General Introduction

The security and safety of individuals, properties and information need to be guaranteed , actually one of the major concerns of our societies, especially after the great spread of terrorism around the world. In fact, people willing to cross boundaries must prove their identities using their passports, people willing to cross buildings or academic institution must validate their access cards, people desiring access to banking services must login using a login and a password. Nevertheless, these traditional methods show great weaknesses for identity verification. Indeed, the identity of a person is directly related to that they possess (such as passport, access card, etc.) or/and that they know (password, PIN codes, etc.). Nonetheless, PIN codes and passwords may be forgotten or compromised and access cards may be falsified or duplicated which lead to identity sponge. In this respect, experts are looking for a technology which resolves these problems by giving more convenience to persons and ensuring a highly secured access, by relating the identity of a person to what they are and not to what they possess or know.

Biometry is the most suitable technology for identity verification and/or person identification by employing their physiological features including biological, morphological and behavioral characteristics. This technology makes identity data theft more difficult and thus, increases user confidence as the physical presence is necessary during identification.[1]

In our work, we have chosen facial recognition as an average of identification compared to other methods because this identification is naturally used by a human being, so, this type of recognition does not stop with the identification of the face, but can apply to the location of an individual in a crowd, unlike other methods, and does not require very complex acquisition equipment that is to say a simple camera can acquire the shape of an individual’s face and then remove certain features. The essential features for face recognition are eyes, mouth, face, nose, etc. Depending on the system used, the individual must be positioned in front of the camera where they may be moving at a distance. The biometric data that is obtained is compared to the reference file. The software must be able to identify an individual despite various physical devices (mustache, beard, glasses, etc.).[2]

This thesis deals with a topic of identification. An identification system is intended to answer the question; ”who is this person?” You have to check the biometrics against others in the database. Which is simplified to **1: Many** It is, therefore, responsible for discovering the identity of an unknown person in a dataset. Several methods have been developed in the literature for face recognition. In our work, we have opted for a technique based on neural networks called the Convolutional Neural Networks (CNN) which is a type of neural network with deep learning, or Deep Neural Network. The latter has several hidden layers. CNN consists of two very distinct parts, part of extraction that can be used to simplify an input image, reducing its size, and part of classification that classifies this data.[2]

We chose to articulate our study around four main chapters. The first chapter is devoted to the general presentation of biometrics. It describes the operating principle of biometric systems and then defines the tools used to evaluate their performance. Then, the place of facial recognition among the other biometric techniques is analyzed. Through this chapter, we want to position the problem of facial recognition and present its issues and interests to other techniques. Finally, we highlight the difficulties faced by face recognition systems. In the second chapter, we will discuss the state of the art of face recognition techniques. We present just the most popular face recognition and face detection algorithms, and quote some of the most used databases for face recognition. The third chapter is composed of two parts. We will first take on the artificial neural network basics (ANN), which is the heart of the recognition system. Then we talk about recognition techniques based on deep neural network (Deep Learning) of the convolutional neural network-type (CNN) in the second part. In the fourth chapter, we present the experimental results obtained by methods of face recognition that we choose and analyze their performance, followed by a discussion with the interpretation of the results.

Finally, the general conclusion will summarize the results obtained by our approach.

Part 1

State of the art

Chapter I

Biometry and facial recognition systems

I.1 Introduction

Biometry is a growing technology which has become increasingly used in our daily life. It aims to establish the identity of a person as reliable as possible using their biological features in order to guarantee the safety of people in public places. In this chapter, we introduce firstly, the identity of a biometric system, structure and the different biometric modalities.

Eventually, we will showcase one of the most efficient modalities to identify a subject; which is the face. And the whole process from taking a picture of a person to identifying the person in it.

I.2 Biometry

I.2.1 Definition of biometry

Biometry is the verification of individual identity based on his biological characteristics which are classified into two categories. The first one is physical characteristics which are most commonly used and rely on physical traits of individuals such as iris, fingerprint, palmprint, face, etc., and the second kind is behavioral characteristics which are less used and rely on individual actions or behaviors such as walking, voice, dynamic signature, etc. These physical and behavioral characteristics that allow persons identification are called biometric modalities [1].

Biometry is the science to understand how to measure these person-specific Characteristics and how to use them to distinguish individuals. Researchers in biometrics try to automatize such processes and make them suitable To be run on a computer or a device by a biometric system [3].

I.2.2 Properties of a biometric modality

Principal properties of a biometric modality are the following:

- Universality : The whole population should possess this modality (physical or behavioral characteristic).

- Distinctiveness: Two different individuals must have different biometric representations.

- Stability : To ensure individual authentication success, biometric modality should be relatively stable over time and it also has to be stable regardless conditions of acquisition (external conditions, emotional conditions of the person, etc.).

- Collectability : The biometric modality must be acquired.

- Acceptance : The acceptance and the facility of usage are related to the acquisition constraints of a biometric modality.

- Circumvention : The biometric modality must not be easily falsified.

- Performance : Biometric recognition should be accurate, fast, and robust with regards to operational and environmental changes.

All modalities do not possess all these properties, or may possess them with different degrees. Hence, there is no ideal or perfect modality. The trade-off between presence and absence of some of these properties is required according to each system needs, regarding the choice of biometric modality [1].

I.2.3 Biometric modalities

There are many different biometric modalities that are used to acquire information about personal traits of humans, and they are classified into three main categories (biological, behavioral and morphological). The modalities that are used the most today are fingerprint, face, iris, and voice. These happen to be the biometric modalities that, today, best meet the tests for uniqueness, permanence, and consistency let alone the ease of capturing them using sensing devices. This section discusses some examples of different biometric modalities that are based on either biological, behavioral or morphological analysis.

- **Biological :** This category is based on the analysis of the biological characteristics of the individual. The premise to this type of analysis is that the biological data of each individual is a personal signature. Biological analysis includes odor, DNA, and physiological signals [4]. However in biometrics for automated user authentication, DNA analysis is not yet used mainly due to two reasons. First, extraction of the DNA sequences still requires biochemical processing, which cannot be fully automated today and is quite time consuming. The second reason is the fact that organic material carrying DNA may be lost easily. Consequently, it may be collected and re-used by other subjects easily, for example by collecting a sample of a lost hair from a brush or leavings of saliva from a glass [5].

- **Behavioral :** This category is based on the analysis of an individual's behavior, such as signature dynamics, demarche, typing, and voice [4]. It is mainly characterized by three categories of individual traits: the biological construction of the organs producing behavior, the learned characteristics of how to produce behavior and the purpose or intention, which action exactly to be produced. For example in speech based biometrics, various aspects of the biological construction of mouth, vocal cords and glottis influence the individual sound characteristics of speech generation. On the other side learned characteristics include linguistic aspects like vocal tones, pronunciation and speech tempo, which are heavily influenced by the way the speaking capability has been acquired [5].

- **Morphological :** This category is based on the use of physical traits that are unique and permanent in the individual. Several modalities have been used to extract this information such as the face, the fingerprint, the geometry of the hand, the iris, etc [4]. Physiological traits of persons represent biological structures, which are individual and which may be acquired without taking physical samples, e.g. by optical means. These can be seen as visible or at least measurable physical results, naturally grown as programmed by the genetic construction code. For example, the structure of the ridges on fingertips has proven to be individual and persistent for most human beings [5].

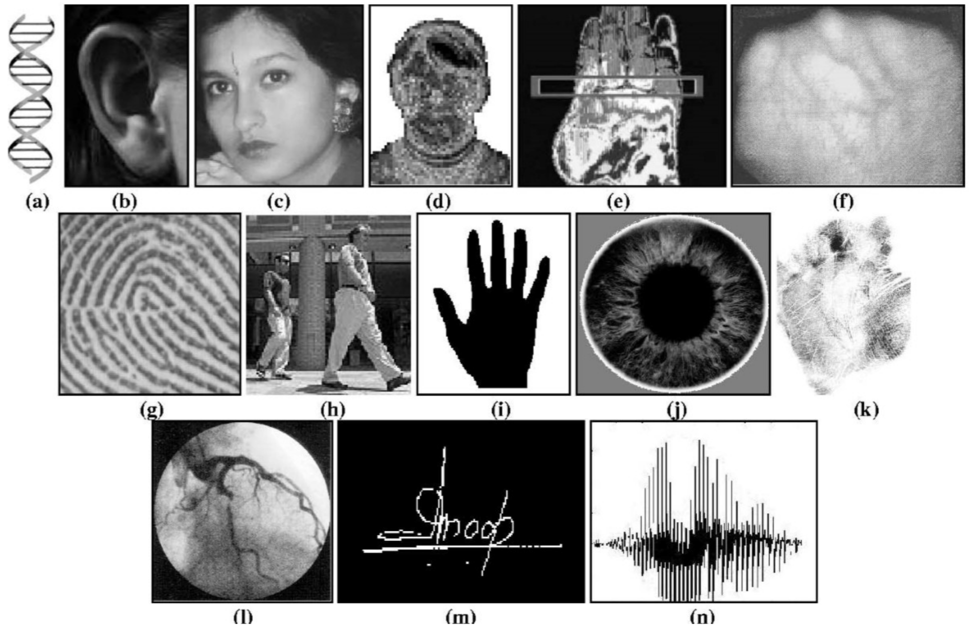

Figure I.1: Examples of biometric characteristics
(a) DNA, (b) ear, (c) face, (d) facial thermogram, (e) hand thermogram, (f) hand vein, (g) fingerprint, (h) gait, (i) hand geometry, (j) iris, (k) palmprint, (l) retina, (m) signature, and (n) voice. [6]

I.2.4 Biometric systems

A biometric system is essentially a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the template set in the database (see Fig. I.2). Depending on the application context, a biometric system may operate either in verification mode or identification mode.

- In the **verification mode**, the system validates a person's identity by comparing the captured biometric data with her own biometric template(s) stored in the system database. In such a system, an individual who desires to be recognized claims an identity, usually via a personal identification number (PIN), a user name, or a smart card, and the system conducts a one-to-one comparison to determine whether the claim is true or not (e.g., "Does this biometric data belong to Bob?"). Identity verification is typically used for positive recognition, where the aim is to prevent multiple people from using the same identity.

The verification problem may be formally posed as follows: given an input feature vector X_Q (extracted from the biometric data) and a claimed identity I, determine if (I, X_Q) belongs to class w_1 or w_2, where w_1 indicates that the claim is true (a genuine user) and w_2 indicates that the claim is false (an impostor). Typically, X_Q is matched against X_I, the biometric template corresponding to user I, to determine its category. Thus

$$(I, X_Q) \in \begin{cases} w_1, & \text{if } S(X_Q, X_1) \geq t \\ w_2, & \text{otherwise} \end{cases} \tag{I.1}$$

where S is the function that measures the similarity between feature vectors X_Q and X_I, and t is a predefined threshold. The value $S(X_Q, X_1)$ is termed as a similarity or matching score between the biometric measurements of the user and the claimed identity. Therefore, every claimed identity is classified into w_1 or w_2 based on the variables X_Q, I, X_I, and t and the function S. Note that biometric measurements (e.g., fingerprints) of the same individual taken at different times are almost never identical. This is the reason for introducing the threshold t.

- In the **identification mode**, the system recognizes an individual by searching the templates of all the users in the database for a match. Therefore, the system conducts a one-to-many comparison to establish an individual's identity (or fails if the subject is not enrolled in the system database) without the subject having to claim an identity (e.g., "Whose biometric data is this?"). Identification is a critical component in negative recognition applications where the system establishes whether the person is who she (implicitly or explicitly) denies to be. The purpose of negative recognition is to prevent a single person from using multiple identities. Identification may also be used in positive recognition for convenience (the user is not required to claim an identity). While traditional methods of personal recognition such as passwords, PINs, keys, and tokens may work for positive recognition, negative recognition can only be established through biometrics .

The identification problem, on the other hand, may be stated as follows. Given an input feature vector X_Q, determine the identity I_K, $K \in \{1, 2....N, N+1\}$. Here $I_1, I_2, ..., I_N$ are the identities enrolled in the system and I_{N+1} indicates the reject case where no suitable identity can be determined for the user. Hence

$$X_Q \in \begin{cases} I_K, & \text{if } max_K \{S(X_Q, X_{I_K})\} \geq t, K = 1, 2, ..., N \\ I_{N+1}, & \text{otherwise} \end{cases} \qquad (I.2)$$

where X_{I_K} is the biometric template corresponding to identity I_K, and t is a predefined threshold [6].

Before we move on to the structure of the biometric system, we have to know that in order to identify/verify a subject we should have a database of templates of individuals, which is filled in the enrollement phase :

- **Enrollment** is common for both verification and identification modes. It is the preliminary phase where the biometric data of a user is registered for the first time in the system. During this phase, one or more biometric modalities are captured and stored as templates in the database. This phase is very crucial since it influences, later, the whole recognition process. In fact, the quality of enrolled data is essential for ulterior identification phases because acquired data are considered as references for the person. A set of samples should be captured to take into account the variability of biometric modality of a person [1].

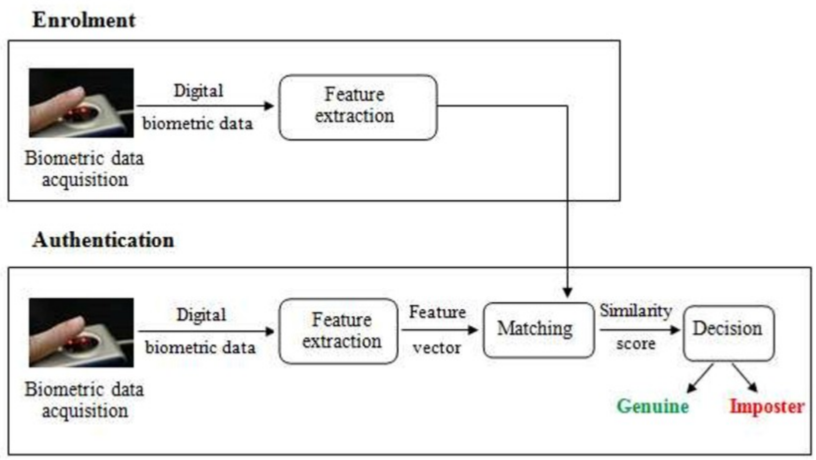

Figure I.2: Block diagrams of enrollment, verification, and identification [1]

I.2.5 Structure of a biometric system

The structure of a biometric system is composed of four modules. A biometric system is designed using the following four main modules (see Fig. I.2).

- **Sensor module,** which captures the biometric data of an individual. An example is a fingerprint sensor that images the ridge and valley structure of a user’s finger.

- **Feature extraction module,** in which the acquired biometric data is processed to extract a set of salient or discriminatory features. For example, the position and orientation of minutiae points (local ridge and valley singularities) in a fingerprint image are extracted in the feature extraction module of a fingerprint-based biometric system.

- **Matcher module,** in which the features extracted during recognition are compared against the stored templates to generate matching scores. For example, in the matching module of a fingerprint-based biometric system, the number of matching minutiae between the input and the template fingerprint images is determined and a matching score is reported. The matcher module also encapsulates a decision making module, in which a user's claimed identity is confirmed (verification) or a user's identity is established (identification) based on the matching score.

- **System database module,** which is used by the biometric system to store the biometric templates of the enrolled users. The enrollment module is responsible for enrolling individuals into the biometric system database. During the enrollment phase, the biometric characteristic of an individual is first scanned by a biometric reader to produce a digital representation of the characteristic. The data capture during the enrollment process may or may not be supervised by a human depending on the application. A quality check is generally performed to ensure that the acquired sample can be reliably processed by successive stages. In order to facilitate matching, the input digital representation is further processed by a feature extractor to generate a compact but expressive representation, called a template. Depending on the application, the template may be stored in the central database of the biometric system or be recorded on a smart card issued to the individual. Usually, multiple templates of an individual are stored to account for variations observed in the biometric trait and the templates in the database may be updated over time [6].

I.2.6 Performance of biometric systems

To evaluate the performance of a biometric system, there are two types of errors to check for :

- **False Acceptance Rate (FAR) :** which is when the system erroneously recognizes two different samples as samples from the same source

$$FAR = \frac{\text{number of accepted imposters (False Accept)}}{\text{total number of imposters' accesses}} \tag{I.3}$$

- **False Rejection Rate (FRR) :** which is when the system erroneously recognizes two samples from the same source as samples from different sources.

$$FRR = \frac{\text{number of rejected clients (False Reject)}}{\text{total number of client accesses}} \tag{I.4}$$

After calculating the FAR and FRR, we can calculate the **Equal Error Rate (EER)**, This rate is calculated from the first two criteria and constitutes a point of measurement of current performance. This point corresponds to the place where FRR = FAR, that is to say the best compromise between the false rejections and the false acceptances [7].

$$EER = \frac{\text{number of false acceptance} + \text{number of false rejection}}{\text{total number of accesses}} \quad (I.5)$$

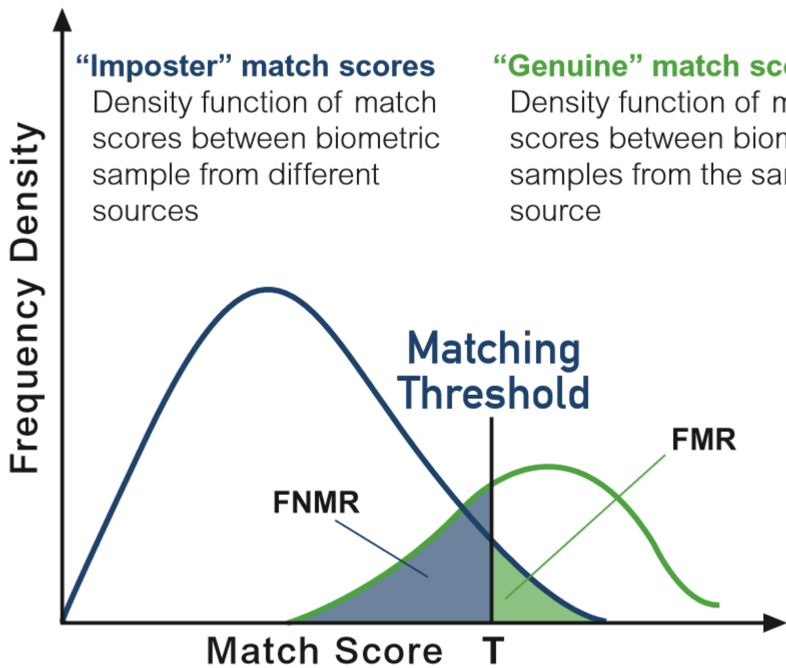

Figure I.3: FAR and FRR diagram
[8]

The system performance at all the operating points (thresholds) can be depicted in the form of a **Receiver Operating Characteristic(ROC)** curve.A ROC curve is a plot of FMR against or FNMR for various threshold values [6].The more this curve fits the mark shape the more the system is efficient with a high Recognition Rate (RR) [1].

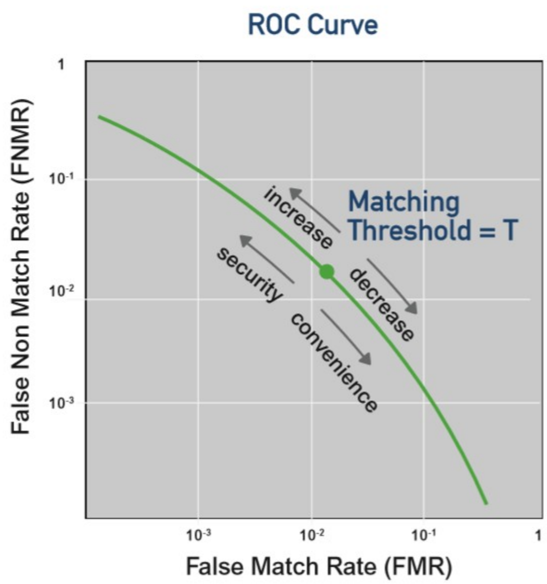

Figure I.4: A ROC curve for a given biometric matching system [8]

I.3 Facial recognition

I.3.1 Why facial recognition ?

So many biometric modalities are used to identify subjects (see figure I.1),and facial recognition is one of the most used biometrics in the world today because of its efficiency. The reason after using the face biometric is not only its efficiency but also :

- The ease of use : facial recognition does not require any process from the user, it's enough to just hold still or walk in front of a camera.

- availability of equipment the equipment used for the acquisition of images and its simplicity and its low price.

I.3.2 Facial recognition system

A facial recognition system must have the ability to identify faces in an image or video automatically. The basic operating principle of a facial recognition system can be summarized in the following steps :

I.3.2.1 Image acquisition

This is the first step in identifying subjects, the sensor used for acquiring face images is digital cameras. We must succeed in capturing information relevant without noise. The image in this step is in a raw state which generates a risk of noise that can degrade the performance of the system [9].

I.3.2.2 Detection

Face detection can be done by detecting the color of the skin, the shape of the head or by methods detecting the different characteristics of the face.This step is dependent on the quality of the images acquired. The overall performance of any automatic system recognition largely depend on the performance of face detection. In the detection step, we identify and locate the face in the image acquired at the beginning, regardless of position, scale, orientation and lighting [9].

I.3.2.3 Preprocessing

Preprocessing consists in eliminating the parasites caused by the quality of the sensors used during the acquisition of the image to keep the essential information alone [2].and also dealing with lighting conditions and the posture of the subject...etc

I.3.2.4 Feature extraction

Mainly two categories of feature extraction can be found in face recognition today: global and component based approaches. In the first category, typically all or part of the original image is used as one single feature vector, which requires alignment between the images in all cases. Such an alignment can be performed for example by detection of corresponding key points in the facial part of the photograph and a subsequent warping of one of the images towards the other(s). The other category of features addresses geometrical properties of the face, such as relation and size of eyes, nose and mouth in the image. Another approach is to identify additional key points on the face and expand an elastic graph model between them [5].

I.3.2.5 Classification

It consists of modeling the parameters extracted from a face or a set of faces of an individual based on their common characteristics. A model is a set of useful, discriminant and non-redundant information that characterizes one or several individuals with similarities, they will be grouped in the same class, and these classes vary depending on the type of decision [9].

I.3.2.6 Learning

After extraction and classification, a learning step consists of memorizing the parameters in a well-ordered database to facilitate the recognition and decision-making phase [2].

I.3.2.7 Decision

This is the step that makes the difference between a system of identification and a verification system. In this step, an identification system is to find the model that best fits the face taken from those stored in the database, it is characterized by its recognition rate. On the other hand, in a verification system it is a question of deciding whether the face in entry is indeed that of the individual (model) proclaimed or he is an impostor. To estimate the difference between two images, it is necessary to introduce a measure of similarity [9].

I.3.3 Difficulties of facial recognition

For the human brain, the process of face recognition is a high-level visual task. Although human beings can detect and identify faces in a scene without much trouble, building an automatic system that performs such tasks is a serious challenge. This challenge is hard as the conditions for acquiring images are very variable. There are two types of variations associated with face images inter and intra-subject. Inter-subject variation is limited because of the physical resemblance between individuals. On the other hand, the intra-subject variation is larger. It can be attributed to several factors that we analyze here.

- **Change of illumination :** The change of illumination of a face is a critical task and this leads to make the facial recognition task very difficult and also lead to misclassification.

- **Pose variations :** The pose variation is another problem for facial recognition systems, if there are pose variations in the images, it affects facial recognition rate.

- **Facialexpressions :** The appearance of a face varies greatly in the presence of facialexpressions, the facial elements such as the mouth or the eyes can suffer significant deformations that can cause a failure of a facial recognition system , it necessarily causes a decrease in the recognition rate.

- **Structural components :** The presence of structural components (beard, mustache, or glasses) can significantly alter the facial features, these components can hide the basic facial features causing a failure of the recognition system.

- **Partial occlusions :** Partial occlusions can be caused by a hand hiding a part of the face, by long hair, glasses, the sun, by any other object (scarf ...), or by another person [2].

I.4 Conclusion

In this chapter, we have chiefly described the general context of biometry by describing the different biometric modalities and their properties. We outlined the structure of a biometric system and how to calculate the performace of such a system. And then we focused on one of the modalities to identify subjects which is the face, and we showed both why face recognition is one of the most used modalities today and how a facial recognition system is structered.

The following chapter will introduce the steps and methods and the needed tools of making a face recognition operation.

Chapter II

Face detection and recognition methods

II.1 Introduction

Face Recognition is a central topic in Face Analysis research. A biometric system may be used for verification or identification. The system in identification must find the identity of the individual presented to the system and the system in verification receives an identity and must make the decision whether or not the image corresponds to the identity. In both cases, the problem comes back, however, to a problem of classification. Many face recognition techniques have been proposed over the past 30 years. In this chapter, we briefly describe some of the most important or popular techniques used in face recognition.

II.2 Face recognition techniques

The ultimate goal of facial recognition is to compete, or even exceed, human abilities of recognition. Several face identification methods have been proposed during the twenty last years. There are three categories of methods: **global methods** , **local methods** and **hybrid methods** .

II.2.1 Global methods

The principle of these approaches is to use the entire surface as a source of information without taking into account local characteristics such as the eyes, the mouth ... etc. Global algorithms are based on well known statistical properties and use linear algebra. They are relatively fast to implement but are sensitive to variations in illumination, pose and expression of face [9]. One of the approaches used here is the artificial neural networks (which we will be talking more in depth about in the next chapter).

II.2.2 Local methods

They are also called line, geometric, local characteristics, or analytic. This type involves applying transformations in specific areas of the image, most often around the characteristic points (corners of eyes, mouth, nose, ...), the focus will be given to small local details avoiding the noise caused by hair, glasses, hats, beard, etc. But their difficulty is present when it comes to taking into consideration several views of the face as well as the lack of precision in the ”extraction” phase of the points constitute their major disadvantage. Specifically, these methods extract local face characteristics such as eyes, nose and mouth, then use their geometry and / or appearance as given input of the classifier [9].

II.2.3 Hybrid methods

The robustness of a recognition system can be increased by merging several methods. It is also possible to use a combination of classifiers based on various techniques in order to unite the strengths of each and thus overcome their weaknesses. Hybrid techniques combine the two previous methods for better characterization of face images [9].

II.3 Face detection algorithms

II.3.1 Viola-Jones (HAAR CASCADE)

The core basis for Haar classifier object detection is the Haar-like features. These features, rather than using the intensity values of a pixel, use the change in contrast values between adjacent rectangular groups of pixels. The contrast variances between the pixel groups are used to determine relative light and dark areas. Two or three adjacent groups with a relative contrast variance form a Haar-like feature. Haar-like features, as shown in figure II.1 are used to detect an image. Haar features can easily be scaled by increasing or decreasing the size of the pixel group being examined. This allows features to be used to detect objects of various sizes.[10]

Due to the nature of the algorithm, the Viola-Jones method is restricted to binary classification tasks (such as object detection) and has a very long training period. However, it classifies images quickly because each weak classifier requires only a small number of parameters, and with a sufficient number of weak classifiers, it has a low rate of false positives.[11]

Rectangle features can be computed very rapidly using an intermediate representation for the image which we call the integral image. The integral image at location x; y contains the sum of the pixels above and to the left of x; y, inclusive:

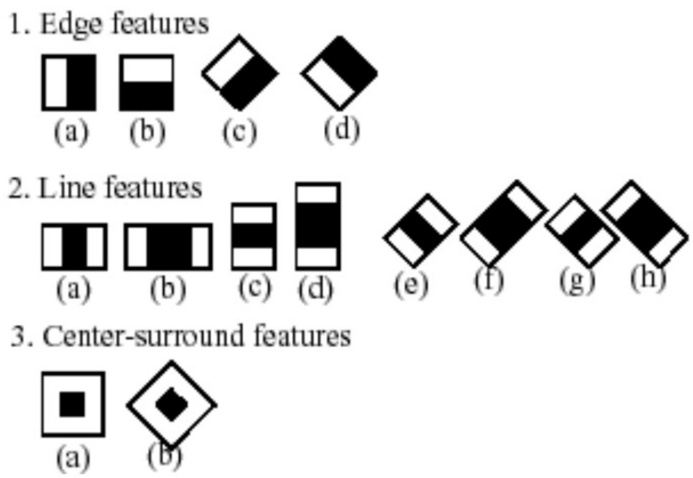

Figure II.1: Common HAAR features [10]

$$ii(x; y) = \sum_{x' \leq x, y' \leq y} i(x', y') \qquad \text{(II.1)}$$

where $ii(x; y)$ is the integral image and $i(x; y)$ is the original image. Using the following pair of recurrences:

$$s(x; y) = s(x, y - 1) + i(x, y) \qquad \text{(II.2)}$$

$$ii(x; y) = ii(x - 1, y) + s(x, y) \qquad \text{(II.3)}$$

(where $s(x; y)$ is the cumulative row sum, $s(x; 1) = 0$, and $ii(1; y) = 0$)

The integral Image can be computed in one pass over the original image. Using the integral image any rectangular sum can be computed in four array references.[12]

II.3.2 Histogram of Oriented Gradients "HOG"

For the histogram of oriented gradients, or HOG, algorithm, to detect faces in a photograph, the first step is to convert the input image to black-and-white. The HOG algorithm does not need color information: it only looks at changes between light and dark areas in an image. basicaly, it devides the image into small cells and compares the pixels in that area to each other and try to measure the variation of darkness, and then find the direction where the biggest change happens. This shows the movement of lighting at this exact point. If we repeat this process for every single pixel in the image, the image turns into a map of transitions from light to dark areas. These lines are called gradients. Each gradient shows how the image flows from a light area to a dark area at that point. But, that's still not enough because the image is still pretty complex and detailed. to detect the face we only need the overall structures, so the image will be further simplified by going over it again with a bigger block this time and we'll count up how many gradients point in each major direction. Instead of keeping track of all separate gradients within this block, we'll just store a count of how many gradients point in each direction. and the direction that has the most counts is the strongest factor that represents that area of the image. There are also gradients pointing in other directions that we'll keep track of. We'll represent those other directions here as lines that are less bold. Now, this can be repeated for the entire image. The original image is now a simple representation that captures the basic structure. We can use this simplified representation to easily train a face detection model.

Figure II.2: Analyzing an image as a histogram of oriented gradients

After converting the images to HOG representations, we will start training a machine learning face detection model by giving it lots of examples of HOG representations of faces so it can learn what this pattern looks like. HOG simplifies the image in a way that still retains the key information needed to spot faces. By simplifying the problem this way, it makes it easier for the machine learning model to solve it. But HOG has some other nice advantages, as well, that make it work better for small training sets. First, the HOG representation of an image doesn't change even when you lighten or darken the image. Since HOG only looks for changes in brightness and not absolute brightness, making an image a little brighter or a little darker doesn't change the HOG representation at all. Second, the HOG representation of an image doesn't change even if you change the shapes in the image a little bit. Because it is only looking at broad changes in the intents the over large areas of the image, small changes in shape don't matter. This is great for face detection because it means that two faces that don't look exactly the same will still have nearly the same HO representation.[13]

II.4 Face databases

Many databases containing information that enables the evaluation of face recognition systems are available on the market. However, these databases are generally adapted to the needs of some specific recognition algorithms, each of which has been constructed with various image acquisition conditions (changes in illumination, pose, facial expressions) as well as the number of sessions for each individual. These databases range in size, scope and purpose.

II.4.1 Labeled faces in the wild (LFW)

The primary contribution of LFW is providing a large set of relatively unconstrained face images. By unconstrained, we mean faces that show a large range of the variation seen in everyday life. This includes variation in pose, lighting, expression, background, race, ethnicity, age, gender, clothing, hairstyles, camera quality, color saturation, and other parameters. The reason we are interested in natural variation is that for many tasks, face recognition must operate in real-world situations where we have little to no control over the composition, or the images are pre-existing. For example, there is a wealth of unconstrained face images on the Internet, and developing recognition algorithms capable of handling such data would be extremely beneficial for information retrieval and data mining. Since LFW closely approximates the distribution of such images, algorithms trained on LFW could be directly applied to web IR applications.[14]
this database contains 13233 images of 5749 persons, and was all collected directly from Yahoo's website.

II.4.2 FERET Database

The FERET database was collected as part of the Facial Recognition Technology program conducted by the US National Institute of Standards and Technology (NIST). This is the largest base available for researchers that were acquired with different poses and during 15 sessions between 1993 and 1996. The images, initially collected from a 35mm camera, were then digitized. The first version of this database was produced in 2001 and contains 14051 grayscale facial images with a resolution of 256 x 384 pixels. The latest version, made in 2003, contains higher quality color digital images with a resolution of 512 x 768 pixels and lossless compression of data, unlike the first grayscale images. In addition, multiple image name, identify and capture date errors, which appear on the first grayscale base, have been corrected. This last database contains 11338 images representing 994 different people.[15]

II.4.3 The AR Database

The AR base was established in 1998 at the Computer Vision Center (CVC) laboratory in Barcelona, Spain. 116 people (63 men and 53 women) are registered. The images are in color of size 768 x 576 pixels. 13 views on each topic were collected. For the majority of these people, 13 other views were acquired during a second session two weeks apart. Image views contain changes in facial expression, lighting, and partial occlusions of the eyes (sunglasses) and the lower part of the face (neck cover). In the second session, the 13 views are collected under the same conditions as for the first one.[16]

II.4.4 ORL Database

Designed by AT n T Laboratories at the University of Cambridge in England, the ORL database (Olivetti Research Laboratory) is a reference database for automatic face recognition systems. In fact, all face recognition systems found in the literature have been tested in relation to the ENT, this popularity is due to the number of constraints imposed by this base because most of the possible and foreseeable changes in the face have been taken into account.count, such as change of heart, beard, glasses, changes in facial expressions, etc. As well as the acquisition conditions such as the change of illumination and the change of scale due to the distance between the acquisition device and the individual. The ORL database consists of 40 individuals, each individual has 10 poses, so the database contains 400 images. The poses were taken over different time intervals of up to three months. The extraction of faces from the images was done manually.[15]

Figure II.3: face database samples of the databases mentioned above [17][18][19]

For this project, we are not going to use any of these databases. we will collect our own database using a google chrome extension to get the images of certain people from google image search, after collecting a good amount of images for each person in our database, we will put the images into a program to detect the faces in those images using the histogram of oriented gradients, and replace the old ones with the new cropped ones to make the features extraction in the CNN better.

II.5 Conclusion

In this chapter, we covered the characteristics of the techniques and methods of detecting and recognizing faces. Now, we will head to our main subject which is neural networks, and more precisely convolutional neural network and it’s role in facial recognition.

Chapter III

Neural networks

III.1 Introduction

Inventors have long dreamed of creating machines that think. This desire dates back to at least the time of ancient Greek. When programmable computers were first conceived, people wondered whether such machines might become intelligent. Today, artificial intelligence is a thriving field with many applications and active research topics. We look to intelligent software to automate routine labor, understand speech or image, make diagnoses in medicine and support basic scientific research. The true challenge to artificial intelligence proved to be solving the tasks that are easy for people to perform, but hard to describe formally problems that we solve intuitively, that feel automatic, like recognizing spoken words or faces in images [20]. The term Machine Learning (ML) refers to the automatic detection of significant patterns in the data. Over the past two decades, it has become a common tool in almost every task that requires extracting information from large data sets [21]. Deep learning is a subset of Machine learning, it is a way to extract useful patterns from data in an automated way which is done by the optimization of artificial neural network.

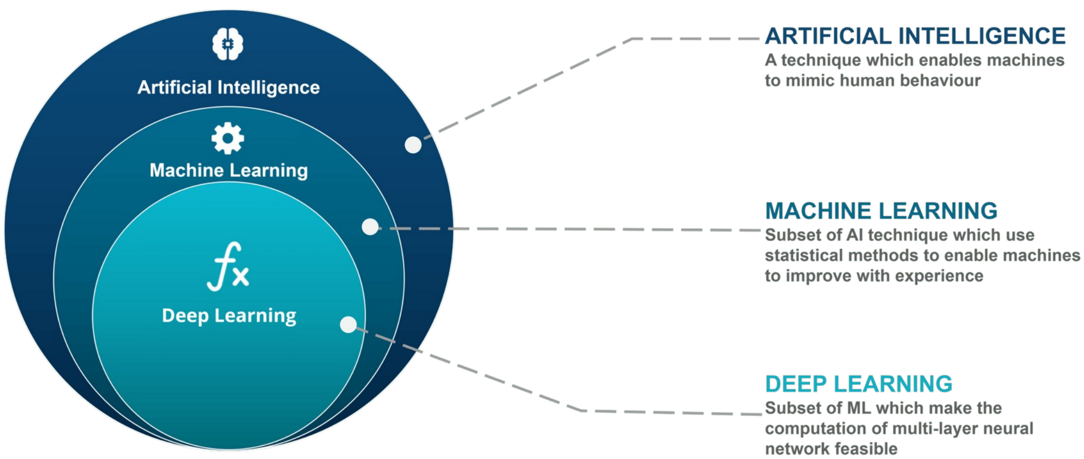

Figure III.1: Relation between AI, ML and DL
[22]

III.2 Artificial neural network

III.2.1 Definition of ANNs

Artificial Neural Networks (ANNs) are computational processing systems of which are heavily inspired by way biological nervous systems (such as the human brain) operate. ANNs are mainly comprised of a high number of interconnected computational nodes (referred to as neurons), of which work entwine in a distributed fashion to collectively learn from the input in order to optimise its final output. The basic structure of an ANN can be modelled as shown in Figure III.2 .We would load the input, usually in the form of a multidimensional vector to the input layer of which will distribute it to the hidden layers.The hidden layers will then make decisions from the previous layer and weigh up how a stochastic change within itself detriments or improves the final output, and this is referred to as the process of learning .Having multiple hidden layers stacked upon each-other is commonly called deep learning.[23]

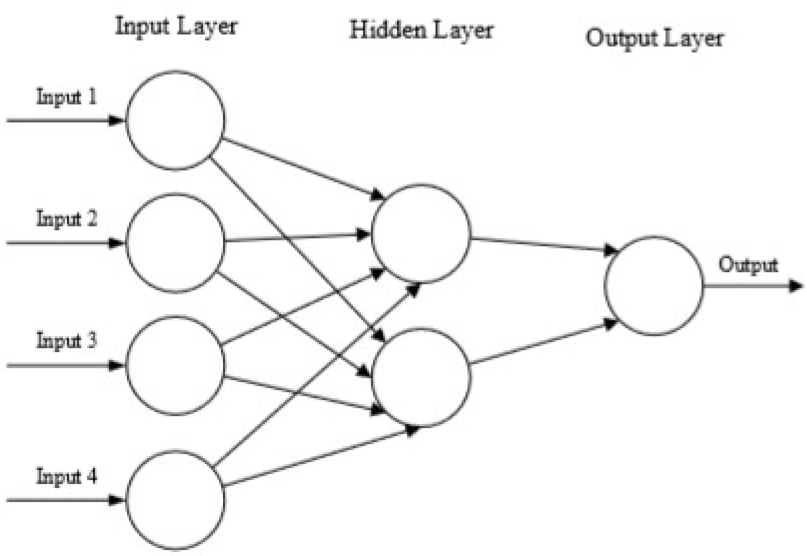

Figure III.2: Basic structure of an ANN
[2]

There is no universally accepted definition of neural network. It is generally considered that a neural network consists of a large set of units (or neurons), each having a small local memory. These units are connected by communication channels (connections, also called synapses in the corresponding biological term), which carry digital data. Units can only act on their local data and the inputs they receive through their connections.[24]

III.2.2 History and inspiration behind ANNs

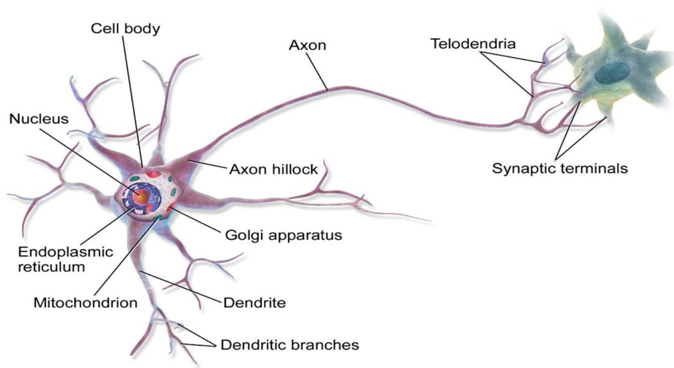

Figure III.3: Representaion of a biological neuron [25]

The physiology of the brain shows that it consists of interconnected cells (neurons). Neurons receive signals (electrical impulses) through highly branched extensions of their cell bodies (dendrites) and send the information through long extensions (axons). Electrical impulses are regenerated during the course along the axon. The duration of each pulse is of the order of 1 ms and its amplitude of about 100 mV. The contacts between two neurons, from the axon to a dendrite, are via the synapses Here is some information about the neurons of the human brain:

- the brain contains about 100 billion neurons.
- There are only a few dozen distinct categories of neurons. No category of neurons is unique to humans.
- The propagation capacity of the nervous impulses is in the range of 100m / s, which is much less than the speed of transmission of information in an electronic circuit.[26]

You can see the evolution of the neural networks through history in the table III.4 bellow:

1943	**McCulloch** and **Pitts** give a first interpretation of the formal neuron under a logic model.
1947	Publication of **Norbert Wiener**'s global reference book: *Cybernetics or Control and Communication in the animal and the machine*
1949	**Hebb**'s publication of a formal theory of biological learning through synaptic modification (neural connections).
1957	Creation of the first system copying the principle of the neuron called the perceptron, invented by **Rosenblatt**.
1962	**John Holland** proposes the current formalization of genetic algorithms.
1970	Creation of the Game of Life by **Conway**, first real artificial ecosystem.
Depuis 1985	Neural networks are becoming more and more commonly used in computer science.
1999	A team of German researchers managed to connect a neuron to a circuit and stimulate it to get answers.
2000	American researchers manufacture for the first time a biological processor, it is composed of four neurons of leeches and manages to make additions.

Figure III.4: Recap of historical dates of the evolution of neural networks [27]

III.2.3 Architecture of ANNs

Layered networks are the most commonly used connectionist models. The architecture, organized in successive layers, comprises an input layer and an output layer and one or more intermediate layers called hidden layers because they are not seen from the outside. Each layer is composed of a number of neurons. The connections are established between the neurons belonging to successive layers but the neurons of the same layer can not communicate with each other in the case of layered networks.

There are two types of ANNs: feedforward Networks and recurrent Networks:

- **Feedforward neural networks:** in a feedforward neural network, the information flowing from the inputs to the outputs without "going back"; if we represent the network graphically, the graph of a network is acyclic, if we move in the network, from any neuron, following the connections, we can not go back to the starting neuron The majority of feedforward neural networks are implemented for automatic classification tasks are organized in several layers, some of which are hidden.

- **Recurrent neural networks:** A network of looped or recurrent connection neurons means that one or more neuron outputs of a down-stream layer are connected to the inputs of the neurons of the upstream layer. These recurrent connections bring the information back to the meaning of defined in an feedforward network. Unlike feedforward neural networks, the connection graph of the recurrent neural networks is cyclic: when one moves in the network, following the direction of the connections, it is possible to find at least one way back to its point of departure (such path is referred to as ”Cycle”).[2]

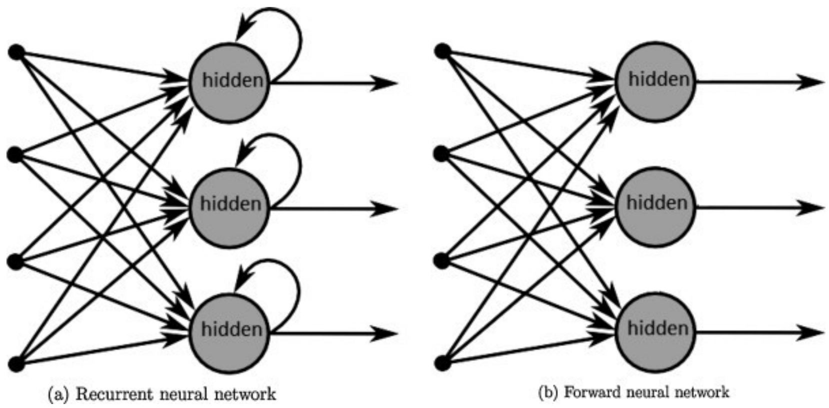

Figure III.5: recurrent network and feedforward network [24]

III.2.4 Learning paradigms

A characteristic of neural networks is their ability to learn (for example to recognize a letter, a sound ...). But this knowledge is not acquired from the beginning. Most neural networks learn by example by following a learning algorithm. There are two main algorithms: supervised learning and unsupervised learning[24]:

- **Supervised learning** is learning through pre-labelled inputs, which act as targets. For each training example there will be a set of input values (vectors) and one or more associated designated output values. The goal of this form of training is to reduce the models overall classification error, through correct calculation of the output value of training example by training.

- **Unsupervised learning** differs in that the training set does not include any labels. Success is usually determined by whether the network is able to reduce or increase an associated cost function. However, it is important to note that most image-focused pattern-recognition tasks usually depend on classification using supervised learning.[23]

III.2.5 Modeling of ANNs

The mathematical model of an artificial neuron, or ”perceptron”, is illustrated in the figure below. A neuron essentially consists of an integrator that performs the weighted sum of its inputs (as the statistical expectancy!). The result *n* of this sum is then transformed by a transfer function *f* which produces the output *a* of the neuron. The *R* inputs of the neuron correspond to the vector noted traditionally in line :

$$\vec{P} = \begin{pmatrix} P_1 \\ P_2 \\ \cdot \\ \cdot \\ \cdot \\ P_R \end{pmatrix}$$

while :

$$\vec{W} = \begin{pmatrix} W_{1,1} \\ W_{1,2} \\ \cdot \\ \cdot \\ \cdot \\ W_{1,R} \end{pmatrix}$$

represents the vector of neuron weights.[28]

Weights are how neural networks learn. We adjust the weights to determine the strength of the signal.

Weights help us come up with different outputs. we randomly initialize the weights *w* and multiply them with the inputs *p* and add the bias term *b*, so for the hidden layer, a compact version is to calculate *n* and then apply the activation function *f*.[29]

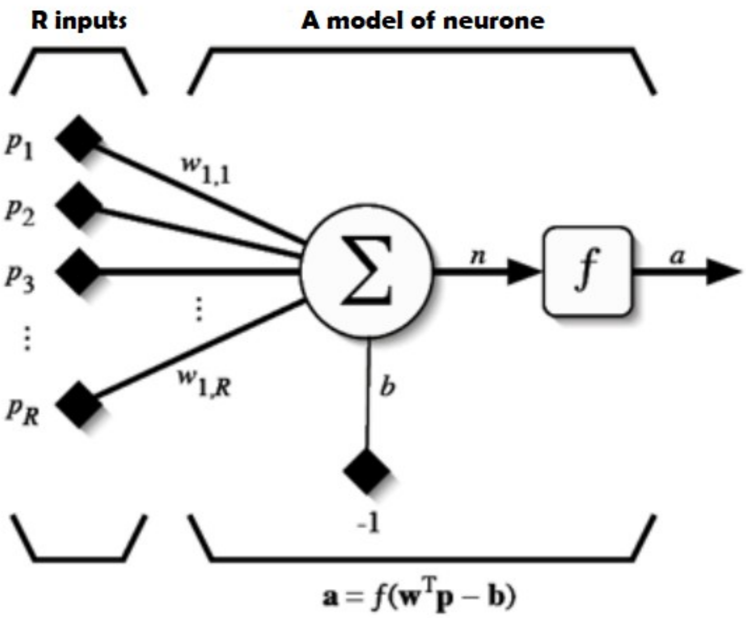

Figure III.6: representation of a mathematical neuron [28]

The output *n* of the integrator is defined (because it is a technique of the engineer) by the following equation:

$$n = \sum_{j=1}^{R} W_{1,j}P_j - b = W_{1,1}P_1 + W_{1,2}P_2 + \ldots + W_{1,R}P_R - b \qquad \text{(III.1)}$$

This output corresponds to a weighted sum of weights and inputs less than what we call "the bias of the neuron" (corrective factor decided by trial and error). The result n of the weighted sum is called the "activation level of the neuron". The bias *b* is also called the "activation threshold of the neuron". When the activation level reaches or exceeds the threshold *b*, then the argument of *f* becomes positive or obviously positive (or zero). Otherwise, it is negative.

As formulated by the preceding equation and adding the activation function *f* to obtain the output of the neuron: [28]

$$a = f(n) = f(wp - b) \qquad \text{(III.2)}$$

Activation function helps decide if we need to fire a neuron or not . If we need to fire a neuron then what will be the strength of the signal.

Activation function is the mechanism by which neurons process and pass the information through the neural network.

There are different types of activation functions and some very common and popular ones are:[29]

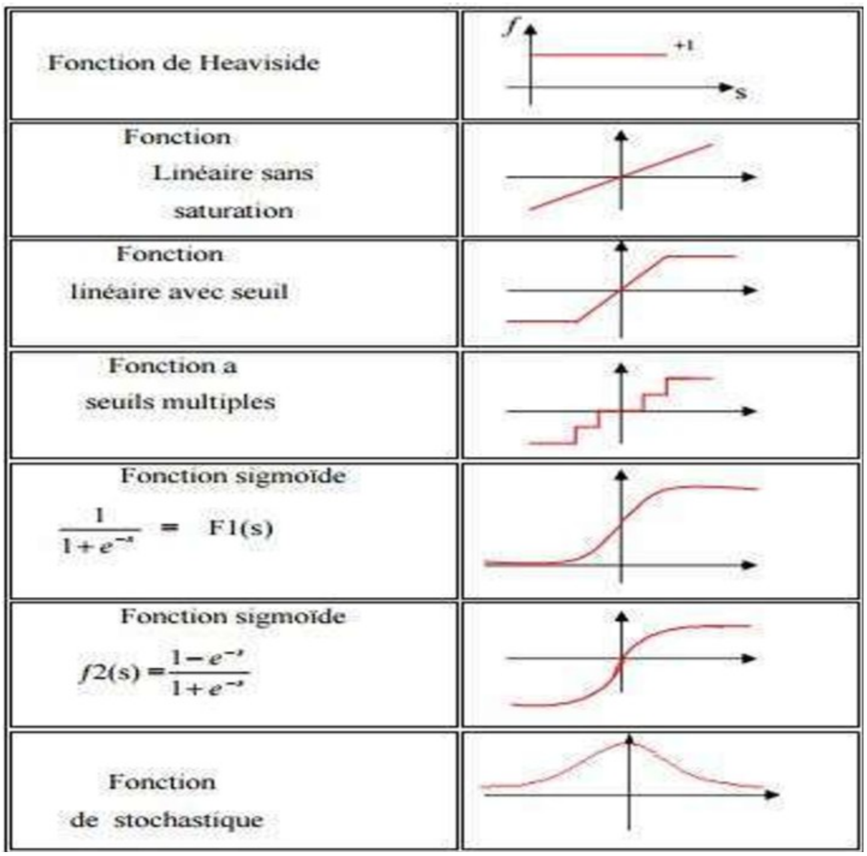

Figure III.7: A few activation functions

[2]

- **Back-Propagation :** After forward propagation we get an output value which is the predicted value. To calculate error we compare the predicted value with the actual output value. Then we calculate the derivative of the error value with respect to each and every weight in the neural network. Back-Propagation uses chain rule of Differential Calculus. In chain rule first we calculate the derivatives of error value with respect to the weight values of the last layer. We call these derivatives, gradients and use these gradient values to calculate the gradients of the second last layer. We repeat this process until we get gradients for each and every weight in our neural network. Then we subtract this gradient value from the weight value to reduce the error value. In this way we move closer (descent) to the Local Minima(means minimum loss).[30]

III.2.6 A few models of ANNs

As we have seen in the previous sections how the neural networks have evolved through history and that J.McCulloch and W.Pitts have established the first logical model of a neural network which gave D.Hebb the chance to elaborate a mathematical formula for it. All of this has led to the emergence of the first technical model wich is the perceptron By FRANK ROSENBLATT (we have talked about the perceptron in the previous section), and after that a lot of new models have emerged in this field, a few of them are :

- **MultiLayer Perceptron** is a perceptron enhancement that includes one or more hidden layers that make the MLP network a robust tool for complex tasks. It is widely used for the decision in the field of facial recognition. MLP networks are generally fully connected networks. The neurons of the first layer receive the input vector, they calculate their outputs which are transmitted to the neurons of the second layer which themselves calculate their outputs and so on from layer to layer to that of output. In the MLP network there is no connection between the cells of the same layer. Multilayer perceptrons are used with supervised learning and also with the backpropagation technique for error correction.[2]

- **Hopfield network** It is a network consisting of two state neurons (-1 and 1, or 0 and 1), whose learning law is the Hebb rule (1949), which states that a synapse improves its activity if and only if the activity of its two neurons is correlated (that is, the weight of a connection between two neurons increases when both neurons are activated at the same time).[24]

- **Convolutional neural network** One of the most impressive forms of ANN architecture is that of the Convolutional Neural Network (CNN). CNNs are primarily used to solve difficult image-driven pattern recognition tasks.[23] In the following sections we'll be talking more precisely about convolutional neural networks because they are the best solution out there for facial recognition tasks which is after all the title of this thesis.

III.3 Convolutional neural network

III.3.1 What is and why CNN ?

CNNs, like neural networks, are made up of neurons with learnable weights and biases. Each neuron receives several inputs, takes a weighted sum over them, pass it through an activation function and responds with an output. The whole network has a loss function and all the tips and tricks that we developed for neural networks still apply on CNNs.[31]

The only notable difference between CNNs and traditional ANNs is that CNNs are primarily used in the field of pattern recognition with in images.This allows us to encode image-specific features into the architecture, making the network more suited for image-focused tasks[23]

This choice was motivated mainly by implicitly incorporating a **feature extraction** phase and has been used successfully in many applications.[32]

One of the largest limitations of traditional forms of ANN is that they tend to struggle with the computational complexity required to compute image data.[23] That is the reason behind implementing CNNs, the properties that CNNs have such as feature extraction make them more efficient when handling images.

III.3.2 Layers in CNN

CNNs are comprised of three types of layers. These are convolutional layers, pooling layers and fully-connected layers. When these layers are stacked, a CNN architecture has been formed.[23]

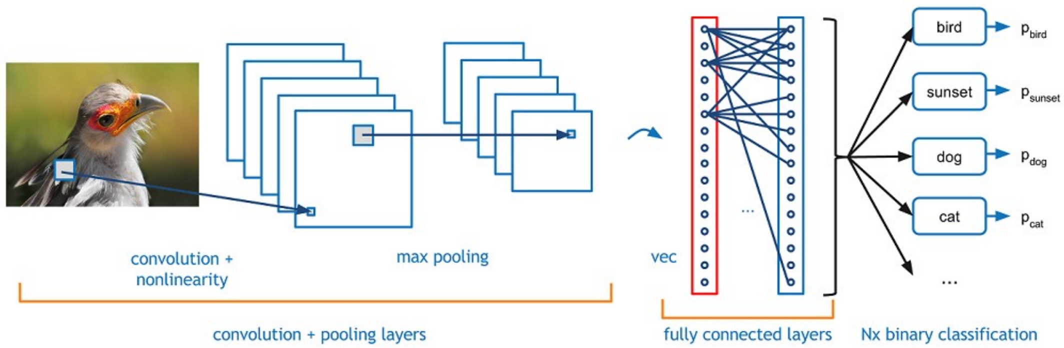

Figure III.8: A simple CNN architecture
[29]

III.3.2.1 Convolution layer

The convolutional layer plays a vital role in how CNNs operate. The layers parameters focus around the use of learnable kernels.

These kernels are usually small in spatial dimensionality, but spreads along the entirety of the depth of the input. When the data hits a convolutional layer, the layer convolves each filter across the spatial dimensionality of the input to produce a 2D activation map. As we glide through the input, the scalar product is calculated for each value in that kernel. (figure III.9) From this the network will learn kernels that 'fire' when they see a specific feature at a given spatial position of the input. These are commonly known as activations.

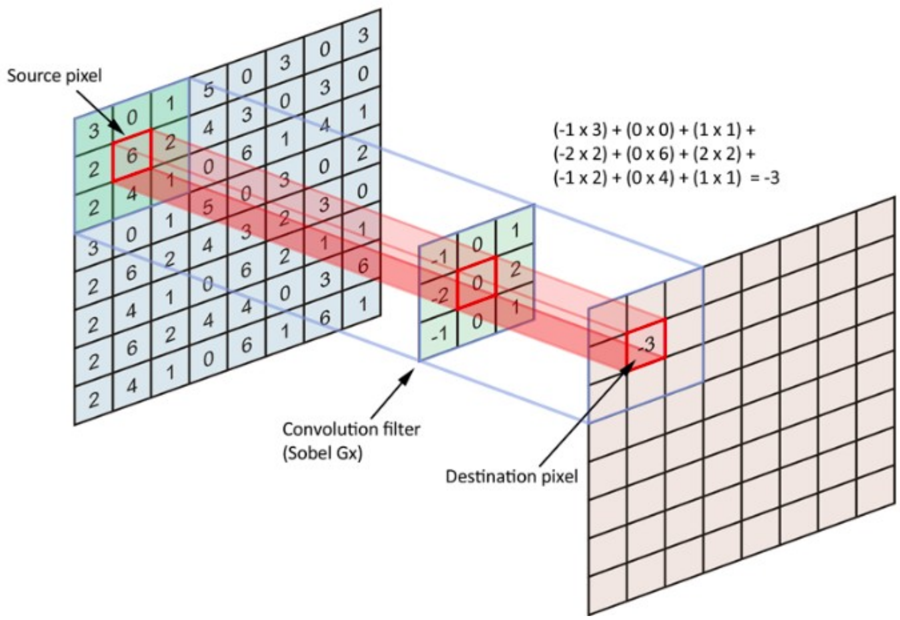

Figure III.9: The convolution operation
[33]

Every kernel will have a corresponding activation map, of which will be stacked along the depth dimension to form the full output volume from the convolutional layer.[23]

We perform numerous convolutions on our input, where each operation uses a different filter. This results in different feature maps. In the end, we take all of these feature maps and put them together as the final output of the convolution layer.[33]

Feature map and activation map mean exactly the same thing. It is called an activation map because it is a mapping that corresponds to the activation of different parts of the image, and also a feature map because it is also a mapping of where a certain kind of feature is found in the image.

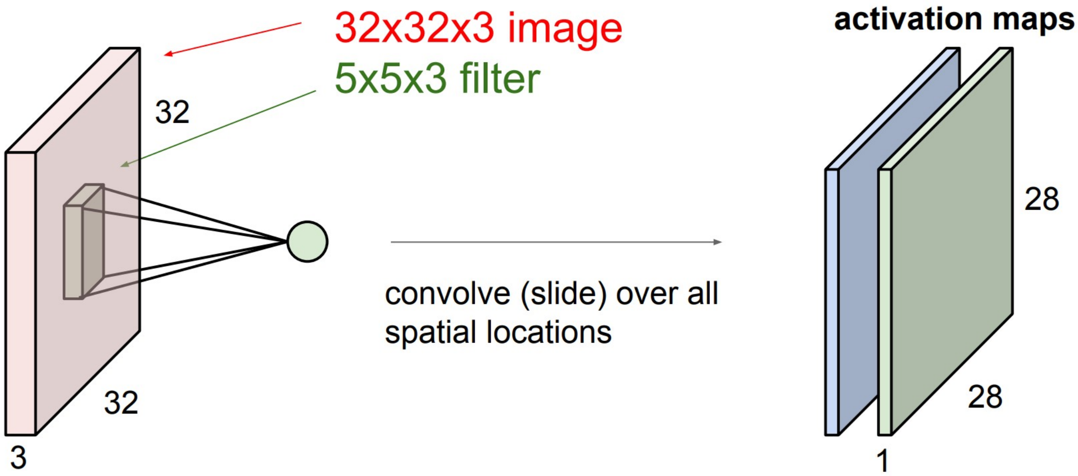

Figure III.10:. Activation maps
[29]

III.3.2.2 Pooling layer

Pooling layers aim to gradually reduce the dimensionality of the representation, and thus further reduce the number of parameters and the computational complexity of the model.[23]

Pooling works very much like convolution, where we take a kernel and move the kernel over the image, the only difference is the function that is applied to the kernel and the image window is not linear.

Max pooling and Average pooling are the most common pooling functions. Max pooling takes the largest value from the window of the image currently covered by the kernel, while average pooling takes the average of all values in the window.[34]

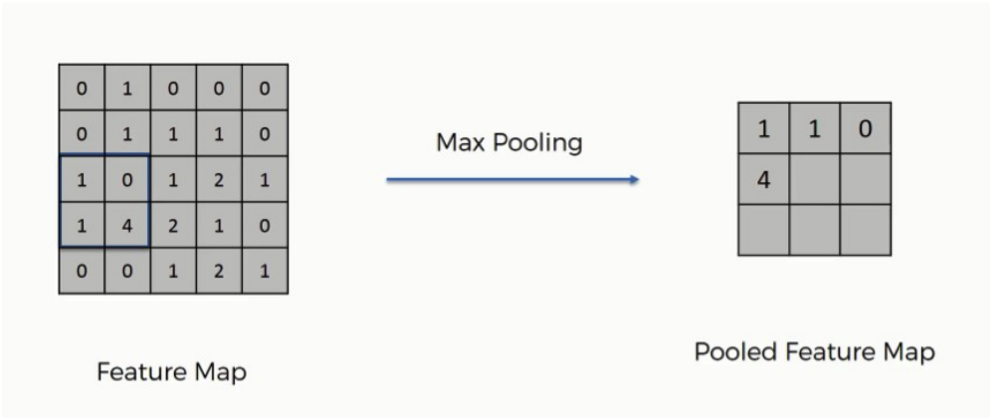

Figure III.11:. Pooling with a kernel of 2*2 and a stride of 2
[21]

In most CNNs, these come in the form of max-pooling layers with kernels of a dimensionality of 2*2 applied with a stride of 2 along the spatial dimensions of the input. This scales the activation map down to 25% of the original size - whilst maintaining the depth volume to its standard size. [23]

stride : is The distance the window moves each time.

III.3.2.3 Fully connected layer

After the feature extraction phase there is a classification phase, which is done by a fully connected layer. A fully connected layer is basically a layer that has neurons that are fully connected to the previous layer (feature map) without being connected to each other.

In the case of supervised learning, This last layer contains N neurons (number of classes in the database), and a sigmoid-type activation function is used to obtain probabilities of belonging to each class.[2]

III.3.3 CNN architectures

Many CNN architectures have been used in image classification through the years, and each one of the them has maximized the performance of image classification in its own way. some of the famous CNN architectures are the following :

III.3.3.1 AlexNet (2012)

AlexNet uses ReLu activation function instead of tanh to add non-linearity, which accelerated the speed of training (by 6 times) and increased the accuracy. It also uses dropout regularisation (a technique prevents complex co-adaptations on training data to reduce overfitting). Another feature of AlexNet is that it overlaps pooling to reduce the size of the network. It reduces the top-1 and top-5 error rates by 0.4 per cent and 0.3 per cent, respectively.[35]

The net contains eight layers with weights; the first five are convolutional and the remaining three are fully connected. The output of the last fully-connected layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels. Our network maximizes the multinomial logistic regression objective, which is equivalent to maximizing the average across training cases of the log-probability of the correct label under the prediction distribution[36]

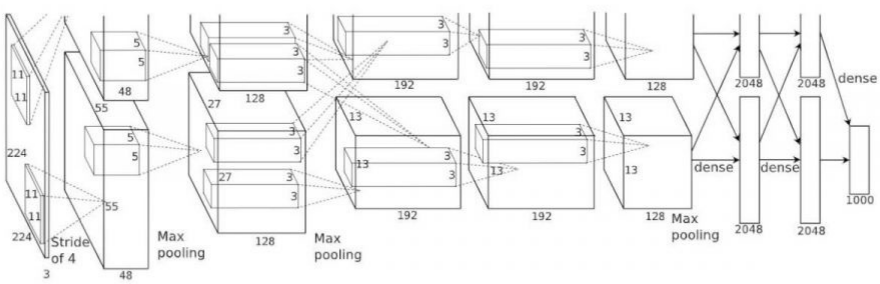

Figure III.12: . AlexNet Architecture

An illustration of the architecture of AlexNet CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150, 528-dimensional,and the number of neurons in the network's remaining layers is given by 253, 440–186, 624–64, 896–64, 896–43, 264–4096–4096–1000. [37]

III.3.3.2 GoogLeNet/Inception(2014)

GoogLeNet is developed based on the idea that several connections between layers are ineffective and have redundant information due to the correlation between them. Accordingly, it uses an “Inception module”, a sparse CNN, with 22 layers in a parallel processing workflow, and benefits from several auxiliary classifiers within the intermediate layers to improve the discrimination capacity in the lower layers. In contrast to conventional CNNs such as AlexNet and VGG, wherein either a convolutional or a pooling operation can be used at each level, the Inception module could benefit from both at each layer. Furthermore, filters (convolutions) with varying sizes are used at the same layer, providing more detailed information and extracting patterns with different sizes.

Importantly, a 1 x 1 convolutional layer, the so-called bottleneck layer, was employed to decrease both the computational complexity and the number of parameters. To be more precise, 1 x 1 convolutional layers were used just before a larger kernel convolutional filter (e.g., 3 x 3 and 5 x 5 convolutional layers) to decrease the number of parameters to be determined at each level (i.e., the pooling feature process).

In addition, 1 x 1 convolutional layers make the network deeper and add more non-linearity by using ReLU after each 1 x 1 convolutional layer. In this network, the fully connected layers are replaced with an average pooling layer.This significantly decreases the number of parameters since the fully connected layers include a large number of parameters. Thus, this network is able to learn deeper representations of features with fewer parameters relative to AlexNet while it is much faster than VGG.[38]

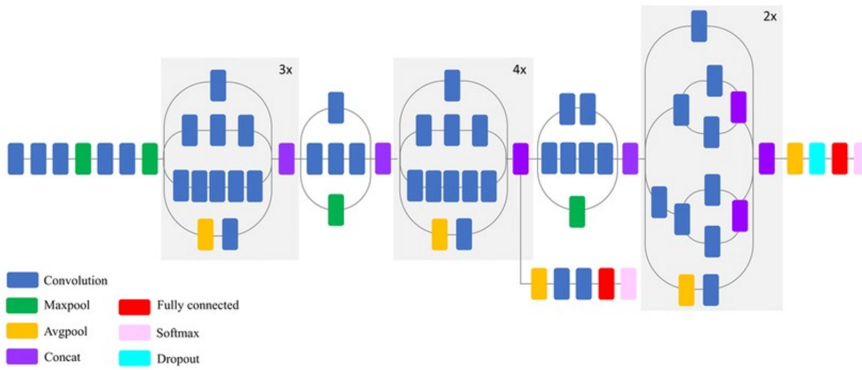

Figure III.13:. Compressed view of the Architecture of GoogLeNet (version 3) [38]

III.3.3.3 ResNet(2015)

Residual Neural Network (ResNet) by Kaiming He et al introduces an architecture which consists of 152 layers with skip connections(gated units or gated recurrent units) and features heavy batch normalization. The whole idea of ResNet is to counter the problem of vanishing gradients. By preserving the gradients, Vanishing gradients is the problem that occurs in networks with high number of layers as the weights of the first layers cannot be updated correctly through the backpropagation of the error gradient (the chain rule multiplies error gradient values lower than one and then, when the gradient error comes to the first layers, its value goes to zero).[35]

The deep ResNet configuration addresses the vanishing gradient problem by employing a deep residual learning module via additive identity transformations. Specifically, the residual module uses a direct path between the input and output and each stacked layer fits a residual mapping rather than directly fitting a desired underlying mapping. Notably, the optimization is much easier on the residual map relative to the original, unreferenced map. Similar to VGG, 3 x 3 filters were mostly employed in this network; however, ResNet has fewer filters and less complexity relative to the VGG network.[38]

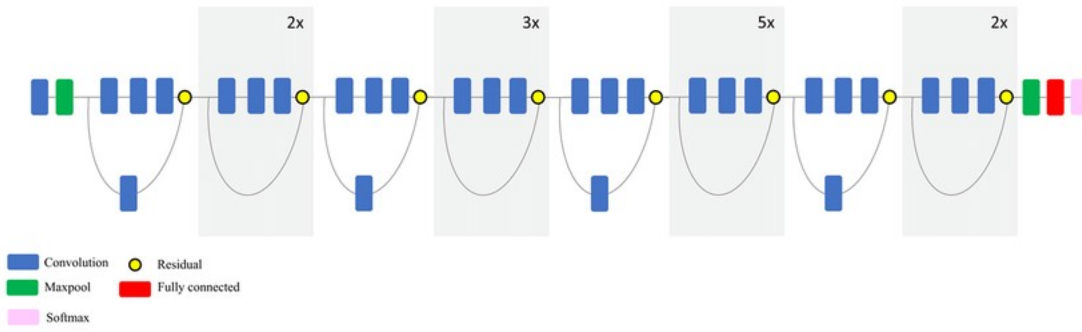

Figure III.14:. Compressed view of the Architecture of ResNet [38]

III.3.3.4 VGGNet (2014)

VGGNet was invented by VGG (Visual Geometry Group) from the University of Oxford.
The image is passed through a stack of convolutional layers, where we use filters with a very small receptive field: 3 x 3 (which is the smallest size to capture the notion of left/right, up/down, center). the padding is 1 pixel for 3 x 3 conv layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2 x 2 pixel window, with stride 2. A stack of convolutional layers (which has a different depth in different architectures) is followed by three Fully-Connected (FC) layers. All hidden layers are equipped with the rectification (ReLU) non-linearity. All configurations follow the generic design presented above, and differ only in the depth: from 11 weight layers in the network (8 conv. and 3 FC layers) to 19 weight layers in the network (16 conv. and 3 FC layers).[17]

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure III.15:. ConvNet configurations.
The convolutional layer parameters are denoted as “conv (receptive field size)-(hnumber of channels)”.The ReLU activation function is not shown for brevity.[17]

The Convolutional neural network that we will be using is the VGG-NET, we will be using a much smaller version of it, but it will have the main characterestics of the full resolution network:

- Using 3 x 3 convolutional layers.
- reducing volume size by max pooling.
- fully connected layers at the end.
- a softmax classifier.

III.3.4 VggNet CNN Classification

CNN image classifications takes an input image, process it and classify it under certain categories (classes). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d(h = Height, w = Width, d = Depth).

III.3.4.1 VggNet Model training

The ConvNet training procedure generally follows Krizhevsky et al. (2012) (except for sampling the input crops from multi-scale training images, as explained later). Namely, the training is carried out by optimising the multinomial logistic regression objective using mini-batch gradient descent (based on back-propagation (LeCun et al., 1989)) with momentum. The batch size was set to 256, momentum to 0.9. The training was regularised by weight decay (the L2 penalty multiplier set to 5 · 10-4) and dropout regularisation for the first two fully-connected layers (dropout ratio set to 0.5).

The learning rate was initially set to 10-2, and then decreased by a factor of 10 when the validation set accuracy stopped improving. In total, the learning rate was decreased 3 times, and the learning was stopped after 370K iterations (74 epochs). We conjecture that in spite of the larger number of parameters and the greater depth of our nets compared to (Krizhevsky et al., 2012), the nets required less epochs to converge due to (a) implicit regularisation imposed by greater depth and smaller conv. filter sizes; (b) pre-initialisation of certain layers. The initialisation of the network weights is important, since bad initialisation can stall learning due to the instability of gradient in deep nets. To circumvent this problem, we began with training the configuration A (Table III.15), shallow enough to be trained with random initialisation. Then, when training deeper architectures, we initialised the first four convolutional layers and the last three fully connected layers with the layers of net A (the intermediate layers were initialised randomly). We did not decrease the learning rate for the pre-initialised layers, allowing them to change during learning.

For random initialisation (where applicable), we sampled the weights from a normal distribution with the zero mean and 10-2 variance. The biases were initialised with zero. It is worth noting that after the paper submission we found that it is possible to initialise the weights without pretraining by using the random initialisation procedure of Glorot and Bengio (2010). To obtain the fixed-size 224x224 ConvNet input images, they were randomly cropped from rescaled training images (one crop per image per SGD iteration). To further augment the training set, the crops underwent random horizontal flipping and random RGB colour shift (Krizhevsky et al., 2012). Training image rescaling is explained below.

Training image size. Let S be the smallest side of an isotropic-ally rescaled training image, from which the ConvNet input is cropped (we also refer to S as the training scale). While the crop size is fixed to 224 x 224, in principle S can take on any value not less than 224: for S = 224 the crop will capture whole-image statistics, completely spanning the smallest side of a training image; for S ¿¿ 224 the crop will correspond to a small part of the image, containing a small object or an object part. We consider two approaches for setting the training scale S. The first is to fix S, which corresponds to single-scale training (note that image content within the sampled crops can still represent multiscale image statistics). In our experiments, we evaluated models trained at two fixed scales: S = 256 (which has been widely used in the prior art (Krizhevsky et al., 2012; Zeiler and Fergus, 2013; Sermanet et al., 2014)) and S = 384. Given a ConvNet configuration, we first trained the network using S = 256. To speed-up training of the S = 384 network, it was initialised with the weights pretrained with S = 256, and we used a smaller initial learning rate of 10-3. The second approach to setting S is multi-scale training, where each training image is individually rescaled by randomly sampling S from a certain range [Smin, Smax] (we used Smin = 256 and Smax = 512). Since objects in images can be of different size, it is beneficial to take this into account during training. This can also be seen as training set augmentation by scale jittering, where a single model is trained to recognise objects over a wide range of scales. For speed reasons, we trained multi-scale models by fine-tuning all layers of a single-scale model with the same configuration, pre-trained with fixed S = 384.[17]

III.3.4.2 VggNet Model testing

At test time, given a trained ConvNet and an input image, it is classified in the following way. First, it is isotropically rescaled to a pre-defined smallest image side, denoted as Q (we also refer to it as the test scale). We note that Q is not necessarily equal to the training scale S (as we will show in Sect.4, using several values of Q for each S leads to improved performance). Then, the network is applied densely over the rescaled test image in a way similar to (Sermanet et al., 2014). Namely, the fully-connected layers are first converted to convolutional layers (the first FC layer to a 7 x 7 conv. layer, the last two FC layers to 1 x 1 conv. layers).

The resulting fully-convolutional net is then applied to the whole (uncropped) image. The result is a class score map with the number of channels equal to the number of classes, and a variable spatial resolution, dependent on the input image size. Finally, to obtain a fixed-size vector of class scores for the image, the class score map is spatially averaged (sum-pooled). We also augment the test set by horizontal flipping of the images; the soft-max class posteriors of the original and flipped images are averaged to obtain the final scores for the image.

Since the fully-convolutional network is applied over the whole image, there is no need to sample multiple crops at test time (Krizhevsky et al., 2012), which is less efficient as it requires network re-computation for each crop. At the same time, using a large set of crops, as done by Szegedy et al. (2014), can lead to improved accuracy, as it results in a finer sampling of the input image compared to the fully-convolutional net.

Also, multi-crop evaluation is complementary to dense evaluation due to different convolution boundary conditions: when applying a ConvNet to a crop, the convolved feature maps are padded with zeros, while in the case of dense evaluation the padding for the same crop naturally comes from the neighbouring parts of an image (due to both the convolutions and spatial pooling), which substantially increases the overall network receptive field, so more context is captured. While we believe that in practice the increased computation time of multiple crops does not justify the potential gains in accuracy, for reference we also evaluate our networks using 50 crops per scale (5 x 5 regular grid with 2 flips), for a total of 150 crops over 3 scales, which is comparable to 144 crops over 4 scales used by Szegedy et al. (2014). [17]

III.3.4.3 Non Linearity

Activation functions are really important for a Artificial Neural Network to learn and make sense of something really complicated and Non-linear complex functional mappings between the inputs and response variable. They introduce non-linear properties to our Network. Their main purpose is to convert a input signal of a node in a A-NN to an output signal. That output signal now is used as a input in the next layer in the stack.

If we do not apply a Activation function then the output signal would simply be a simple linear function. A linear function is just a polynomial of one degree. Now, a linear equation is easy to solve but they are limited in their complexity and have less power to learn complex functional mappings from data. A Neural Network without Activation function would simply be

a Linear regression Model, which has limited power and does not performs good most of the times. We want our Neural Network to not just learn and compute a linear function but something more complicated than that. Also without activation function our Neural network would not be able to learn and model other complicated kinds of data such as images, videos , audio , speech etc.
Hence it allcomes down to this, we need to apply a Activation function $f(x)$ so as to make the network more powerfull and add ability to it to learn something complex and complicated form data and represent non-linear complex arbitrary functional mappings between inputs and outputs. Hence using a non linear Activation we are able to generate non-linear mappings from inputs to outputs[39]

ReLU (Rectified Linear Unit): ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = max(0, x)$. Why ReLU is important : ReLU's purpose is to introduce non-linearity in our ConvNet.There are other non linear functions such as tanh or sigmoid can also be used instead of ReLU. Most of the data scientists use ReLU since performance wise ReLU is better than the other two.[40]

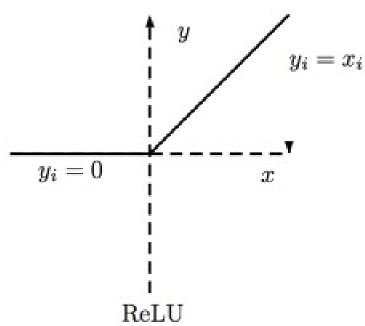

Figure III.16:. A ReLU activation function
[40]

Hence for output layers we should use a Softmax function for a Classification problem to compute the probabilites for the classes[39]

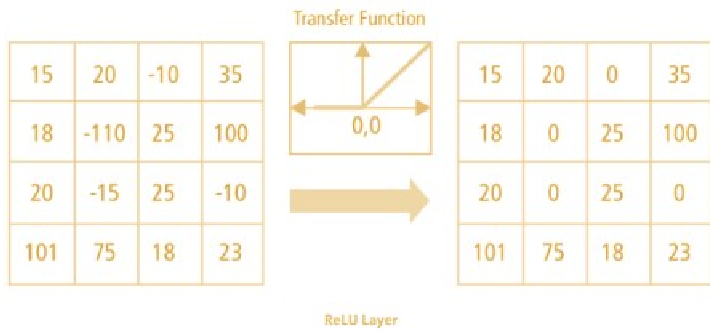

Figure III.17:. ReLU operation
[40]

III.3.4.4 Softmax Function:

Softmax function takes an N-dimensional vector of real numbers and transforms it into a vector of real number in range (0,1) which add upto 1.

$$p_i = \frac{e^{a_i}}{\sum_{K=1}^{N} e^{a_k}} \qquad \text{(III.3)}$$

As the name suggests, softmax function is a "soft" version of max function. Instead of selecting one maximum value, it breaks the whole (1) with maximal element getting the largest portion of the distribution, but other smaller elements getting some of it as well.

This property of softmax function that it outputs a probability distribution makes it suitable for probabilistic interpretation in classification tasks.[41]

III.3.4.5 Cross Entropy Loss:

Cross entropy indicates the distance between what the model believes the output distribution should be, and what the original distribution really is. It is defined as, $H(y, p) = -\sum_i yilog(pi)$ Cross entropy measure is a widely used alternative of squared error. It is used when node activations can be understood as representing the probability that each hypothesis might be true, i.e. when the output is a probability distribution. Thus it is used as a loss function in neural networks which have softmax activations in the output layer.[41]

III.3.5 Conclusion

After we had an idea of how ANNs basically work and their role in the deep learning field, we know that CNNs are the most used and the best neural networks to deal with image recognition and pattern detection because of their features.

In the next part we will see an implementation of CNNs to do a facial recognition task on a certain database.