



Département Informatique et Gestion Filière Sciences Mathématiques  
et Informatique

Etude de l'intelligence Artificielle et Conception et  
réalisation d'une application Desktop pour la  
reconnaissance et d'identification faciale.

El allaoui Mohamed  
Naji Abderrahim  
EL-Adarissi Abdelaziz

El allaoui Mohamed  
Naji Abderrahim  
EL-Adarissi Abdelaziz

Sous la direction de  
Prof.Salma Gaou

Soutenu le : ...2022

Devant le jury :

Prof.Salma Gaou :Professeur à la Faculté Polydisciplinaire de Ouarzazate

Prof. Ayoub Ellhyani : Professeur à la Faculté Polydisciplinaire de Ouarzazate

Année Universitaire : 2021-2022

# Table des matières

<b>Remerciements</b>	<b>6</b>
<b>1 Introduction</b>	<b>8</b>
1.1 Intelligence Artificielle . . . . .	8
1.1.1 Types De L'intelligence Artificiel . . . . .	9
1.2 Machine Learning . . . . .	9
1.2.1 Définition . . . . .	9
1.2.2 Méthodes d'apprentissage . . . . .	9
1.2.3 Fonctionnement D'apprentissage Automatique . . . . .	10
1.3 L'apprentissage Profond . . . . .	11
1.3.1 Définition . . . . .	11
1.3.2 Applications du deep Learning . . . . .	12
1.4 Réseau de Neurones . . . . .	12
1.4.1 Définition . . . . .	12
1.4.2 Comment fonctionne le réseau de neurones artificiels . . . . .	12
1.4.3 Structure D'un Réseaux De Neurone . . . . .	13
1.4.4 Fonction d'activation . . . . .	13
1.4.5 Les types de réseaux neuronaux . . . . .	15
<b>2 Méthodes de Détection et Comparaison des visages</b>	<b>16</b>
2.1 Notions pour le traitement des images . . . . .	16
2.1.1 Définition d'une image . . . . .	16
2.1.2 L'image numérique . . . . .	17
2.1.3 Caractéristiques d'une image numérique . . . . .	17
2.1.4 Matrices et images numériques . . . . .	19
2.2 Détection des visages . . . . .	20
2.2.1 Définition . . . . .	20
2.2.2 Classification des méthodes de détection de visage . . . . .	20
2.2.3 Algorithme de MTCNN . . . . .	21
2.2.4 algorithme HOG . . . . .	21
2.3 Linéaire Machine à vecteur de support . . . . .	22
2.3.1 Définition . . . . .	22
2.3.2 Fonctionnement des machines à vecteur support . . . . .	23
2.3.3 Avantages . . . . .	24
2.3.4 Inconvénients . . . . .	24
2.3.5 Les domaines d'applications . . . . .	24
2.4 Comparaison des visages . . . . .	24

<b>3</b>	<b>Conception</b>	<b>26</b>
3.0.1	Diagramme de cas d'utilisation . . . . .	26
3.0.2	Diagramme de Séquence . . . . .	27
3.0.3	Diagramme de Classe . . . . .	27
<b>4</b>	<b>Implémentation</b>	<b>29</b>
4.0.1	Logiciels et librairies Utilisés dans l'implémentation : . . . . .	29
4.0.2	Présentation De L'interface Graphique : . . . . .	32
4.1	conclusion . . . . .	32

# Table des figures

1.1	Process . . . . .	9
1.2	Exemple de Types de Apprentissage Supervisé . . . . .	10
1.3	Exemple de Types de Apprentissage non Supervisé . . . . .	10
1.4	Machine Learning Process . . . . .	11
1.5	Exemple réseaux de neurone . . . . .	12
1.6	structure interne d'un réseaux de neurones . . . . .	14
1.7	Exemple des Fonction d'Activation . . . . .	14
2.1	histogramme d'image . . . . .	18
2.2	Exemple de svm . . . . .	23
3.1	Diagramme cas d'utilisation . . . . .	26
3.2	Diagramme de Séquence . . . . .	27
3.3	Diagramme de Class . . . . .	28
4.1	Python . . . . .	29
4.2	OpenCv . . . . .	29
4.3	Pillow . . . . .	30
4.4	Tkinter . . . . .	30
4.5	Numpy . . . . .	30
4.6	PyCharm . . . . .	31
4.7	Visual Studio Code . . . . .	31
4.8	. . . . .	32

# Liste des tableaux

# Remerciements

*Avant tous, nous remercions notre créateur le tout puissant qui nous a donné la force et la volonté pour pouvoir finir ce proje.*

*Nous tenons à remercier vivement notre encadrante **Gaou Salma** d'avoir accepté de diriger ce projet et pour la confiance qu'elle nous a accordée, ses encouragements et ses précieux conseils.*

*Nous vaudrions remercier également les membres de jury de soutenance de ce projet en particulier monsieur **Ellhyani Ayoub** d'avoir accepté d'examiner notre travail.*

*Nous tenons à remercier, tous ceux qui nous ont enseigné durant toutes notre études et en particuliers nos enseignants à Faculté Polydisciplinaire de Ouarzazate.*

*Nous tenons aussi à remercier tous les personnes qui nous ont encouragés pendant la réalisation de ce travail, famille, collège, amis, sans exception.*

# Introduction générale

De toute l'histoire de l'humanité, l'accès à la technologie, aux capitaux et à la connaissance n'ont jamais été aussi massifs. C'est ainsi que nous avons vécu, ces cinquante dernières années, la plus formidable accélération que notre humanité n'ait jamais vécue. Et l'intelligence artificielle constituera, sans aucun doute, le moteur d'une accélération encore plus importante. celle-ci est devenue un très large domaine de l'informatique, et indispensable dans notre vie quotidienne et pratique et tout les sociétés a travers le monde s'appuie sur l'intelligence artificielle

L'intelligence artificielle est très vaste qu'aucun rapport ou deux ou trois ne peuvent l'entourer et puisque le traitement visuelle ou bien le "computer vision" est Parmi les utilisations les plus courantes de l'intelligence artificielle on va intéresser au domaine de la reconnaissance et de l'identification des visages. Nous devons rechercher une personne dans une vidéo en streaming ou déjà stoker sur une mémoire, pour le faire nous devons entraîner notre machine à des milliers de visages, mais dans le cadre de la recherche, nous n'avons qu'une seule image de la personne que nous recherchons.

Cette technologie sert à bien des choses. Cela inclut :

- Le déverrouillage des téléphones
- Application de la loi
- Contrôles dans les aéroports et aux frontières
- Retrouver des personnes disparues
- Réduire la criminalité dans les magasins
- Améliorer les expériences d'achat
- Services bancaires
- Reconnaissance des conducteurs
- Santé
- Surveiller la présence des employés ou des élèves
- etc...

Nous devons utiliser un système de comparaison de visages comme le système faceNet. Nous allons commencer par la réduction du bruit en vidéo. La détection des visages sera la prochaine étape et la comparaison des visages sur les visages détectés sera la dernière étape. Il fonctionnera sur les vidéos stockées et une image d'entrée avec le visage à rechercher.

mais tout d'abord on va regarder de près l'étonnant intelligence artificielle

# Chapitre 1

## Introduction

### 1.1 Intelligence Artificielle

selon la définition de l'Association française pour l'intelligence artificielle celle-ci est : « l'ensemble de théories et de techniques mises en oeuvre en vue de réaliser des machines capables de simuler l'intelligence humaines ».

Les uns utilisent le mot pour désigner le machine learning (l'apprentissage automatique), d'autres pour expliquer le computer vision (la vision par le numérique), le natural language processing (le traitement du langage naturel) ou les simples algorithmes consubstantiels aux programmes informatiques classiques. La vérité est que l'intelligence artificielle ne se résume à aucun de ces éléments, mais elle est tout cela à la fois et plus encore. On peut donc parler d'un écosystème dans toute sa diversité.

Historiquement, On peut considérer que le point de départ de l'intelligence artificielle a été donné avec les travaux d'Alan Turing qui posait le paradigme qu'une machine pouvait penser, notamment, dans son article « Computing Machinery and Intelligence », publié en 1950. Il propose un test, connu maintenant comme test de Turing, pour définir si la machine sait raisonner et peut se créer une autonomie.

Pour être plus précis, disons que l'intelligence artificielle est un programme informatique pas comme les autres. Contrairement à un programme informatique traditionnel, l'IA ne se programme pas, elle « s'éduque ». Cette « éducation » se fait par des méthodes d'apprentissage machine (ou apprentissage automatique, ou machine learning) qui est un composant nécessaire de l'intelligence artificielle aujourd'hui. Dans la conception actuelle, il n'y a pas d'intelligence artificielle sans apprentissage machine préalable. et que nous détaillerons un peu plus loin. L'apprentissage aboutit au paramétrage du programme. Lorsque l'apprentissage est continu au cours du cycle de vie du programme, on constate un effet d'adaptation à l'environnement propre à faire penser à nos propres qualités d'intelligence.

Ces solutions ne sont rien d'autre que des assemblages « sur mesure » de plusieurs briques logicielles (algorithmes) correspondant à un besoin spécifique et donc ayant fait l'objet d'un entraînement (paramétrage) spécifique. Les algorithmes sont très nombreux et en grande partie disponibles en open source. Ils sont classés par type d'actions qu'ils sont capables d'effectuer, allant de la détection et reconnaissance de langue au traitement de son, d'image, à l'exploitation de bases



de données, etc. Ils sont combinés pour rendre un service spécifique, par ex., transformer la parole en texte, analyser un texte ou décrire une image, etc. Chaque solution est entraînée (« éduquée ») pour un domaine particulier pour effectuer des tâches très spécifiques plus efficacement, c'est-à-dire, plus vite, en grande quantité et à moindre coût qu'un humain ne peut le faire. L'intégration des algorithmes est un enjeu technique d'une grande complexité.

### 1.1.1 Types De L'intelligence Artificiel

#### 1. intelligence artificielle forte

Compétente dans tous les domaines qui égalerait et dépasserait l'intelligence humaine.

#### 2. intelligence artificielle faible

Elle correspondrait à la capacité des machines à traiter de problèmes dans un domaine précis. Tous les exemples de solutions actuelles relèvent de l'IA faible, c'est-à-dire spécialisée, limitée à une seule compétence.

## 1.2 Machine Learning

### 1.2.1 Définition

Le Machine Learning ou apprentissage automatique est un domaine scientifique, et plus particulièrement une sous-catégorie de l'intelligence artificielle. Elle consiste à laisser des algorithmes découvrir des "patterns", à savoir des motifs récurrents, dans les ensembles de données. Ces données peuvent être des chiffres, des mots, des images, des statistiques...

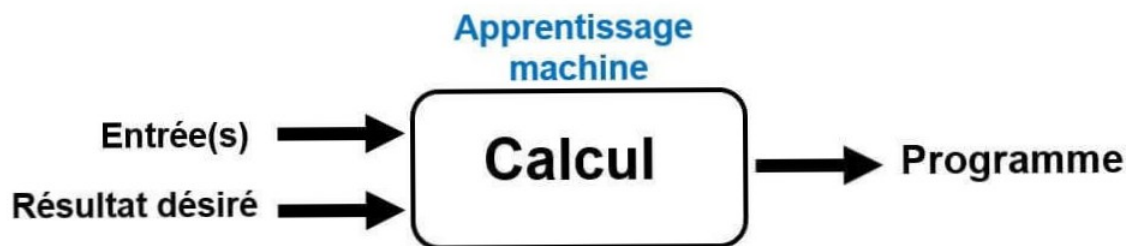


FIGURE 1.1 – Process

### 1.2.2 Méthodes d'apprentissage

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage qu'ils emploient généralement on deux grands types sont :

#### Apprentissage Supervisé :

Dans ce type d'apprentissage automatique, On donne à l'algorithme un certain nombre d'exemples (input) sur lesquels apprendre, et ces exemples sont « labellisés », c'est-à-dire qu'on leur associe un résultat désiré (output). L'algorithme a alors pour tâche de trouver la loi qui permet de trouver l'output en fonction des inputs.

## Types de Apprentissage Supervisé :

Le machine learning avec supervision peut se subdiviser en deux types :

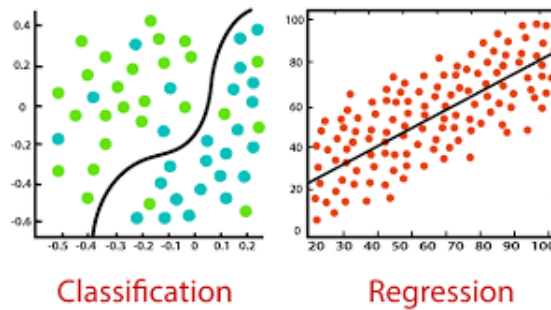


FIGURE 1.2 – Exemple de Types de Apprentissage Supervisé

1. **La classification** : la variable de sortie est catégorie
2. **La régression** : la variable de sortie est une valeur spécifiée

## Apprentissage Non Supervisé :

Dans ce type d'apprentissage automatique, l'algorithme est laissé à lui-même pour déterminer la structure de l'entrée (aucun label n'est communiqué à l'algorithme). Cette approche peut être un but en soi (qui permet de découvrir des structures enfouies dans les données) ou un moyen d'atteindre un certain but. Cette approche est également appelée « apprentissage des caractéristiques » (feature learning). Le machine learning sans supervision peut se subdiviser en deux types :

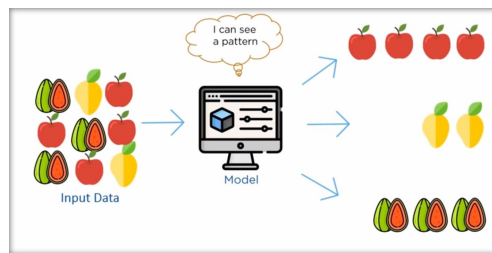


FIGURE 1.3 – Exemple de Types de Apprentissage non Supervisé

1. **Clustering** : L'objectif consiste à trouver des regroupements dans les données.
2. **Association** : L'objectif consiste à identifier les règles qui permettront de définir de grands groupes de données.

### 1.2.3 Fonctionnement D'apprentissage Automatique

Le développement d'un modèle de Machine Learning repose sur quatre étapes principales :

1. **La première étape** : consiste à sélectionner et à préparer un ensemble de données d'entraînement. Ces données seront utilisées pour nourrir le modèle de Machine Learning pour apprendre à résoudre le problème pour lequel il est conçu.
2. **La deuxième étape** : consiste à sélectionner un algorithme à exécuter sur l'ensemble de données d'entraînement. Le type d'algorithme à utiliser dépend du type et du volume de données d'entraînement et du type de problème à résoudre.
3. **La troisième étape** : est l'entraînement de l'algorithme. Il s'agit d'un processus itératif. Des variables sont exécutées à travers l'algorithme, et les résultats sont comparés avec ceux qu'il aurait du produire. On exécute ensuite de nouveau les variables jusqu'à ce que l'algorithme produise le résultat correct la plupart du temps. L'algorithme, ainsi entraîné, est le modèle de Machine Learning
4. **La quatrième étape** : est l'utilisation et l'amélioration du modèle. On utilise le modèle sur de nouvelles données, dont la provenance dépend du problème à résoudre. Par exemple, un modèle de Machine Learning conçu pour détecter les spams sera utilisé sur des emails.

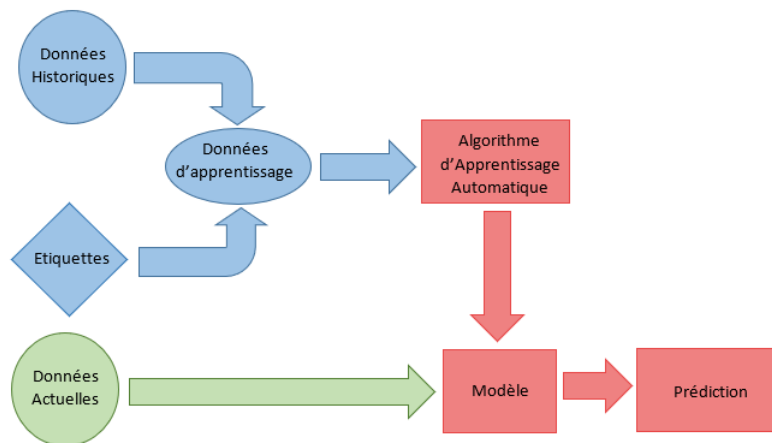


FIGURE 1.4 – Machine Learning Process

## 1.3 L'apprentissage Profond

### 1.3.1 Définition

Apprentissage en profondeur (en anglais : deep learning, deep structured learning, hierarchical learning) est un ensemble des méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, reconnaissance vocale, la vision par ordinateur, et traitement automatisé du langage. Dans les années 2000, ces progrès ont suscité des investissements privés, universitaires et publics importants, notamment de la part des GAFAM (Google, Apple, Facebook, Amazon, Microsoft).

### 1.3.2 Applications du deep Learning

Le deep Learning est utilisé dans de nombreux domaines :

- reconnaissance d'images
- traduction automatique
- voiture autonome
- diagnostic médical
- recommandations personnalisées
- modération automatique des réseaux sociaux
- prédiction financière et trading automatisé

## 1.4 Réseau de Neurones

### 1.4.1 Définition

Réseau de neurone est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques.

Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type probabiliste, en particulier bayésien. Ils sont placés d'une part dans la famille des applications statistiques, qu'ils enrichissent avec un ensemble de paradigmes permettant de créer des classifications rapides (réseaux de Kohonen en particulier), et d'autre part dans la famille des méthodes de l'intelligence artificielle auxquelles ils fournissent un mécanisme perceptif indépendant des idées propres de l'implémenteur, et des informations d'entrée au raisonnement logique formel.

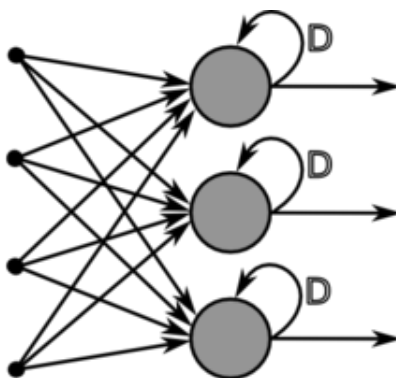


FIGURE 1.5 – Exemple réseaux de neurone

### 1.4.2 Comment fonctionne le réseau de neurones artificiels

La conception des réseaux de neurones artificiels s'appuie sur la structure des neurones biologiques du cerveau humain.

Les réseaux de neurones artificiels peuvent être décrits comme des systèmes composés d'au moins deux couches de neurones - une couche d'entrée et une couche de sortie - et comprenant généralement des couches intermédiaires - hidden layers -. Plus le problème à résoudre est complexe,

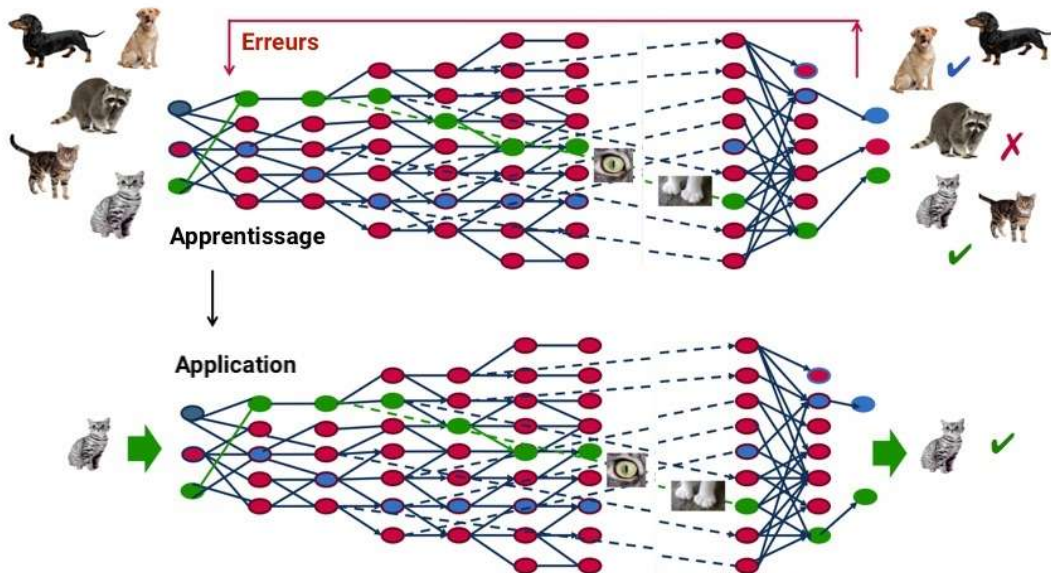
plus le réseau de neurones artificiels doit comporter de couches. Chaque couche contient un grand nombre de neurones artificiels spécialisés.

### Traitement de l'information au sein du réseau de neurones

Au sein d'un réseau de neurones artificiels, le traitement de l'information suit toujours la même séquence : les informations sont transmises sous la forme de signaux aux neurones de la couche d'entrée, où elles sont traitées. À chaque neurone est attribué un poids particulier, et donc une importance différente. Associé à la fonction dite de transfert, le poids permet de déterminer quelles informations peuvent entrer dans le système.

À l'étape suivante, une fonction dite d'activation associée à une valeur seuil calculent et pondèrent la valeur de sortie du neurone. En fonction de cette valeur, un nombre plus ou moins grand de neurones sont connectés et activés.

Cette connexion et cette pondération dessinent un algorithme qui fait correspondre un résultat à chaque entrée. Chaque nouvelle itération permet d'ajuster la pondération et donc l'algorithme de façon à ce que le réseau donne à chaque fois un résultat plus précis et fiable.



Figure[2] fonctionnement du réseaux de neurones

### 1.4.3 Structure D'un Réseaux De Neurone

Figure 1.5 montre la structure d'un neurone artificiel. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amonts. À chacune de ces entrées est associée un poids  $w$  (abréviation de weight (poids en anglais)) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. À chaque connexion est associée un poids.

### 1.4.4 Fonction d'activation

Dans le domaine des réseaux de neurones artificiels, la fonction d'activation est une fonction mathématique appliquée à un signal en sortie d'un neurone artificiel. Le terme de "fonction d'ac-

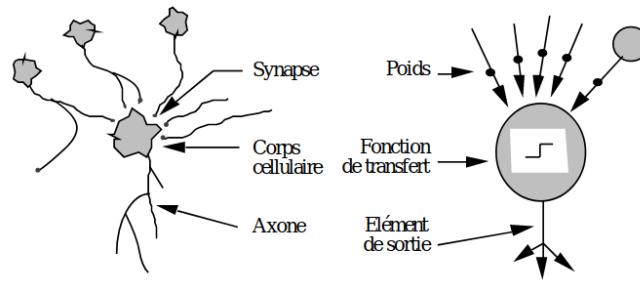


FIGURE 1.6 – structure interne d'un réseaux de neurones

tivation" vient de l'équivalent biologique "potentiel d'activation", seuil de stimulation qui, une fois atteint entraîne une réponse du neurone. La fonction d'activation est souvent une fonction non linéaire. Un exemple de fonction d'activation est la fonction de Heaviside, qui renvoie tout le temps 1 si le signal en entrée est positif, ou 0 s'il est négatif.

### caractéristiques d'un fonction d'activation

- Non-linéarité
- Partout différentiable
- Étendue
- Monotone
- Douce (dérivée monotone)
- Identité en 0 ( $f(x) = 0$  quand  $x = 0$ )

### Liste de fonctions d'activation usuelles

Nom	Graphes	Équation	Dérivée
Identité/Rampe		$f(x) = x$	$f'(x) = 1$
Marche/Heaviside		$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{si } x \neq 0 \\ ? & \text{si } x = 0 \end{cases}$
Logistique (ou marche douce, ou sigmoïde)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tangente hyperbolique		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
Arc tangente		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Signe doux <sup>6</sup>		$f(x) = \frac{x}{1 +  x }$	$f'(x) = \frac{1}{(1 +  x )^2}$
Unité de rectification linéaire (ReLU) <sup>7</sup>		$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$

FIGURE 1.7 – Exemple des Fonction d'Activation

### 1.4.5 Les types de réseaux neuronaux

- Les réseaux de neurones feed-forwarded
- Les réseaux de neurones récurrents
- Les réseaux de neurones à résonance
- Les réseaux de neurones auto-organisés

# Chapitre 2

## Méthodes de Détection et Comparaison des visages

La reconnaissance faciale est une technique d'identification ou de vérification d'une personne à l'aide de son visage à travers une image ou une vidéo. Il capture, analyse et compare les modèles en fonction des détails du visage de la personne. Le processus de détection de visage est une étape essentielle car il détecte et localise les visages humains dans les images et les vidéos. L'identification signifie comparer votre identité à toutes les identités présentes dans la base de données. Alors que la vérification fait référence à une comparaison de votre identité avec une correspondance particulière dans la base de données.

Dans notre proposition nous avons s'adapter à deux notions principales : la détection des visages et la comparaison faciale. Un système de détection des visages est conçu pour répondre à la question : cette image contient-elle un visage ? Un système de détection des visages détermine la présence, l'emplacement, les dimensions et (éventuellement) l'orientation de n'importe quel visage présent dans une image fixe ou une trame vidéo. Ce système est conçu pour détecter la présence de visages quels que soient leurs attributs : sexe, âge...

Un système de comparaison faciale est conçu pour répondre à la question : le visage présent dans une image correspond-il au visage d'une autre image ? Un système de comparaison faciale prend l'image d'un visage et prédit si ce visage correspond à d'autres visages figurant dans une base de données fournie. Les systèmes de comparaison faciale sont conçus pour comparer et prédire les éventuelles mises en correspondance de visages, quels que soient leur expression, et leur âge.

### 2.1 Notions pour le traitement des images

#### 2.1.1 Définition d'une image

C'est la représentation d'une personne ou d'un objet par la peinture, la sculpture, la photographie, etc. C'est aussi un ensemble structuré d'informations qui après affichage sur écran, ont une signification pour l'œil humain. Elle peut être écrite sous forme de fonction  $I(x,y)$  ou  $I$  est



une fonction d'intensité lumineuse ou de couleur aux coordonnées spatiales  $(x,y)$  . De cette façon l'image est exploitable par la machine, d'où la nécessité de sa numérisation.

### **2.1.2 L'image numérique**

Il est clair que les images manipulées par l'ordinateur sont numériques (série de bits). L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés pixels, ayant comme caractéristique le niveau de gris ou de couleur. La numérisation d'une image est la conversion de celle-ci en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques  $f(x,y)$  qui sont les niveaux de gris des coordonnées réelles  $(x,y)$ .

### **2.1.3 Caractéristiques d'une image numérique**

#### **Pixel**

Le pixel est le plus petit élément graphique composant une image, il est rempli par une couleur unique.

#### **Dimension**

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentant les intensités lumineuses (pixel).

#### **Résolution**

C'est la clarté ou la finesse des détails atteinte par un moniteur ou une imprimante dans la production d'image, sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixel par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichable horizontalement ou verticalement sur un moniteur ; plus grand est ce nombre, meilleure est la résolution.

#### **Niveaux de gris**

C'est la valeur numérique qui reflète l'intensité de la luminosité d'un point. Pour niveaux de gris compris entre 0 et 255, chaque pixel sera codé sur 8 bits, et ses niveaux de gris seront obtenus après dégradation de la couleur noire, représente le blanc et représente le noir.

#### **Bruit**

C'est un phénomène de brusques variations d'un pixel par rapport à ces voisins suivant un certain seuil. Il existe quatre sources de dégradation induisant le bruit, qui sont :

le bruit lié au contexte d'acquisition

le bruit lié au capteur

le bruit lié à l'échantillonnage,

## Histogramme

En imagerie numérique, l'histogramme représente la distribution des intensités (ou des couleurs) de l'image. C'est un outil fondamental du traitement d'images, avec de très nombreuses applications. Les histogrammes sont aussi très utilisés en photographie et pour la retouche d'images.

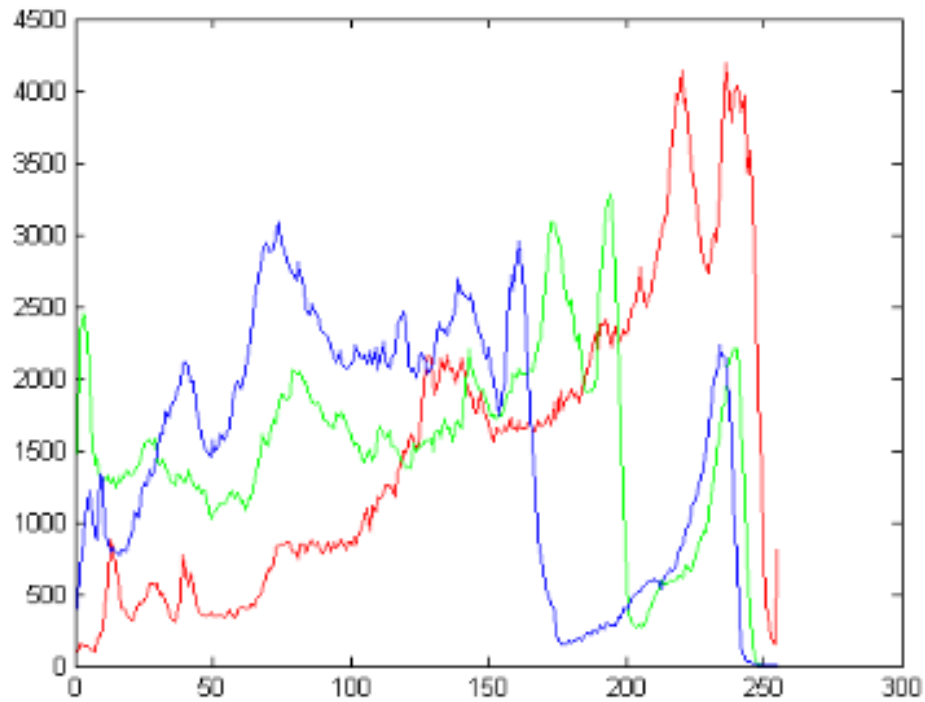


FIGURE 2.1 – histogramme d'image

## 2.1.4 Matrices et images numériques

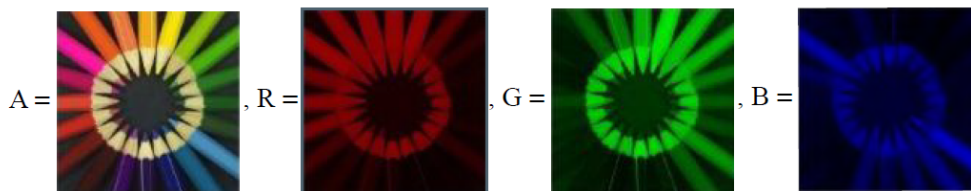
### Images binaires, en noir et blanc, et en couleurs

Les images que l'on trouve sur le net ainsi que celles des appareils numériques et des téléphones portables sont numérisées . Pour comprendre ce que cela signifie, on peut utiliser les matrices. Prenons par exemple les 4 images de mohamed .



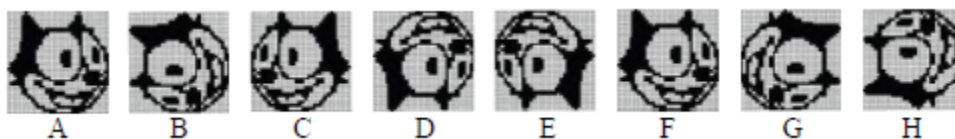
les 4 images sont successivement représentées par une matrice  $35 \times 35$  ,  $78 \times 98$  ,  $112 \times 123$  et  $400 \times 234$  dont les éléments valent 0 ou 1, 0 donnant un pixel noir et 1 donnant un pixel blanc ; la première image sera ainsi constituée de  $35 \times 35 = 1225$  pixels. De telles images sont nommées images binaires, ou images booléennes.

Les images dites en noir et blanc sont en fait des images en dégradé de gris. Elles peuvent aussi être représentées par des matrices : chaque élément de la matrice correspondra comme précédemment à un pixel, mais ces éléments seront des nombres entiers donnant l'intensité de gris voulue. On utilise habituellement des nombres de 0 à 255, 0 pour le noir et 255 pour le blanc, soit  $256 = 2^8$  niveaux de gris, ce qui permet de coder l'image par  $8N$  bits, si  $N$  est le nombre de pixels. Les images en couleur sont représentées par trois matrices de taille identique, donnant l'intensité de Rouge, Vert, et Bleu pour chaque pixel. Cette méthode est désignée par le sigle RGB (Red, Green, Blue). Les éléments de ces trois matrices sont encore des entiers de 0 à 255. Dans le système RGB, il y a donc : Teintes possibles pour chaque pixel. L'image finale est obtenue en combinant les trois images. Ci-dessous, A Est le résultat final.

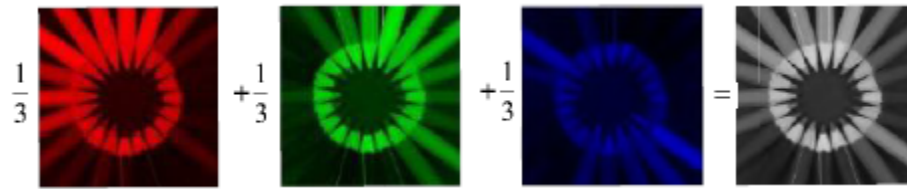


### Traitement d'images numériques et opérations sur les matrices

À tout traitement effectué sur une image numérique va correspondre une transformation sur la matrice qui la représente, et vice-versa. Voici un exemple élémentaire. À l'image carrée A ci-dessous est associée la matrice  $A = (a_{i,j})$ . L'image suivante B est alors associée à  $B = (b_{i,j}) = (a_{j,i}) = A^t$  , c'est-à-dire la transposée de A, obtenue en échangeant les coefficients symétriques par rapport à la diagonale. L'image H correspond à la matrice  $(a_{j,35-i+1})$ . Et comme chaque image correspond à une matrice



Un deuxième exemple. Pour passer une image colorée en noir et blanc, il suffit de former la matrice moyenne arithmétique des trois matrices R, G, B :



Il se pose ici un petit problème d'arrondi, car l'élément de la matrice moyenne est un entier divisé par 3. On choisira l'entier le plus proche. Troisième exemple, les effets de transition. Pour donner l'illusion optique d'une image se transformant progressivement en une nouvelle image (ou d'une image qui disparaît), on peut utiliser une famille de matrices. Si  $A$  est l'image de départ et  $Z$  celle d'arrivée, on va définir pour tout réel  $t$ ,  $t \in [0,1]$  la matrice  $M(t) = (1 - t)A + tZ$ . On remarque que  $M(0) = A$ ,  $M(1) = Z$ , et que quand  $t$  varie, chaque élément de  $M(t)$  varie linéairement de la valeur qu'il a dans  $A$  à celle qu'il a dans  $Z$ . Pour des images en couleurs, on procédera de même pour chacune des trois matrices. Là encore, à chaque étape il faudra effectuer des arrondis. Ce procédé est utilisé dans tous les logiciels de création visuelle, diaporamas, films, présentation de diapos, etc.

## 2.2 Détection des visages

### 2.2.1 Définition

La détection de visage fait partie de la détection d'objet qui est l'une des technologies informatiques, qui est connectée à la vision par ordinateur et interagit avec la détection de visages dans l'image numérique ou la vidéo, qui peut être utilisée dans divers domaines d'intérêt (tels que la biométrie, reconnaissance faciale, divertissement, etc.).

### 2.2.2 Classification des méthodes de détection de visage

**Yan, Kriegman et Ahuja** ont présenté une classification des méthodes de détection des visages. Ces méthodes se divisent en quatre catégories et les algorithmes de détection de visage peuvent appartenir à deux groupes ou plus. Ces catégories sont les suivantes :

- Feature-based
- Appearance-based
- Knowledge-based
- Template matching

Dans notre proposition nous avons s'adapter à la méthode de l'apparence, Car elle dépend d'un ensemble d'images de visage pour découvrir les modèles de visage. Cette méthode fonctionne mieux par rapport aux méthodes restantes car elle tente de localiser / détecter le visage avec l'extraction de caractéristiques.

### 2.2.3 Algorithme de MTCNN

L'algorithme le plus largement utilisé dans la détection des visages est MTCNN (abréviation de Multi-task Cascaded Convolutional Networks). L'algorithme MTCNN est une méthode de détection et d'alignement des visages basée sur l'apprentissage profond. Il peut effectuer les tâches de détection et d'alignement des visages en même temps. Comparé aux algorithmes traditionnels, il offre de meilleures performances et une vitesse de détection plus rapide.

#### Les étapes d'algorithme MTCNN

- **Pyramide des images**
- **P-Net (réseau de proposition)**
- **R-Net (Affiner le réseau)**
- **O-Net (réseau de sortie)**

#### Inconvénients majeur de MTCNN

- plus lent que la méthode en cascade HOG
- Nécessite du matériel haute performance

### 2.2.4 algorithme HOG

#### Description générale

Un histogramme de gradient orienté (en anglais, histogram of oriented gradients ou HOG) est une caractéristique utilisée en vision par ordinateur pour la détection d'objet. La technique calcule des histogrammes locaux de l'orientation du gradient sur une grille dense, c'est-à-dire sur des zones régulièrement réparties sur l'image. Elle possède des points communs avec les SIFT, les Shape contexts et les histogrammes d'orientation de contours, mais en diffère notamment par l'utilisation d'une grille dense. La méthode est particulièrement efficace pour la détection de personnes. Les HOG ont été proposés par Navneet Dalal et Bill Triggs, chercheurs à l'INRIA de Grenoble, à la conférence CVPR de juin 2005.

#### Calcul du gradient

Le calcul du gradient est une étape critique dans la formation de descripteurs. La justesse des orientations calculées, et des histogrammes, dépend de cette étape et les résultats sont donc étroitement liés à la méthode employée pour calculer le gradient dans l'image. Le calcul rapide du gradient peut être fait, par exemple, par des masques de dérivation simple 1-D (centré  $[-1, 0, 1]$  et non centré  $[-1, 1]$ ), par les opérateurs 2-D de Sobel, ou encore par les opérateurs récursifs de Deriche. Dans notre méthode, nous utilisons l'algorithme de Sobel. Il s'agit de l'un des opérateurs les plus simples qui donne toutefois des résultats corrects. L'opérateur utilise des matrices de convolution. La matrice de taille  $3 \times 3$  subit une convolution avec l'image pour calculer des approximations des dérivées horizontale et verticale. Soit  $I$  l'image source,  $G_x$  et  $G_y$  deux images qui en chaque point contiennent des approximations respectivement de la dérivée horizontale et verticale de chaque point. Ces images sont calculées comme suit :

$$G_x(r, c) = I(r, c + 1) - I(r, c - 1) \quad G_y(r, c) = I(r - 1, c) - I(r + 1, c)$$

En chaque point, les approximations des gradients horizontaux et verticaux sont combinées comme suit pour obtenir une q approximation de la norme du gradient :

$$G = \sqrt{G_x^2 + G_y^2}$$

On calcule également la direction du gradient comme suit :

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

## Calcul des histogrammes d'orientation de gradient

Un histogramme est un tableau de nombres dans lequel chaque élément correspond à la fréquence d'apparition d'un intervalle de valeurs pour un ensemble donnée. Dans le cadre d'une image, par exemple, chaque case de l'histogramme peut représenter les pixels de la même couleur. Un histogramme est donc une transformation de l'espace des données vers les nombres réels positifs. D'un point de vue statistique, un histogramme fournit la distribution d'un certain type de données sur un ensemble. L'image est découpée en cellules de tailles 8 x 8 pixels, et pour chaque cellule on calcule l'histogramme d'orientation de gradient. Chaque pixel des cellules participe au vote. Celui-ci est pondéré par la norme du gradient à l'emplacement du pixel.

On crée 9 plans d'images (9 parce qu'on a choisi la taille d'une case d'histogramme égale à 20 degrés et gradient non signée ( $180/20 = 9$ )), qu'on appellera images bin, dont tous les pixels seront mises à zéro sauf les pixels dans l'image originale pour laquelle les valeurs de l'orientation de gradient correspondent à la case particulière. Pour chaque image bin, on calcule l'image intégrale. Ces 9 images intégrales constituent l'histogramme intégral. Pour chaque cellule de taille 8 x 8 pixels, on calcule les valeurs des 9 cases d'histogramme d'orientation de gradient en utilisant les 4 points qui délimitent la cellule correspondante dans l'histogramme intégral. Ceci exige 36 opérations d'accès ( $4 \times 9$ ) à l'image au lieu de 64 pour le descripteur HoG standard, ce qui permet l'accélération de traitement. Pour un bloc de 2 x 2 cellules, on calcule le descripteur HoG Intégral pour chaque cellule, on assemble les vecteurs obtenus en un seul vecteur 1-D de 36 composantes et on le normalise suivant la norme L1. Ce vecteur normalisé correspond au descripteur HOG Intégral pour un bloc.

## 2.3 Linéaire Machine à vecteur de support

### 2.3.1 Définition

Les "Support Vector Machines" ou "Séparateurs à Vaste Marge" (SVM) sont un ensemble de techniques d'apprentissage supervisée destinés à résoudre les problèmes de classification et de régression. Initiée par Vladimir Vapnik comme méthode de classification binaire qui cherche le meilleur hyperplan qui sépare linéairement les positifs et les négatifs en garantissant que la marge entre le deux soit maximal. Ils sont ensuite développés par l'introduction des fonctions dites les noyaux (kernel) pour régler la non linéarité des modèles.

## 2.3.2 Fonctionnement des machines à vecteur support

### Principe

Pour deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Le classificateur linéaire, dans SVM, appelé hyperplan. Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points. Les points les plus proches entre les deux classes sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support. Il est évident qu'il existe une multitude d'hyperplan valide mais la propriété remarquable des SVM est que cet hyperplan doit être optimal. Nous allons donc en plus chercher parmi les hyperplans valides, celui qui passe « au milieu » des points des deux classes d'exemples. Intuitivement, cela revient à chercher l'hyperplan le « plus sûr ». Il est possible d'exprimer le modèle de décision par une fonction linéaire en  $X$  donnée par

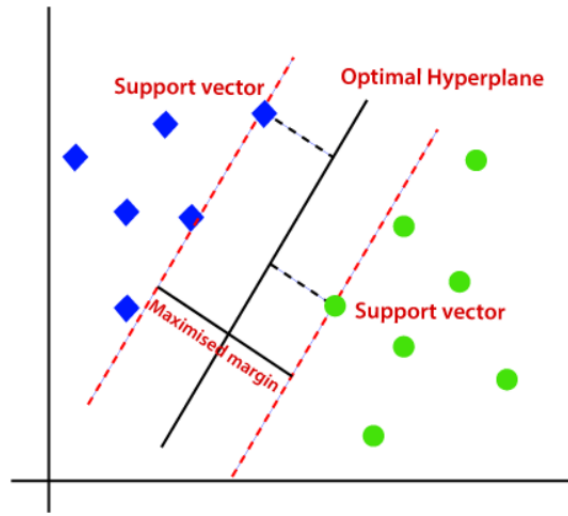


FIGURE 2.2 – Exemple de svm

$$Y = W^T \times X + b$$

Où  $W$  et  $b$  sont des paramètres de l'hyperplan par :  $X \in \mathbb{R}^n$  et  $b \in \mathbb{R}$ .

**Remarque :**

Pour le cas de dimension 2, l'hyperplan est une droite et dans le cas de dimension 3, l'hyperplan est un plan.

### Algorithme de SVM

Soit les deux relation suivant :

$$\max_{\alpha} \left\{ l_p(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j Y_i Y_j X_i^T X_j \right\}. \quad (1)$$

avec

$$\begin{cases} \alpha_i \geq 0, \forall i = \overbrace{1, \dots, n} \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (2)$$

Donc l'algorithme SVM donner comme suit :

- Donner  $Y_i$  et  $X_i$  Pour  $i = \overbrace{1, \dots, n}$
- Trouver  $\alpha$  à partir de l'équation (1) et (2)
- Calculer  $W = \sum_{i=1}^n (\alpha_i Y_i X_i)$
- Calculer  $b = \left( \frac{1 - Y_i W^T \cdot X_i}{Y_i} \right)$

### 2.3.3 Avantages

- Sa grande précision de prédiction.
- Fonctionne bien sûr des petits base de données.
- Ils peuvent être plus efficaces car ils utilisent un sous ensemble de points d'entraînement.

### 2.3.4 Inconvénients

- Ne convient pas à des bases de données plus volumineux, car le temps d'entraînement avec le SVM peut être long.
- Moins efficace sur des bases de données contenant du bruits et beaucoup des anomalies .

### 2.3.5 Les domaines d'applications

- Détection des visages
- Catégorisation du texte et de l'hypertexte
- Classification des images
- Reconnaissance de l'écriture manuscrite

## 2.4 Comparaison des visages

L'approche la plus simple de la comparaison faciale consiste à extraire quelques mesures de base de chaque visage. Ensuite, nous pourrions mesurer notre visage inconnu de la même manière et trouver le visage connu avec les mesures les plus proches. Par exemple, nous pourrions mesurer la taille de chaque oreille, l'espacement entre les yeux, la longueur du nez, etc.

Il s'avère que les mesures qui semblent évidentes pour nous les humains (comme la couleur des yeux) n'ont pas vraiment de sens pour un ordinateur qui regarde des pixels individuels dans une image. Les chercheurs ont découvert que l'approche la plus précise consiste à laisser l'ordinateur déterminer les mesures à collecter lui-même. L'apprentissage en profondeur fait un meilleur travail que les humains pour déterminer quelles parties d'un visage sont importantes à mesurer.

La solution consiste à former un réseau de neurones convolution profonde .nous allons l'entraîner à générer 128 mesures pour chaque visage

Le processus de formation fonctionne en regardant 3 images de visage à la fois :

- Charger une image de visage d'entraînement d'une personne connue



- Charger une autre photo de la même personne connue
- Charger une photo d'une personne totalement différente

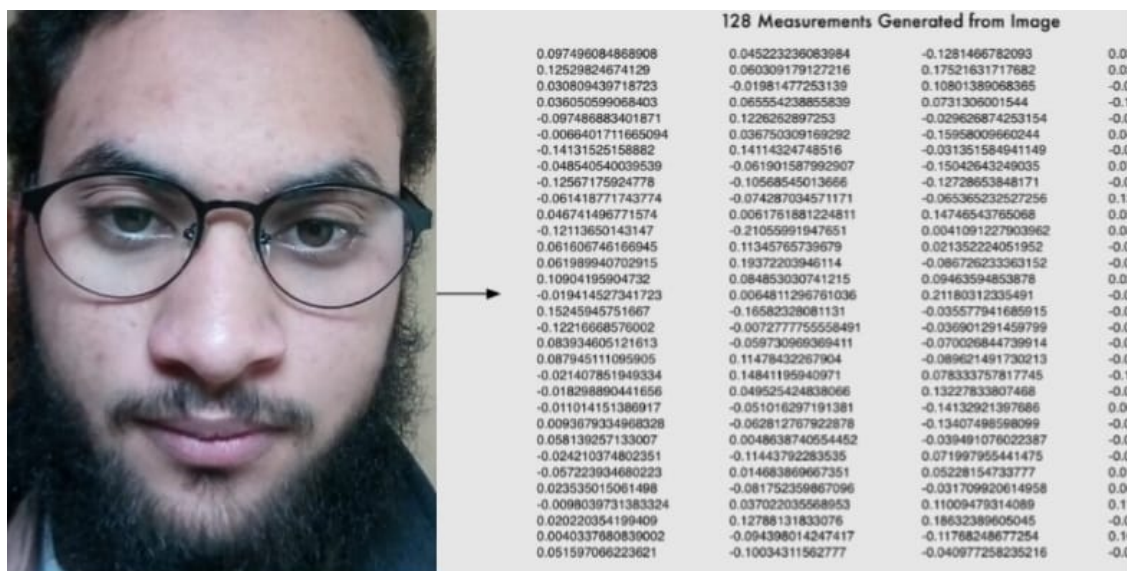
Après avoir répété cette étape des millions de fois pour des millions d'images de milliers de personnes différentes, le réseau de neurones apprend à générer de manière fiable 128 mesures pour chaque personne. Toutes les dix images différentes de la même personne doivent donner à peu près les mêmes mesures.

Les spécialistes de l'apprentissage automatique appellent les 128 mesures de chaque face une intégration. L'idée de réduire des données brutes compliquées comme une image en une liste de nombres générés par ordinateur revient souvent dans l'apprentissage automatique (en particulier dans la traduction linguistique). L'approche exacte pour les visages que nous utilisons a été inventée en 2015 par des chercheurs de Google, mais de nombreuses approches similaires existent

Ce processus de formation d'un réseau neuronal convolutif pour produire des incorporations de visage nécessite beaucoup de données et de puissance informatique. Même avec une carte vidéo NVidia Telsa coûteuse, il faut environ 24 heures de formation continue pour obtenir une bonne précision.

Mais une fois le réseau formé, il peut générer des mesures pour n'importe quel visage, même ceux qu'il n'a jamais vus auparavant ! Cette étape n'est donc à faire qu'une seule fois. Heureusement pour nous,

Donc, tout ce que nous devons faire nous-mêmes est de faire passer nos images de visage à travers leur réseau pré-formé pour obtenir les 128 mesures pour chaque visage.



Figure[3] les measurement générer depuis l'image

et en fin . Tout ce que nous avons à faire est de comparer les mensurations des images. nous pouvons le faire en utilisant n'importe quel algorithme de classification d'apprentissage automatique de base. Aucune astuce sophistiquée d'apprentissage en profondeur n'est nécessaire. Nous utiliserons un classificateur SVM linéaire simple, mais de nombreux algorithmes de classification pourraient fonctionner. L'exécution de ce classificateur prend quelques millisecondes.

# Chapitre 3

## Conception

### 3.0.1 Diagramme de cas d'utilisation

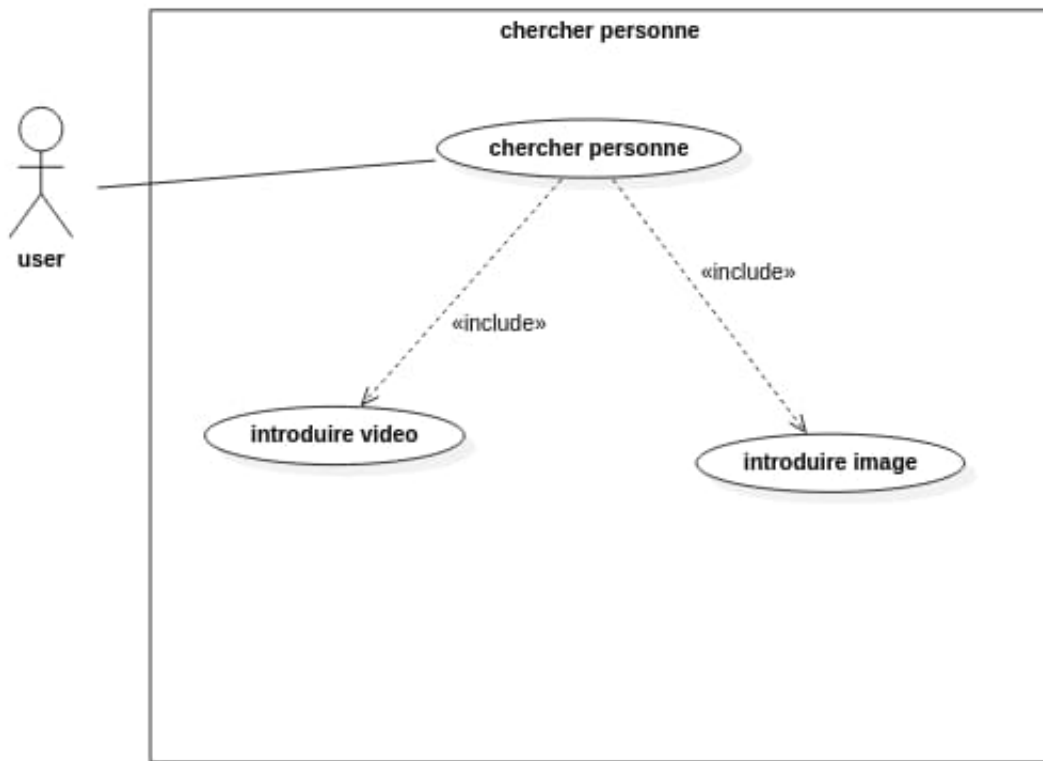


FIGURE 3.1 – Diagramme cas d'utilisation

### 3.0.2 Diagramme de Séquence

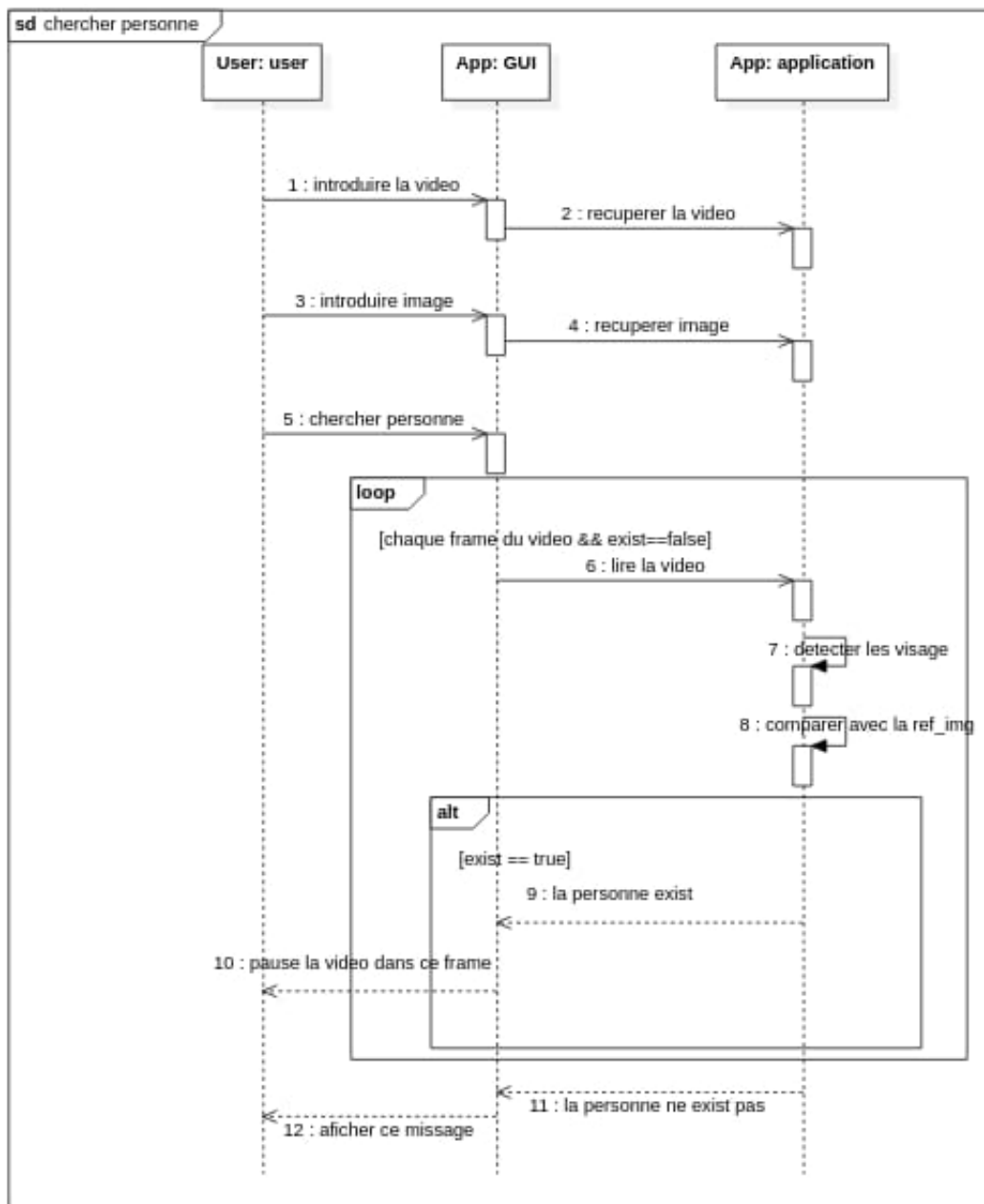


FIGURE 3.2 – Diagramme de Séquence

### 3.0.3 Diagramme de Classe

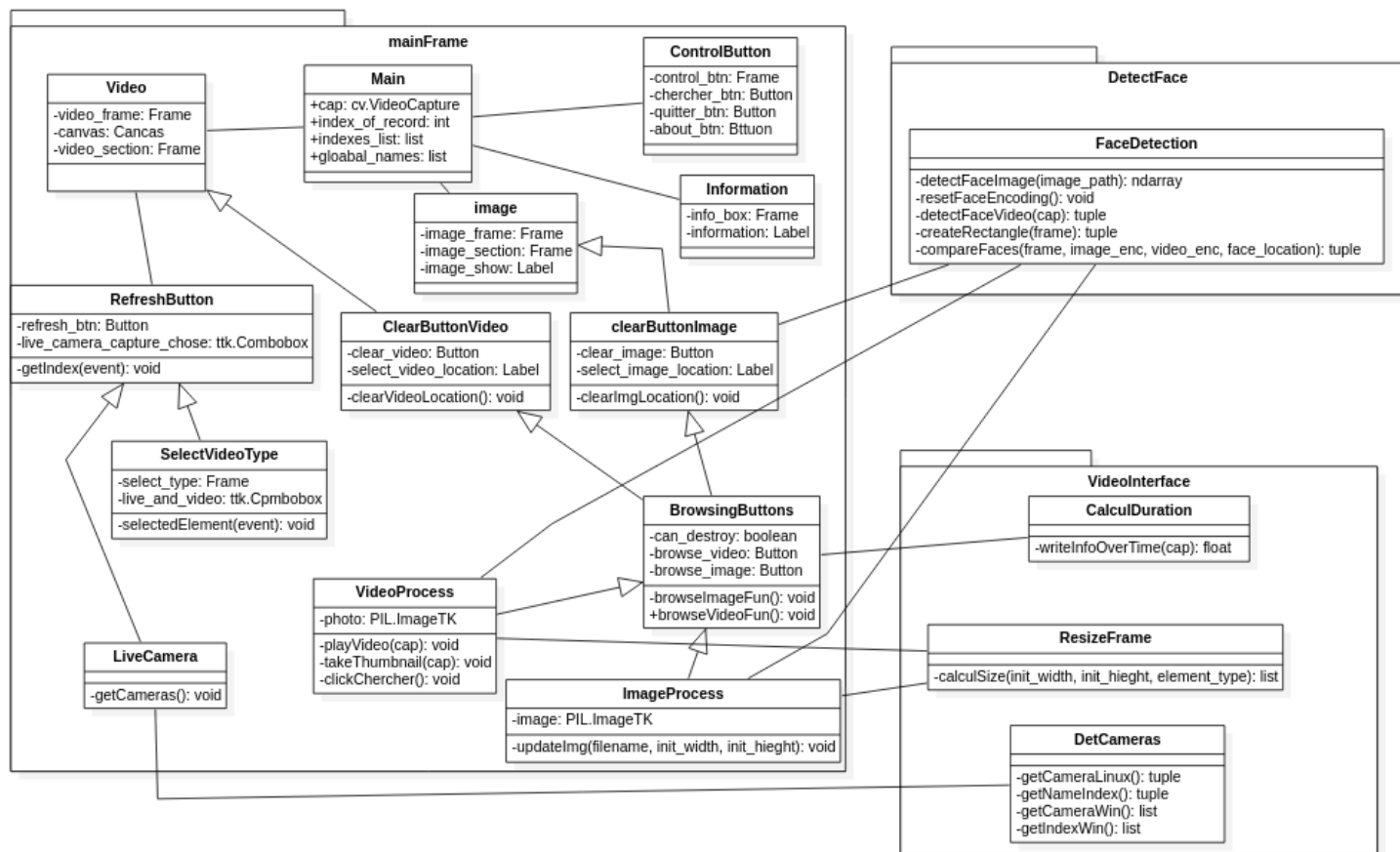


FIGURE 3.3 – Diagramme de Class

# Chapitre 4

## Implémentation

### 4.0.1 Logiciels et librairies Utilisés dans l'implémentation :



FIGURE 4.1 – Python

Le langage Python est un langage de programmation open source, multi-plateformes, haut niveau, interprété et orienté objet. Grâce à des bibliothèques spécialisées, Python s'utilise pour de nombreuses situations comme le développement logiciel, l'analyse de données, ou la gestion d'infrastructures.



FIGURE 4.2 – OpenCv

Initialement développée par Intel, OpenCV (Open Computer Vision) est une bibliothèque graphique. Elle est spécialisée dans le traitement d'images, que ce soit pour de la photo ou de la vidéo. Sa première version est sortie en juin 2000. Elle est disponible sur la plupart des systèmes d'exploitation et existe pour les langages Python, C++ et Java.



FIGURE 4.3 – Pillow

Pillow est une bibliothèque de traitement d'image, qui est un fork et successeur du projet PIL (Python Imaging Library). Elle est conçue de manière à offrir un accès rapide aux données contenues dans une image, et offre un support pour différents formats de fichiers tels que PPM, PNG, JPEG, GIF, TIFF et BMP. Tkinter (de l'anglais Tool kit interface) est la bibliothèque

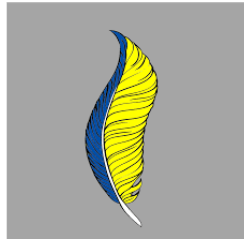


FIGURE 4.4 – Tkinter

graphique libre d'origine pour le langage Python, permettant la création d'interfaces graphiques. Elle vient d'une adaptation de la bibliothèque graphique Tk écrite pour Tcl.



FIGURE 4.5 – Numpy

NumPy est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux. Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.



FIGURE 4.6 – PyCharm

PyCharm est un environnement de développement intégré utilisé pour programmer en Python. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django.

Développé par l'entreprise tchèque JetBrains, c'est un logiciel multi-plateforme qui fonctionne sous Windows, Mac OS X et GNU/Linux. Visual Studio Code est un éditeur de code open-source



FIGURE 4.7 – Visual Studio Code

développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'autocomplétion, la coloration syntaxique, le débogage, et les commandes git.

## 4.0.2 Présentation De L'interface Graphique :

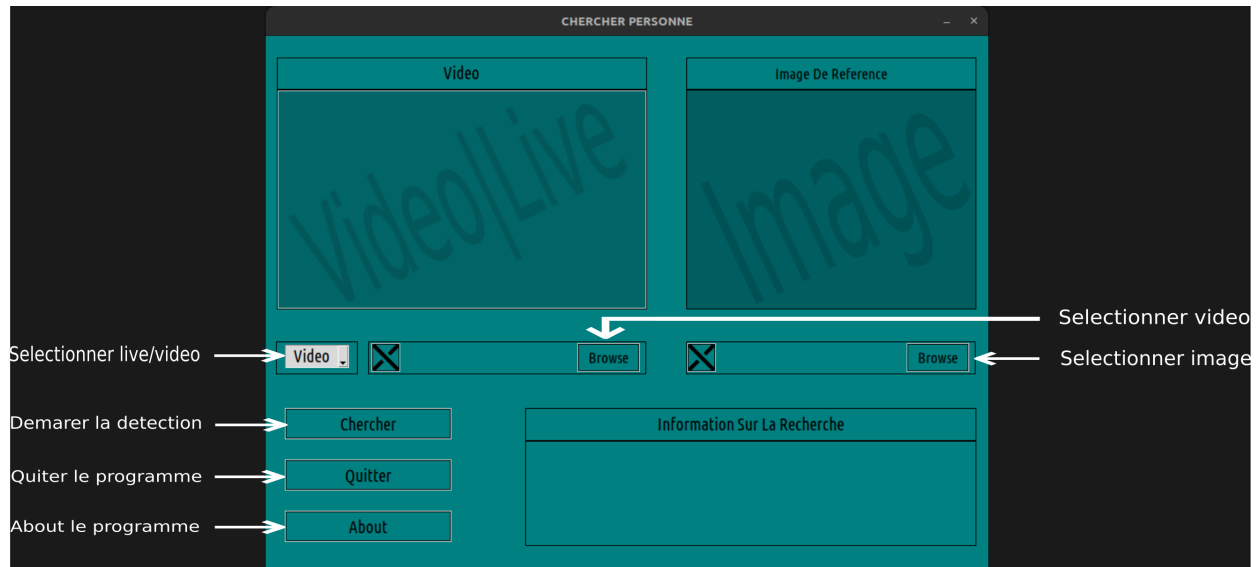


FIGURE 4.8 –

## 4.1 conclusion

Les systèmes de reconnaissance du visage utilisés aujourd'hui fonctionnent mais dans des conditions limitées, bien que tous les systèmes fonctionnent beaucoup mieux avec des images de face et de profil et un éclairage constant. Tous les algorithmes actuels de reconnaissance de visage ne parviennent pas à identifier des visages dans les conditions largement variables en vertu de laquelle les humains sont capables d'identifier d'autres personnes. Les systèmes de reconnaissance de personne de la prochaine génération devront reconnaître les gens en temps réel et dans des situations beaucoup moins contraintes. Nous croyons que les systèmes d'identification qui sont robustes dans les milieux naturels, en présence de changements de bruit et d'éclairage, ne pourront pas compter sur une seule modalité, ce qui signifie que la fusion avec d'autres modalités sera essentielle. Compte tenu de toutes les exigences, les systèmes d'identification qui utilisent la reconnaissance faciale et l'identification vocale nous semblent avoir le plus de potentiel pour des applications très répandues.



# Bibliographie

- [1] <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ipr.2019.0141>
- [2] <https://ichi.pro/fr/pipeline-de-reconnaissance-faciale-clairement-explique-269929031411536>
- [3] <https://ichi.pro/fr/mtcnn-reseau-de-convolution-en-cascade-multi-taches-186543706018893>
- [4] [https://face-recognition.readthedocs.io/en/latest/face\\_recognition.html](https://face-recognition.readthedocs.io/en/latest/face_recognition.html)
- [5] Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems
- [6] <https://www.talend.com/fr/resources/what-is-machine-learning/>
- [7] <https://datascientest.com/machine-learning-tout-savoir>
- [8] <https://www.techno-science.net/glossaire-definition/Machine-a-vecteurs-de-support.html>
- [9] Alan Turing, « Computing machinery and intelligence », Mind (en), Oxford University Press, vol. 59, no 236, octobre 1950, p. 433-460

24 septembre 2023