$

But sometimes there may occur some major problems in using ReLU activation function. For example, consider a larger gradient is flowing during error back-propagation algorithm, and when this larger gradient is passed through a ReLU function it may cause the weights to be updated in such a way that the neuron never gets activated again. This problem is known as the Dying ReLU problem. To solve these types of problems there are some variants of ReLU is available, some of them are discussed below

4.3 Leaky ReLU:

Unlike ReLU, a Leaky ReLU activation function does not ignore the negative inputs completely, rather than it down-scaled those negative inputs. Leaky ReLU is used to solve Dying ReLU problem. The mathematical representation of Leaky ReLU is:

where m is a constant, called leak factor and generally it set to a small value (like 0.001).

4.4 Noisy ReLU:

Noisy ReLU is used Gaussian distribution to make ReLU noisy. The mathematical representation of Noisy ReLU is:

$$f(x)\text{Noisy ReLU} = \max(x + Y), \text{ with } Y \sim N(0, \sigma(x))$$

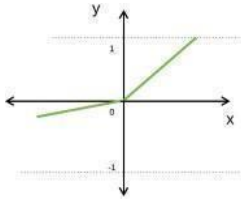


Figure 2.13: Leaky ReLU:

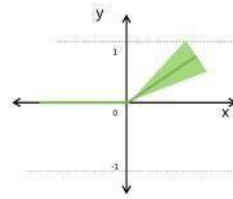


Figure 2.14: Noisy ReLU

4.5 Parametric Linear Units:

It is almost similar to Leaky ReLU, but here the leak factor is tuned during the model training process.

5. Facial expression recognition :

The facial recognition system using convolutional neural network (CNN) is a system that can categorized the type of facial expression recognition to seven category angry, sad, disgust, neutral, happy, fear, surprised, by using a specific algorithms and methods in deep learning and convolutional neural network

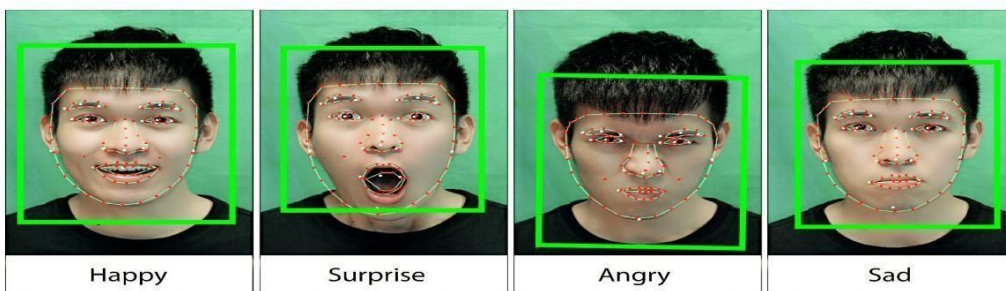


Figure 2.15: facial expression

This system enables machines to understand and categorize human emotions expressed through

because the image have a low resolution , or the faces are not in same position , some images have a text written on them , some people hide part of their faces with their hands

5.1 Features extracting images

We chose to use a Convolutional Neural Network in order to tackle this face recognition problem. Indeed this type of Neural Network (CNN) and is widely used for image analysis subjects like image classification.

6. Proposed face expression Recognition System

The recognition system relies on two fundamental stages: Training and Testing. In the training phase, input-data, feature extraction, and classification . During testing, the system performs data-set extracts features, and delivers final conclusions.

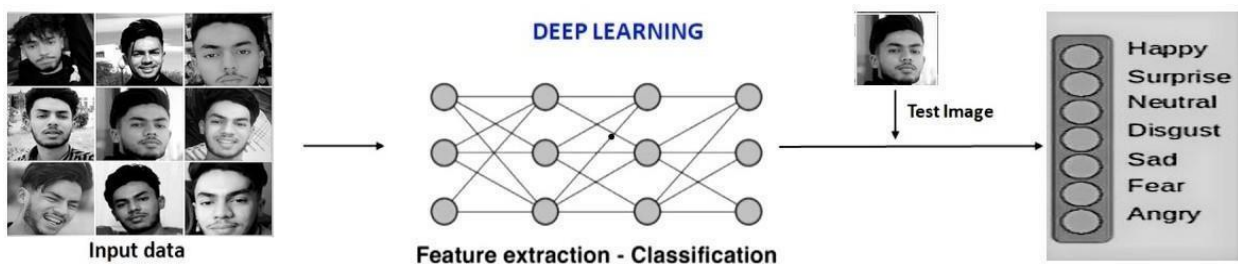


Figure2.16 : proposed system

6.1 Pre-processing:

Pre-processing plays a crucial role in expression recognition systems as it directly influences the matching accuracy. The objective of image pre-processing is to enhance the quality of iris images and eliminate any noise or artifacts that could potentially impact the accuracy of the recognition system. Techniques like segmentation, normalization, and enhancement are utilized as part of the pre-processing stage to enhance the accuracy of iris recognition systems.

6.1.1 Feature extraction-classification :

Considerable research efforts have been dedicated to exploring Convolutional Neural Networks (CNNs) for finger vein identification. However, a challenge arises when utilizing CNNs to determine the optimal parameters that result in the highest recognition rate percentage.

We chose to use a Convolutional Neural Network in order to tackle this face recognition problem. Indeed this type of Neural Network (NN) is good for extracting the features of images and is widely used for image analysis subjects like image classification.

We define our CNN with the following global architecture:

- 4 convolutional layers
- 2 fully connected layers

The convolutional layers will extract relevant features from the images and the fully connected layers will focus on using these features to classify well our images

our convolution layers work. Each of them contain the following operations:

A convolution operator: extracts features from the input image using sliding matrices to preserve the spatial relations between the pixels. The following image summarizes how it works:

The green matrix corresponds to the raw image values. The orange sliding matrix is called a 'filter' or 'kernel'. This filter slides over the image by one pixel at each step (stride). During each step, we multiply the filter with the corresponding elements of the base matrix. There are different types of filters and each one will be able to retrieve different image features:


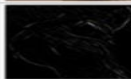
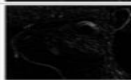
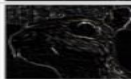



Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Figure 2.17 : operations

We apply the ReLU function to introduce non linearity in our CNN. Other functions like tanh or sigmoid could also be used, but ReLU has been found to perform better in most situations. Pooling is used to reduce the dimensionality of each feature while retaining the most important information. Like for the convolutional step, we apply a sliding function on our data. Different functions can be applied: max, sum, mean... The max function usually performs better.

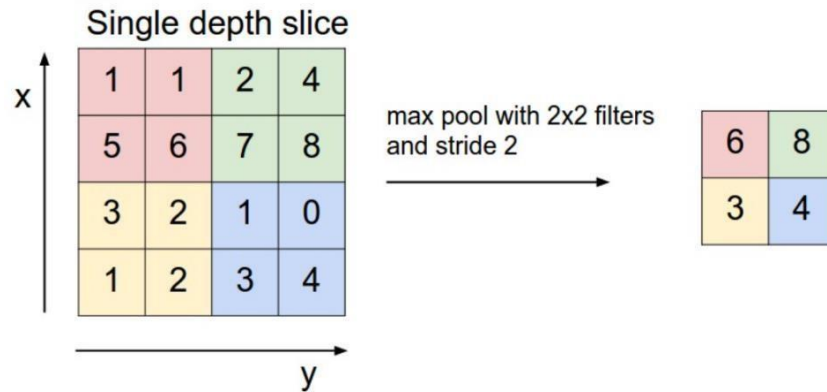


Figure 2.18: max pooling operation

We also use some common techniques for each layer:

Batch normalization: improves the performance and stability of NNs by providing inputs with zero mean and unit variance.

Dropout: reduces overfitting by randomly not updating the weights of some nodes. This helps prevent the NN from relying on one node in the layer too much.

We chose softmax as our last activation function as it is commonly used for multi-label classification.

Now that our CNN is defined, we can compile it with a few more parameters. We chose the Adam optimizer as it is one of the most computationally effective. We chose the categorical cross-entropy as our loss function as it is quite relevant for classification tasks. Our metric will be the accuracy, which is also quite informative for classification tasks on balanced datasets.

-II. Conclusion:

the second chapter of our presentation delved into the process of face expression recognition using Convolutional Neural Networks (CNN). We explored the various stages involved in this process, from data acquisition and preprocessing to the design and training of the CNN model. By leveraging the power of deep learning, we aimed to extract meaningful features from facial images and enable accurate classification of different facial expressions.

This chapter laid the foundation for our subsequent discussions on the results and findings of our study.

Chapter 03

Results and Discussions

"Face recognition: Unmasking the future."

I. Introduction :

This chapter delves into the comprehensive depiction of the database employed in our research. Furthermore, we elucidate the evaluation metrics employed, while presenting our outcomes through a combination of quantitative and qualitative discussions. These discussions aim to underscore the merits and drawbacks of our system, emphasizing its strengths and weaknesses. Additionally, we undertake a comparative study to delve deeper into the performance analysis of our system.

1. Data-set description :

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. Each image corresponds to a facial expression in one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The dataset contains approximately 36K images.

The original data consisted in arrays with a greyscale value for each pixel. We converted this data into raw images and splitted them in multiple folders

80% of our images are contained inside the train folder, and the last 20% are inside the validation folder.



Figure 3.1:dataset images content

1.2 Evaluation metrics

Performance assessment in early research often relied on the term "Accuracy" to evaluate models. Accuracy is typically calculated by comparing the model's predictions to the ground truth. In the case of Object Detection, the ground truth consists of bounding boxes and object classes in an image. A high level of accuracy indicates that the model accurately predicts both the bounding boxes and the object classes.

Furthermore, we illustrate the performance of each model across various ranks using Cumulative Match Characteristics (CMC) curves.

For our performance assessment, we utilize the confusion matrix to compute metrics such as Accuracy, Precision, and Recall.

The confusion matrix is a very usual metric used to solve classification problems , it can be utilized for binary classification and multi classes classifications problems

	Classified as Positive	Classified as Negative
Really is Positive	True Positive (TP)	False Negative (FN)
Really is Negative	False Positive (FP)	True Negative (TN)

TABLE 3.1: Confusion Matrix.

Accuracy

Accuracy is the amount of correct classifications over all of the classifications made

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (3.1)$$

PRECISION

Precision is used to show how accurate the predicted result is.

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{TP}} \quad (3.2)$$

Recall

Recall is used to indicate how well a positive is given as an input.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3.3)$$

2.Results and Discussion:

To get successful results, We used many techniques to test our model. We fixed the learning rate and change the type of the optimizer each time

we changed the number of epochs we used 10 and 50 epochs each time ,16, 32, 64 and 128 are the values of the batch size that we employed in our research. The 3 optimizer that we use are: Adam,SGD,RMSprop The results we will provide in the tables

2.1 Adam Results :

models	Hyper-parameters			Evaluation					
	optimizer	#of epochs	Batch size	Loss	Acc	Spec	Sens	Pre	F1-score
1st Model	Adam	10	16	1.1223	0.5752	0.48	0.47	0.26	0.75
			32	1.1431	0.5680	0.44	0.44	0.3	0.76
			64	1.1997	0.5366	0.31	0.37	0.26	0.77
			128	1.2829	0.5103	0.28	0.31	0.17	0.74
		50	16	1.2236	0.6484	0.56	0.71	0.49	0.84
			32	1.1717	0.6453	0.55	0.6	0.49	0.84
			64	1.0737	0.6525	0.56	0.7	0.52	0.85
			128	1.0687	0.6416	0.55	0.68	0.43	0.82

Table3.1 the results of Adam optimizer

we set the learning rate and the optimizer in advance and start changing the number of epochs and batchsize and at 50 epochs and batch size 64 we achieve the best estimated value accuracy of 65,25 % and we use the four necessary parameters for evaluating our model

Optimizer : Adam

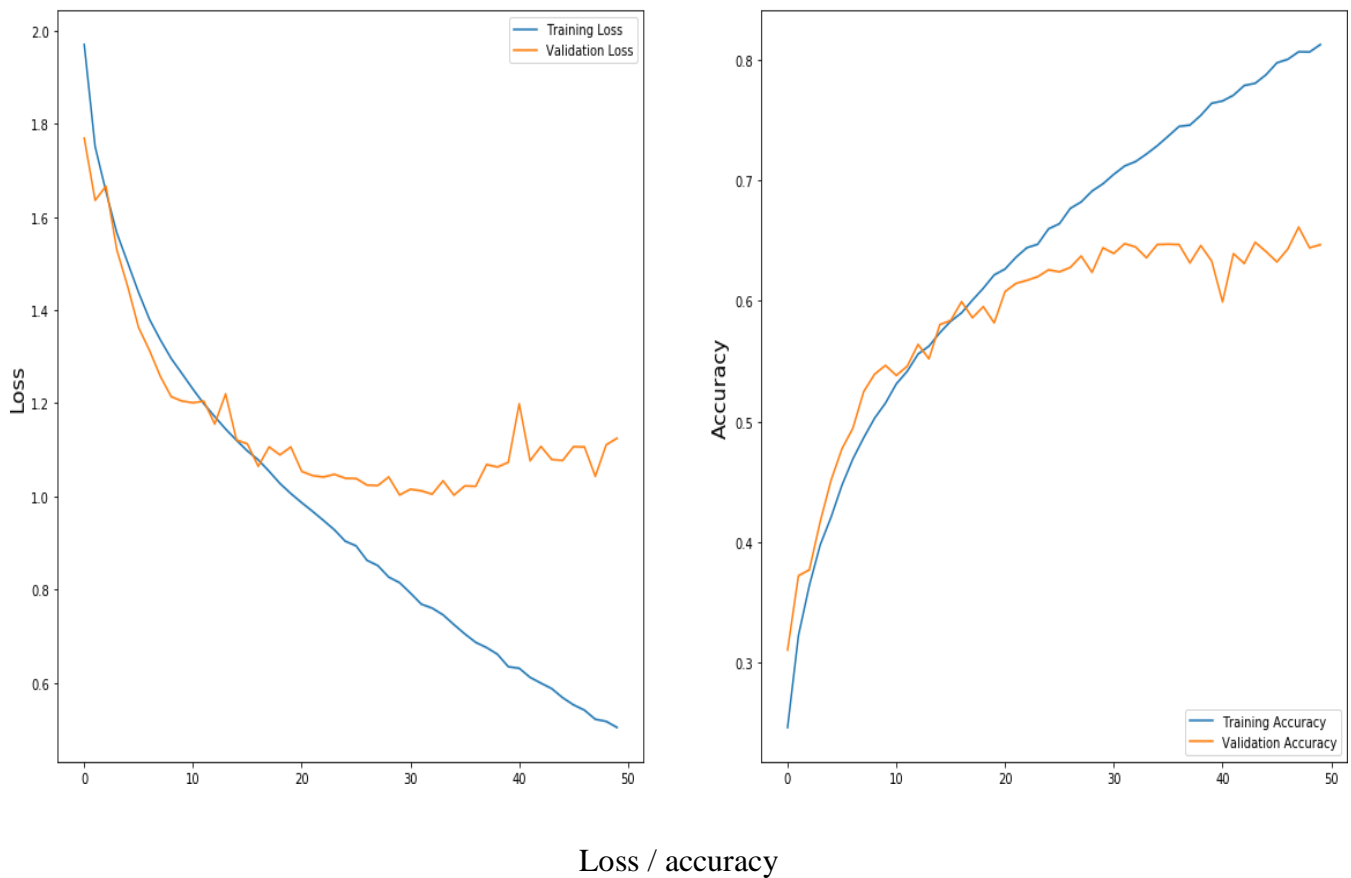


Figure 3.2 loss / acc with accuracy 64,54 %

In figure 3.2

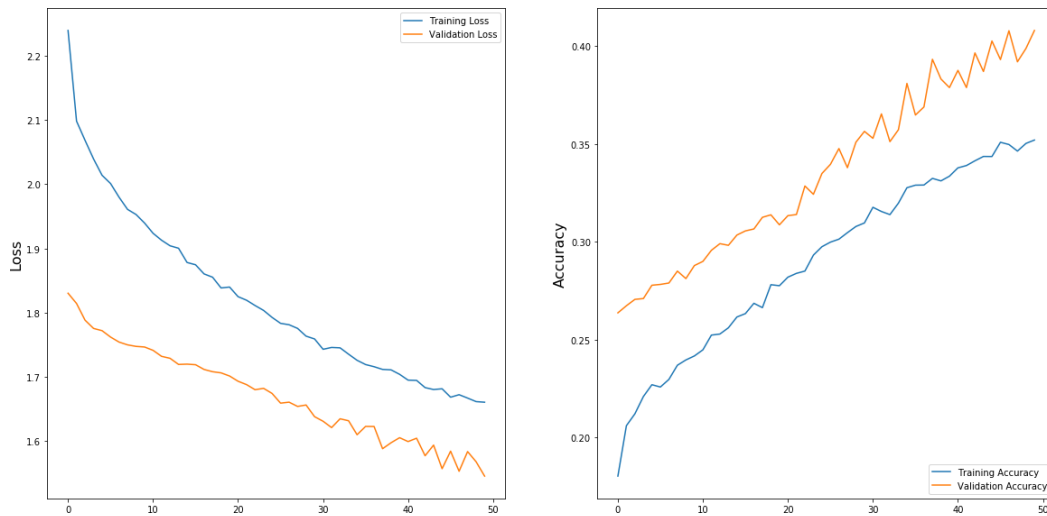
curve(acc) the training and validation accuracies are similar and both show improvement over time, it indicates that the model learning effectively and generalizing well to unseen data and both training and validation show a steady increase and eventually plateau, it mean that the model has reached its maximum potential with the given training data.in the other hand curve (loss) show us the training loss consistently decreases over time, it indicates that the model is effectively learning from the training data and minimizing the error and the validation loss decreases and follows a similar trend as the training loss, it that mean the model is generalizing well to the training data and preform well for unseen data

2.2 SGD Results :

models	Hyper-parameters			Evaluation					
	optimizer	#of epochs	Batch size	Loss	Acc	Spec	Sens	Pre	F1-score
2nd Model	SGD	10	16	1.790	0.2662	0.01	0.0	0.01	0.45
			32	1.777	0.2807	0.0	0.0	0.01	0.44
			64	1.8178	0.2667	0.04	0.0	0.01	0.42
			128	1.8127	0.2640	0.08	0.0	0.07	0.42
		50	16	1.5964	0.3853	0.02	0.0	0.08	0.57
			32	1.6875	0.335	0.0	0.0	0.06	0.5
			64	1.7185	0.3171	0.05	0.0	0.1	0.5
			128	1.7876	0.2725	0.0	0.0	0.03	0.43

Table3.2 the results of SGD optimizer

we set the learning rate and the optimizer in advance and start changing the number of epochs and batch size and at 50 epochs and batch size 64 we achieve the best estimated value accuracy of 38,53 % as you can see the sensibility (recall) is 0 in all the tests and precision also very low we found out that SGD optimiser not working well in face expression recognition



Loss / accuracy

Figure 3.3 loss / acc with accuracy 38,53 %

In figure 3.2 , we can see in **curve (acc)** when the validation accuracy consistently surpasses the training accuracy without crossing, that mean the model struggle with over fitting and have difficulties in general nation and indicate instability in the training process. in the other hand **curve(loss)** the training loss continues to decrease while the validation loss remains high, it mean that the model may not generalize well to new data that what lead to low Accuracy

2.3 RMSprop RESULTS:

models	Hyper-parameters			Evaluation					
	optimizer	#of epochs	Batch size	Loss	Acc	Spec	Sens	Pre	F1-score
3th Model	RMSprop	10	16	1.1160	0.5806	0.49	0.44	0.26	0.79
			32	1.1022	0.5863	0.48	0.46	0.36	0.8
			64	1.1996	0.5400	0.43	0.37	0.32	0.78
			128	1.2563	0.5185	0.36	0.27	0.21	0.75
		50	16	1.0783	0.6171	0.51	0.61	0.46	0.83
			32	1.0495	0.6511	0.56	0.68	0.5	0.84
			64	1.1383	0.6508	0.54	0.67	0.51	0.83
			128	1.0664	0.6563	0.55	0.65	0.5	0.82

Table3.3 the results of RMSprop optimizer

we set the learning rate and the optimizer in advance and start changing the number of epochs and batch size and at 50 epochs and batch size 128 we achieve the best estimated value accuracy of 65,14 % this test have the best Accuracy In the study

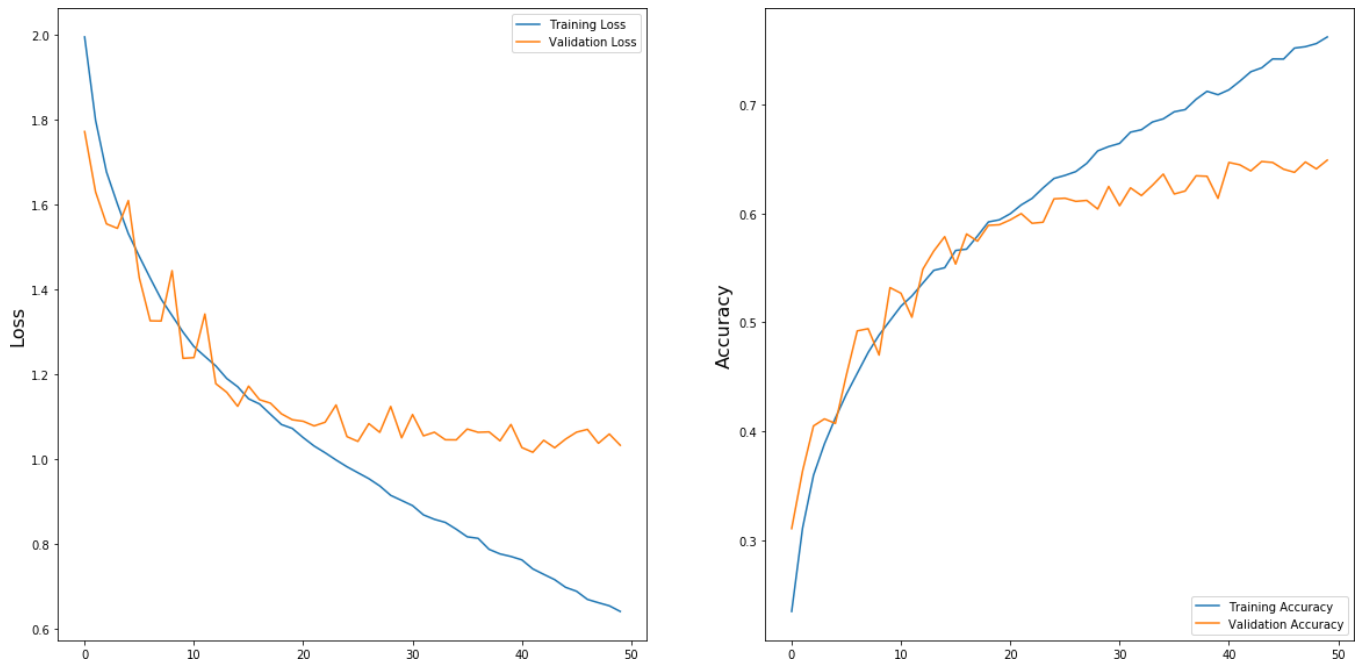


Figure 3.4 loss / acc with accuracy 65,14 %

In figure 3.3 , we can see In terms of the accuracy curve, both the training and validation accuracies exhibit a comparable pattern, demonstrating consistent improvement over time. This suggests that the model is effectively learning and capable of generalizing to unfamiliar data. Moreover, the training and validation accuracies exhibit a steady rise, followed by reaching a plateau, indicating that the model has maximized its potential with the available training data.

On the other hand, when examining the loss curve, it is evident that the training loss consistently diminishes as the training progresses. This signifies that the model is effectively learning from the training data and minimizing errors. Additionally, the validation loss follows a similar trend as the training loss, demonstrating that the model generalizes well to the training data and performs satisfactorily on unseen data.

3. Confusion matrix :

The confusion matrix provides an overview of the model's performance in classifying different facial expressions. This confusion matrix we use the model who has the best accuracy RMSprop optimiser 65.63%

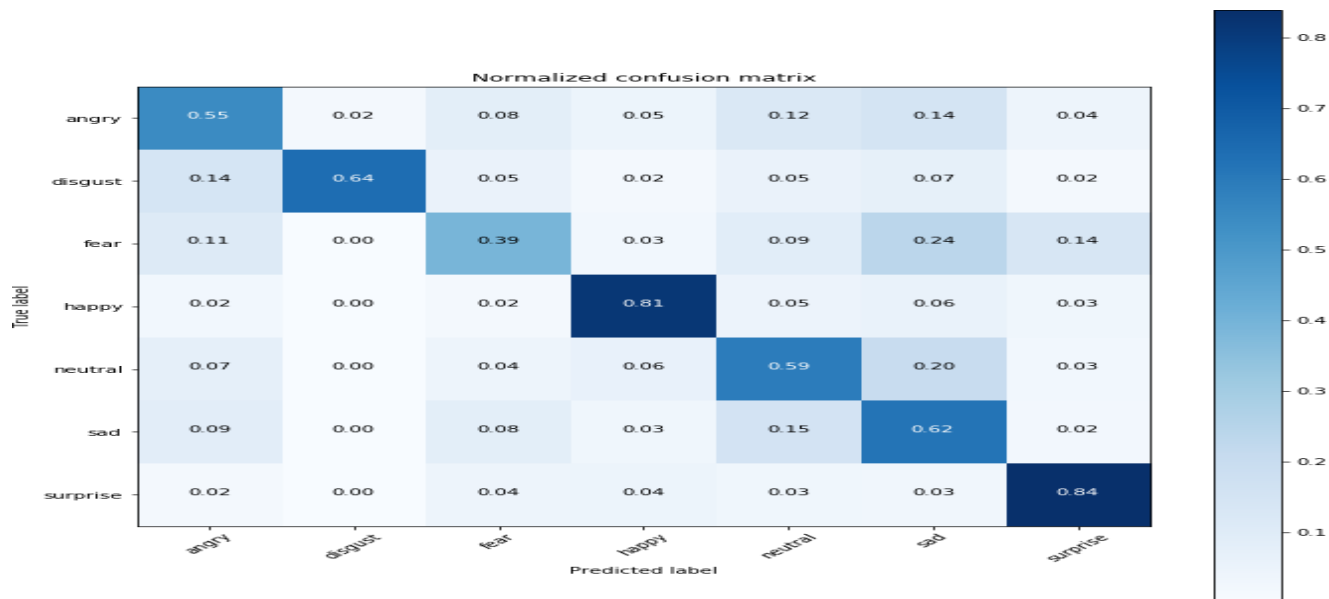


Figure 3.5 Confusion matrix result

And this the classification results :

	precision	recall	f1-score	support
Angry	0.58	0.55	0.56	960
disgust	0.72	0.64	0.68	111
fear	0.57	0.39	0.46	1018
happy	0.87	0.81	0.84	1825
neutral	0.59	0.59	0.59	1216
sad	0.48	0.62	0.54	1139
surprise	0.69	0.84	0.76	797

Table 3.4 classification rapport

The classification report provides more detailed metrics for each class, including precision, recall (sensitivity), specificity, and F1-score. These metrics help assess the model's performance on individual facial expressions.

II. Conclusion :

In this chapter, we have discussed the database employed and the criteria for evaluating our work. Throughout this chapter, we have engaged in practical exercises and evaluated the outcomes using four distinct models.

GENERAL CONCLUSION:

face recognition using Convolutional Neural Networks (CNNs) has shown significant progress in accurately detecting and classifying facial expressions. CNNs are a popular deep learning architecture known for their effectiveness in image processing tasks, including facial expression recognition.

Facial expression recognition aims to identify and classify the emotional state or facial expression of a person based on their facial features. CNNs excel in this area due to their ability to automatically learn and extract relevant features from facial images.

By training CNNs on large datasets of labeled facial expressions, such as the Facial Expression Recognition Challenge (FERC) or the Extended Cohn-Kanade (CK+) database, researchers have achieved impressive results in accurately recognizing emotions such as happiness, sadness, anger, surprise, fear, and disgust

Referances :

- [1] Tarun Kumar: A Theory Based on Conversion of RGB image to Gray image, International Journal of Computer Applications (0975 – 8887) Volume 7– No.2, September 2010.
- [2] Burger, W., and Burge, M. J. Principles of digital image processing: fundamental techniques, vol. 1. Springer Science & Business Media, 2010.
- [3] Berra ilyes zine abidine Cryptanalysis and Improvement on an Image Encryption Algorithm Design Using a Pseudo Random generator Democratic Republic of Algeria Mohamed Boudiaf – M'sila University Faculty of Mathematics and Computer Science-Computer Science department
- [4] Ashu Kumar, Amandeep Kaur, Munish Kumar Face detection techniques: a review Springer Nature B.V. 2018
- [5] deep learning : A comprehensive overview on techniques , taxonomy applications and research direction springer link [6] MACHINE LEARNING, An Algorithmic Perspective, Ralf Herbrich, 2015, International Standard Book Number-13: 978
- [7] Raktim Nath ,Kaberi Kakoty Ditya, jyoti Bora Face Detection and Recognition Using Machine January 2021
- [8] Ignas Kukenys Support Vector Machines for Human Face Detection article January 2008
- [9] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In COLT '92
- [10] Applying Haar Cascade Classifier Conference Paper · January
- [11] Ammad Ali Syed Farhan Khan Face Recognition with Local Binary Patterns Conference Paper · December 2012
- [12] Adrian Rosebrock OpenCV Face detection with Haar cascades On April 5, 2021
- [13] Varshachoudhary WEB article what SQL? 10 May, 2023
- [14] Ahmad F. Al Musawi Introduction to Machine

- [15]Anirudha Ghosh A. Sufian Farhana SultanaAmlan Chakrabarti
Fundamental Concepts of Convolutional Neural Network Chapter ·
January2020
- [16] Saad Albawi Tareq Abed Mohammed Understanding of a
ConvolutionalNeural Network Conference Paper · August 2017
- [17]M.A. Aizerman, E.M. Braverman, and L.I. Rozonoer. Theoretical
foundationsofthe potential function method in pattern recognition learning.
Automation
- [18]Anila S, Devarajan N. Preprocessing technique for face recognition
applicationsunder varying illumination conditions. Global Journal of
Computer Science and Technology. 2012 Jul 31.
- [19]Timea Bezdan Nebojsa BacaninConvolutional Neural Network
Layersand Architectures Conference Paper · January 2019
- [20] Dadi HS, Pillutla GM. Improved face recognition rate using
HOGfeatures and SVM classifier. IOSR Journal of Electronics and
Communication Engineering. 2016 Apr;11(04):34-44.
- [21]Deepa, S. N., Sivanandam, S. N., Sumathi, S., 2006, Introduction
toNeural Networks using MATLAB 6.0, vol. 1, Tata McGraw-Hill. [22]J.
Wu, “Introduction to Convolutional Neural Networks,” pp. 1–28,2016.
- [23]Bellot P, De Los Campos G, Pérez-Enciso M (2018) Can deep learning
improve genomic predictionof complex human traits? Genetics210:809–819
- [24]M. Turk and A. Pentland (Jun. 1991). “Face Recognition Using
Eigenfaces.” Proceedings of CVPR IEEE Computer Society
- [25] Mitchell, T. M. The Discipline of Machine Learning, Machine Learning
Department technical report CMU-ML-06-108.
- [26]V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep
learning,” pp. 1–28, 2016

