

الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université –Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et de Technologie
Département D'électronique et des Télécommunications



Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master Académique en : Electronique
Domaine : Sciences et Technologie
Filière : Electronique
Spécialité : Instrumentation
Thème

**Conception et Réalisation d'un Système Intelligent de Surveillance
basée sur Raspberry Pi dans le Contexte du Covid-19**

Présenté Par :

Mr CHERRAK Abdenasser
Mr BOUKLI HACENE Adel

Devant le jury composé de :

Mme BOUTKHIL Malika	MAA	UATBB	Présidente
Dr BEMMOUSSAT Chems Eddine	MCB	UATBB	Examineur
Dr BENZINA Amina	MCB	UATBB	Invitée
Dr SEKKAL Mansouria	MCB	UATBB	Invitée
Dr BENTAIEB Samia	MCB	UATBB	Encadrante

Année Universitaire 2021/2022

*C'est avec une grande émotion et un grand plaisir que je dédie ce modeste travail.
A la mémoire de ma grand-mère qui nous a quittés cette année, que Dieu lui accorde
sa miséricorde et son paradis.*

*A mon père qui m'a soutenu et me donne tout ce dont j'ai besoin pour être là, que
Dieu le guérisse et le garde.*

*A ma chère maman pour son soutien, sa patience, ses sacrifices et son amour, qu'AL-
LAH te garde et te bénisse.*

A mes frères et sœurs et toute ma famille.

A tous mes enseignants tout au long de ma scolarité.

*A tous mes chers amis, à mon compagnon Adel et à tous ceux qui m'aiment et que
j'aime.*

Nasser

Je dédie ce modeste travail

A mes chers parents : Quoi que je fasse ou quoi que je dise, je ne saurai vous remercier comme il se doit. Votre affection me couvre, votre bienveillance me guide et votre présence a toujours été ma source de force pour affronter les différents obstacles.

A mes chers frères et belles sœurs : Puisse DIEU vous donner santé, bonheur et réussite dans tous vos projets.

A la mémoire de mes grands-parents décédés.

A ma grand-mère à qui je souhaite une longue vie.

A mes enseignants et camarades spécialement Nasser.

Adel

Remerciements

Nous remercions Dieu tout puissant de nous avoir comblés de ses faveurs.

Nous souhaitons rendre hommage et exprimer notre profonde gratitude à tous ceux qui, de près ou loin, ont contribué à la réalisation de ce mémoire et à son aboutissement.

Nous tenons à exprimer notre profond respect et notre reconnaissance à notre Dr BENTAIEB Samia pour sa confiance, sa disponibilité et le grand intérêt qu'elle nous a toujours manifestés durant notre formation et la préparation de ce mémoire. Nous tenons aussi à exprimer nos remerciements à Mme BOUTKHIL Malika qui a accepté de présider notre jury, tout comme Dr BEMMOUSSAT Chems Eddine d'avoir accepté d'examiner et de juger notre travail. Nous ne pouvons passer sous silence l'aide et le soutien de Dr BENZINA Amina et Dr SEKKAL Mansouria qui nous ont honorés de leur présence en tant qu'invitées.

Nous renouvelons toute notre amitié et notre sympathie à ceux qui nous ont accordés du temps et nous ont témoigné un soutien constant dans ce travail de recherche.

Table des matières

Table des figures	vii
Liste des tableaux	viii
Glossaire	x
I Partie Théorique	6
1 Généralité sur le Traitement d'Images	7
1.1 Introduction	8
1.2 Définitions	8
1.3 Caractéristiques d'une image numérique	9
1.3.1 Le type d'image	9
1.3.1.1 Les images binaires	9
1.3.1.2 Les images monochromes	10
1.3.1.3 Les images polychromes	10
1.3.2 La résolution d'une image	10
1.3.3 Les formats d'images	11
1.3.4 TIFF	11
1.3.4.1 JPEG	11
1.3.4.2 GIF	12
1.3.4.3 PNG	12
1.4 Le prétraitement des images	12
1.4.1 Élimination du bruit	13

1.4.2	Recadrage d'une image	14
1.4.3	Redimensionnement d'une image	14
1.5	La vidéo	16
1.5.1	Définition	16
1.5.2	Nombre d'images par seconde	16
1.5.3	La résolution	17
1.5.4	Domaine d'utilisation de la vidéo	18
1.6	Conclusion	18
2	Intelligence Artificielle	19
2.1	Introduction	20
2.2	L'intelligence Artificielle (IA)	20
2.3	Machine Learning (ML)	21
2.3.1	L'apprentissage supervisé	22
2.3.2	L'apprentissage non-supervisé	22
2.3.3	L'apprentissage semi-supervisé	22
2.4	Les Principaux Algorithmes du ML	22
2.4.1	K-Nearest Neighbor (K-NN)	22
2.4.2	Support Vector Machine (SVM)	23
2.5	Deep Learning (DL)	23
2.5.1	Cerveau humain et réseaux neuronaux artificiels	24
2.5.2	Les fonctions d'activation	26
2.5.3	Convolutional Neural Network	27
2.5.3.1	La couche de convolution	27
2.5.3.2	La couche de pooling	28
2.5.3.3	La couche flatten	29
2.5.4	Quelques architectures de Deep Learning	32
2.5.4.1	LeNet	32
2.5.4.2	AlexNet	32
2.5.4.3	VGG	32
2.5.4.4	ResNet	33
2.5.4.5	MobilNet	33
2.6	Conclusion	33

3	La Vidéo-surveillance dans le Contexte du COVID-19 par Deep Learning :	34
3.1	Introduction	35
3.2	Contexte et études connexes	35
3.2.1	Détection de visages par la couleur de peau	35
3.2.2	Détection de visages à l'aide de Eigenface	36
3.2.3	Détection robuste des visages en utilisant la distance de Hausdorff	37
3.2.4	Le détecteur de visage Viola-Jones	38
3.3	Détection de visages et de masques	38
3.3.1	Détection de personnes	38
3.3.2	Single Shot MultiBox Detector (SSD)	40
3.3.3	Détection de masques	42
3.3.3.1	Le CNN pour la détection	42
3.3.3.2	Environnement de travail	43
3.3.3.3	Base de données et protocole d'évaluation	43
3.3.3.4	Résultat obtenus	46
3.3.4	Mesure de distanciation	47
3.4	Conclusion	48
II	Partie Pratique	49
4	Environnement de développement	50
4.1	Introduction	51
4.2	Environnement hardware	51
4.2.1	Présentation du Raspberry pi	51
4.2.1.1	Historique	51
4.2.1.2	Composant de base	52
4.2.1.3	Spécifications	54
4.2.2	Ecran TFT LCD 7	55
4.2.3	Une caméra Pi	56
4.2.3.1	Spécifications	56
4.2.4	Carte Micro SD	57
4.3	Environnement software	58
4.3.1	Raspbian	58

4.3.2	Langage Python	58
4.3.3	Bibliothèques utilisées	59
4.3.3.1	Math	59
4.3.3.2	Numpy	59
4.3.3.3	Mathplotlib.pyplot	60
4.3.3.4	OpenCV	60
4.3.3.5	TensorFlow	62
4.3.3.6	Keras	62
4.3.3.7	Time	63
4.3.4	L'IDE Thonny	63
4.3.5	VNC Viewer	64
4.4	Conclusion	64
5	Implémentation su Raspberry Pi 4	65
5.1	Introduction	66
5.2	Configuration matérielle	66
5.2.1	Téléchargement de l'image de distribution	66
5.2.2	Enregistrement du contenu de l'image sur la carte SD	67
5.2.3	Configuration initiale	67
5.3	Conception du système	70
5.3.1	Détection de visage	71
5.3.2	Détection de masque	72
5.3.3	Mesure de distanciation	73
5.4	Conclusion	73
	Bibliographie	77

Table des figures

1.1	Les pixels d'une image	8
1.2	Le système de coordonnées utilisé en traitement d'images	9
1.3	Les différents types d'image	10
1.4	Une image avec différentes résolutions	11
1.5	Une image avec différents types de bruit	13
1.6	Exemple d'une image recadrée	14
1.7	Une image redimensionnée avec différents types d'interpolationt	15
1.8	Illustration de la notion de FPS	17
1.9	Comparatif des différentes résolutions	17
2.1	Les application de l'IA	21
2.2	Marge optimale de l'hyperplan de séparation linéaire	23
2.3	Illustration de réseau de neurones	24
2.4	Architecture d'un neurone biologique	25
2.5	Architecture d'un neurone artificiel (perceptron)	25
2.6	fonction d'activation	27
2.7	La couche de convolution	28
2.8	Illustration des différents types de pooling	29
2.9	Illustration de la couche flatten	30
2.10	La couche dropout layer	30
2.11	Exemple d'une couche entièrement connectée	31
3.1	Détection du visage en utilisant la couleur de la peau détection du visage en utilisant la couleur de la peau [Kovac et al., 2003]	36
3.2	Détection de visages en utilisant la distance de Hausdorff	37

3.3	Diagramme de la méthode utilisée	39
3.4	Détection de visage dans une image	40
3.5	Architecture du réseau SSD 300	41
3.6	Architecture du modèle de détection de masques	42
3.7	Colaboratory (Exemple)	44
3.8	Exemple de visages avec et sans masque	44
3.9	La matrice de confusion	46
3.10	Le taux d'apprentissage et de l'erreur en fonction du nombre d'époque	47
3.11	Distance Sociale	48
4.1	Evolution du Raspberry Pi	52
4.2	Raspberry Pi 4 model B	52
4.3	Raspberry Pi 4 model B	55
4.4	Ecran TFT LCD 7" pour Raspberry	56
4.5	Raspberry Pi Camera Module 2	57
4.6	Logo du langage Python	58
4.7	Types de tracés utilisé par Matplotlib	60
4.8	Logo de la bibliotheque OpenCV	61
5.1	Modification du nom d'hôte	68
5.2	Activation du SSH	68
5.3	Activation du VNC	69
5.4	Activation de le caméra	70
5.5	Prototype de vidéo-surveillance proposé	70
5.6	Vue de près du prototype de vidéo-surveillance proposé	71
5.7	Code de détection du visage et du masque	71
5.8	Quelques résultats de la détection du visage	72
5.9	Quelques résultats de la détection du masque	74
5.10	Quelques résultats de la mesure de distanciation	75

Liste des tableaux

3.1	Nombre de boites pour le réseau SSD 300	41
3.2	Matrice de confusion d'un problème à deux classes	46
4.1	Comparaison entre différents modèles de Raspberry	53

Listings

3.1 Répartition de la base de données	43
3.2 Hyper paramètres et compilation du modèle	45

Abréviations

AdaBoost : **A**daptive **B**oosting
AI : **A**rtificial **I**ntelligence
CMYK : **C**yan **M**agenta **Y**ellow **blacK**
CNN : **C**onvolution **N**eural **N**etwork
CPU : **C**entral **P**rocessing **U**nit
CSI : **C**amera **S**erial **I**terface
DL : **D**eep **L**earning
DPI : **D**ote **P**er **I**nh
FCN : **F**ully **C**onected **N**etwork
FPS : **F**rames **P**er **S**econd
GIF : **G**aphic **I**terchange **F**ormat.
GPIO : **G**eneral **P**urpose **I**nput/**O**utput
GPU : **G**raphics **P**rocessing **U**nit
HDMI : **H**igh **M**ultimedia **M**ultimedia **I**nterface
IA : **I**ntelligence **A**rtificielle
IDE : **I**ntegrated **D**evelopment **E**nvironment
ISO : **I**nternational **O**rganization for **S**tandardization
JPEG : **J**oint **P**hotographic **E**xperts **G**roup
K-NN : **K**-**N**earest **N**eighbor
LCD : **L**iquid **C**rystal **D**isplay
ML : **M**achine **L**earning
NB : **N**aive **B**ayes
OCR : **O**ptical **C**haracter **R**ecognition
OMS : **O**rganisation **M**ondiale de la **S**anté

OpenCV : **O**pen **S**ource **C**omputer **V**ision
OS : **O**perating **S**ystem
PCA : **P**ripcipal en **C**omponent **A**nalysis
PNG : **P**ortable **N**etwork **G**raphics
ReLU : **R**ectified **L**inear **U**nits
RGB : **R**ed **G**een **B**lue
RMSProp : **R**oot **M**ean **P**Square **P**ropagation
SBC : **S**ingle **B**oard **C**omputer
SD : **S**ecure **D**igital
SGD : **S**tochastic **G**radient **D**escent
SSH : **S**ecure **S**Hell
SVM : **S**upport **V**ector **M**achine
TFT : **T**hin **F**ilm **T**ransistor
TIFF : **T**agged **I**mage **F**ile **F**ormat
VGG : **V**isual **G**eometry **G**roup
VNC : **V**irtual **N**etwork **C**omputing
XML : e**X**tensible **M**arkup **L**anguage

Résumé

Le SARS-CoV-2 est l'une des pandémies les plus meurtrières que le monde ait connues. Plus de 6 millions de personnes ont perdu leur vie à ce jour. Il s'agit d'une sorte de pandémie qui s'est développée à un rythme rapide et a provoqué une crise sanitaire. Ainsi, les médecins et les scientifiques ont suggéré des moyens de protection en particulier le port du masque et le respect de la distanciation sociale.

Dans notre travail, la technique de détection des visages basée sur le deep learning est utilisée. Nous proposons un nouveau modèle de détection de masque facial qui est réalisé à l'aide d'un algorithme de classification d'images de vision par ordinateur basé sur le réseau neuronal de l'architecture CNN ainsi les bibliothèques du langage Python. Pour la mesure de distanciation nous utilisons la la distance euclidienne pour le calcul. Toutes ces méthodes ont été implémentées dans le Raspberry Pi 4 et visualisées par une caméra en temps réel.

Mots clés : Vidéo surveillance, détection de masque, détection de visage, mesure de distanciation, Raspberry Pi 4, CNN.

ملخص

يعد $SARS - CoV - 2$ أحد أكثر الأوبئة فتكًا في العالم. لقد فقد أكثر من 6 ملايين شخص حياتهم حتى الآن. هذا نوع من الوباء تطور بوتيرة سريعة وتسبب في أزمة صحية. على سبيل المثال، اقترح الأطباء والعلماء طرقًا لحماية أنفسهم، لا سيما من خلال ارتداء الأقنعة والحفاظ على التباعد الاجتماعي.

في عملنا، يتم استخدام تقنية اكتشاف الوجه القائمة على *deep learning*. نقترح نموذجًا جديدًا للكشف عن أقنعة الوجه يتم تحقيقه باستخدام خوارزمية تصنيف صور رؤية الكمبيوتر بناءً على الشبكة العصبية لبنية *CNN* بالإضافة إلى مكتبات لغة *python*. لقياس المسافة نستخدم المسافة الإقليدية للحسا. تم تن فيذ كل هذه الأساليب في *Raspberry Pi 4* وعرضها بواسطة كاميرا في الوقت الحقيقي.

الكلمات الرئيسية : المراقبة بالفيديو ، كشف القناع ، كشف الوجه ، قياس المسافة ،

CNN ، Python ، RaspberryPi

Abstract

SARS-CoV-2 is one of the deadliest pandemics the world has ever seen. More than 6 million people have lost their lives to date. It is a pandemic that has developed at a rapid pace and has caused a health crisis. Thus, doctors and scientists have suggested means of protection in particular the wearing of masks and the respect of social distancing.

In our work, the face detection technique based on deep learning is used. We propose a new face mask detection model which is realized using a computer vision image classification algorithm based on the neural network of the CNN architecture and the libraries of the Python language. For the measure of distancing, we use the Euclidean distance for the calculation. All these methods have been implemented in the Raspberry Pi 4 and visualized by a camera in real time.

Keywords : Video surveillance, mask detection, face detection, distancing measurement, Raspberry Pi, CNN.

Introduction générale

Contexte

Récemment, la transmission rapide du Coronavirus 2019 (COVID-19) a provoqué une importante crise sanitaire dans le monde entier. L'organisation Mondiale de la Santé (OMS) a publié plusieurs lignes directrices pour se protéger contre la propagation du COVID-19. Selon l'OMS, la mesure préventive la plus efficace contre le COVID-19 est le port d'un masque dans les lieux publics et les endroits bondés. Il est assez difficile de surveiller manuellement et de déterminer les personnes qui portent un masque et celles qui n'en portent pas ou celles qui respectent la mesure de distanciation. Grâce à l'intelligence artificielle et ses différentes architectures d'apprentissage profond de créer des modèles pour une meilleure évaluation des résultats L'objectif final de ce travail est d'utiliser la vidéo-surveillance avec l'aide de l'AI pour faciliter cette tâche.

Motivation

"Mieux vaut prévenir que guérir" est l'une des mesures efficaces pour empêcher la propagation de toute pandémie et protéger l'humanité. Les précautions indiquées par les gouvernements de tous les pays sont la distanciation sociale et le port de masques pour aider à prévenir la propagation de l'infection et empêcher la personne de recevoir des germes infectieux.

Les technologies émergentes telles que l'intelligence artificielle ont été appliquées dans différents domaines pour lutter contre la COVID-19. Les modèles d'e l'IA ont été utilisés par des scientifiques, des chercheurs pour intensifier le processus de dépis-

tage des pandémies, la détection et la cartographie en temps réel des points chauds, l'imagerie thermique, la prévision. etc. Il est donc primordial d'appliquer des modèles d'intelligence artificielle pour détecter les masques faciaux contre la COVID-19. Les modèles d'intelligence artificielle ont obtenu des résultats prometteurs en matière de classification des images et de détection des objets. D'où la nécessité d'appliquer ces modèles à la détection des masques faciaux.

Objectif

L'objectif de notre travail est la détection de masque dans un visage et la mesure la distanciation entre les personnes. Plus précisément, nous devons créer un modèle capable de prédire ou de classer avec précision si une personne porte un masque ou non et si elle respecte la distanciation sociale avec l'aide des techniques de l'intelligence artificielle et de son implémentation dans un Raspberry Pi 4.

Structure

Notre travail se compose de cinq chapitres divisés en deux parties : une partie théorique et l'autre pratique . La partie théorique contient 3 chapitres. Dans le chapitre 1, nous présentons d'abord les concepts de base nécessaires à la compréhension des différentes méthodes de traitement d'images. Le chapitre 2 est une revue des principales techniques de l'intelligence artificielle. Le chapitre 3 : présente la technique proposée pour la vidéo-surveillance, plus précisément la détection des visages, le port du masque et la mesure de distanciation.

La partie pratique se compose de 2 chapitres :

Le chapitre 4 est une présentation de l'environnement de développement basé sur le Raspberry Pi 4.

Dans le chapitre 5 implémentation de le méthodes utilisée sur le Raspberry Pi 4 est présenté et discutée en détails. Finalement, une conclusion générale est présentée.

Première partie
Partie Théorique

Chapitre 1

Généralité sur le Traitement d'Images

Sommaire

1.1	Introduction	8
1.2	Définitions	8
1.3	Caractéristiques d'une image numérique	9
1.3.1	Le type d'image	9
1.3.2	La résolution d'une image	10
1.3.3	Les formats d'images	11
1.3.4	TIFF	11
1.4	Le prétraitement des images	12
1.4.1	Élimination du bruit	13
1.4.2	Recadrage d'une image	14
1.4.3	Redimensionnement d'une image	14
1.5	La vidéo	16
1.5.1	Définition	16
1.5.2	Nombre d'images par seconde	16
1.5.3	La résolution	17
1.5.4	Domaine d'utilisation de la vidéo	18
1.6	Conclusion	18

1.1 Introduction

Le traitement d'images est une méthode qui permet d'effectuer certaines opérations sur une image, afin d'obtenir une image améliorée ou d'en extraire des informations utiles. Il s'agit d'un type de traitement du signal dans lequel l'entrée est une image et la sortie peut être une image ou des caractéristiques associées à cette image. De nos jours, le traitement d'images fait partie des technologies en plein essor. Il constitue un domaine de recherche essentiel dans les disciplines de l'ingénierie et de l'informatique.

1.2 Définitions

Une image statique est une représentation spatiale d'un objet, d'une scène bidimensionnelle ou tridimensionnelle ou d'une autre image. Elle peut être réelle ou virtuelle. Une image peut être considérée comme une fonction continue définissant une région généralement rectangulaire d'un plan.

Une image numérique est une représentation d'une image sous la forme d'un ensemble de nombres qui peuvent être stockés et manipulés par un ordinateur. Afin de traduire l'image en chiffres, elle est divisée en petites zones appelées pixels (éléments d'image). La figure 1.1 montre un exemple de pixels dans une image monochrome.

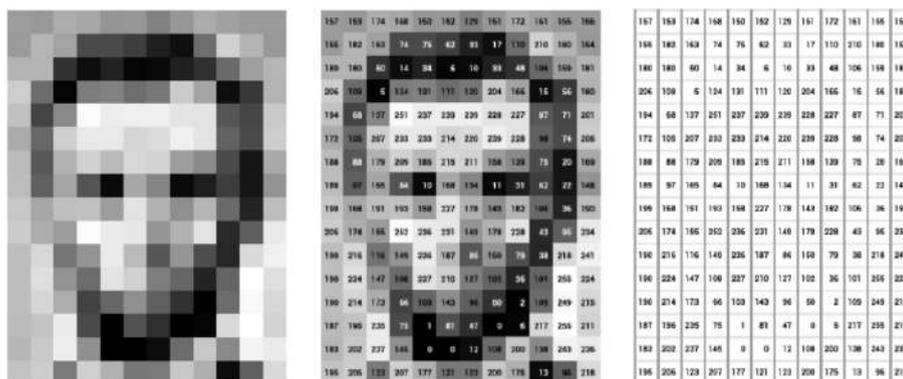


FIGURE 1.1 – Les pixels d'une image

Une image peut être définie comme une fonction bidimensionnelle $I(x, y)$, où x et y sont des coordonnées spatiales et l'amplitude de I à une paire de coordonnées est appelée intensité de l'image en ce point [Thompson, 1981].

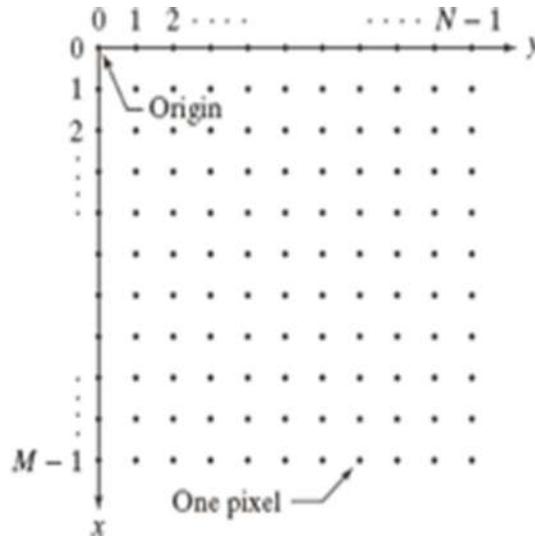


FIGURE 1.2 – Le système de coordonnées utilisé en traitement d’images

Supposons qu’une image I soit de taille $M \times N$, les valeurs des coordonnées étant discrètes. L’origine de l’image est définie comme étant $(x, y) = (0, 0)$. Les valeurs de coordonnées suivantes le long de la première ligne de l’image sont $(x, y) = (0, 1)$. La notation (m, n) est utilisée pour indiquer le $m^{\text{ème}}$ pixel le long de la $n^{\text{ème}}$ ligne. La figure 1.2 illustre cette convention de coordonnées.

1.3 Caractéristiques d’une image numérique

Les images numériques présentent plusieurs caractéristiques de base telles que le type, le format et la résolution.

1.3.1 Le type d’image

1.3.1.1 Les images binaires

Comme leur nom l’indique, elles ne contiennent que deux éléments de pixel, à savoir 0 et 1, où 0 correspond au noir et 1 au blanc [Thompson, 1981].

Les images binaires ne nécessitent qu’un seul chiffre binaire pour représenter chaque pixel, c’est pourquoi elles sont également connues sous le nom d’images à 1 bit.

1.3.1.2 Les images monochromes

Ces images sont également connues sous le nom de Gray Scale Image ou unicolores. Les images en niveaux de gris ne contiennent que des informations sur les niveaux de gris ; elles ne contiennent aucune information sur les couleurs. Le nombre de niveaux de gris différents disponibles est déterminé par le nombre de bits utilisés pour chaque pixel.

Les images en niveaux de gris contiennent 256 teintes de gris, la valeur zéro étant le noir et la valeur 255 le blanc. Le nombre 256 est lié à la quantification de l'image. Chaque entier représentant un niveau de gris est codé sur 8 bits. Il est donc compris entre 0 et $2^8 - 1 = 255$.

1.3.1.3 Les images polychromes

Les images polychromes ou images en couleur sont également connues sous le nom d'image RGB (Red, Green et Blue), car les images couleur sont représentées par le rouge, le vert et le bleu. La figure 1.3 montre un exemple des types d'images mentionnés.



(a) Image binaire (b) Image en niveaux de gris (c) Image couleur

FIGURE 1.3 – Les différents types d'image

1.3.2 La résolution d'une image

C'est le nombre de points contenus dans une longueur donnée (en pouce). Elle est exprimée en Dots Per Inch (DPI) ; un inch mesure 2.54 cm. La résolution permet ainsi d'établir le rapport entre la définition en pixels d'une image et la dimension réelle

de sa représentation sur un support physique (affichage écran, impression papier...). La figure 1.4 montre la même image avec différentes résolutions.



FIGURE 1.4 – Une image avec différentes résolutions

1.3.3 Les formats d'images

Il existe 4 formats principaux pour stocker les images :

1.3.4 TIFF

TIFF est l'abréviation de Tagged Image File Format. Les images TIFF créent des fichiers de très grande taille. Les images TIFF ne sont pas compressées et contiennent donc beaucoup de données d'image détaillées (c'est pourquoi les fichiers sont si volumineux). Les TIFF sont également extrêmement flexibles en termes de couleur (ils peuvent être en niveaux de gris, ou CMYK pour l'impression, ou RGB pour le web).

1.3.4.1 JPEG

JPEG est l'abréviation de Joint Photographic Experts Group, qui a créé cette norme pour ce type de format d'image. Les fichiers JPEG sont des images qui ont été compressées pour stocker beaucoup d'informations dans un fichier de petite taille. La plupart des appareils photo numériques stockent les photos au format JPEG, car il est alors possible de prendre plus de photos sur une carte d'appareil photo qu'avec d'autres formats.

Un JPEG est compressé de manière à perdre une partie des détails de l'image au cours de la compression, afin de réduire la taille du fichier (on parle alors de compression "avec perte").

Les fichiers JPEG sont généralement utilisés pour les photographies sur le Web, car ils créent un petit fichier qui se charge facilement sur une page Web et qui est de bonne qualité.

Les fichiers JPEG sont mauvais pour les dessins au trait, les logos ou les graphiques, car la compression leur donne un aspect "bitmappy" (lignes irrégulières au lieu de lignes droites).

1.3.4.2 GIF

GIF est l'abréviation de Graphic Interchange Format. Ce format compresse les images mais, à la différence du JPEG, la compression est sans perte (aucun détail n'est perdu dans la compression, mais le fichier ne peut pas être rendu aussi petit qu'un JPEG).

Les GIF ont également une gamme de couleurs extrêmement limitée, adaptée au Web mais pas à l'impression. Ce format n'est jamais utilisé pour la photographie, en raison du nombre limité de couleurs. Les GIF peuvent également être utilisés pour les animations.

1.3.4.3 PNG

Le PNG est l'abréviation de Portable Network Graphics. Il a été créé en tant que format ouvert pour remplacer le GIF, car le brevet du GIF était détenu par une société et personne d'autre ne voulait payer les frais de licence. Il permet également une gamme complète de couleurs et une meilleure compression.

Il est utilisé presque exclusivement pour les images Web, jamais pour les images imprimées. Pour les photographies, le PNG n'est pas aussi bon que le JPEG, car il crée un fichier plus volumineux. Mais pour les images contenant du texte ou des dessins au trait, le PNG est meilleur, car les images ont l'air moins "bitmappy".

1.4 Le prétraitement des images

Le prétraitement est utilisé pour supprimer le bruit et éliminer les informations non pertinentes et visuellement inutiles. Le bruit est une information indésirable qui peut résulter du processus d'acquisition de l'image. D'autres étapes de prétraitement peuvent inclure la recherche de régions intéressantes pour un traitement ultérieur.

1.4.1 Élimination du bruit

Le bruit est toute information indésirable qui vient contaminer une image. Il apparaît dans l'image à partir d'une variété de sources. Lors du processus d'acquisition d'images numériques, l'image optique est convertit en un signal électrique continu qui est ensuite échantillonné. Ceci est le principal processus par lequel le bruit apparaît dans les images numériques. D'autres causes telles que le mauvais fonctionnement du capteur, les défauts de la procédure de transmission des données et les interférences électroniques peuvent réduire de manière significative la qualité des images.

Tout ceci conduit à l'ajout une valeur aléatoire à la luminosité exacte d'un pixel donné. Dans une image, le bruit peut être modélisé avec l'une des distributions suivantes :

- Distribution Gaussienne
- Distribution uniforme
- Distribution du sel et du poivre.

La figure 1.5 montre un exemple de bruits d'images mentionnés affectant une image.



(a) Image originale (b) Bruit de distribution Gaussienne (c) Bruit de distribution du sel et du poivre

FIGURE 1.5 – Une image avec différents types de bruit

Afin de supprimer différents types de bruit dans les images numériques et effectuer un certain type d'amélioration de l'image, un filtrage spatial est généralement utilisé. Les trois types de filtres sont :

- Les filtres moyens
- Filtres médians

— Filtres de rehaussement

Les filtres moyens et médians sont principalement utilisés pour masquer ou supprimer le bruit, mais ils peuvent également être utilisés pour des applications spéciales. Par exemple, un filtre moyen donne un aspect plus "doux" à une image. Le filtre de rehaussement fait ressortir les bords et les détails de l'image. Les filtres spatiaux sont implémentés avec des masques de convolution. Comme l'opération du masque de convolution fournit un résultat qui est la somme pondérée des valeurs d'un pixel et de ses voisins, on l'appelle un filtre linéaire.

1.4.2 Recadrage d'une image

Souvent, pour l'analyse d'images, nous voulons étudier de plus près une zone spécifique de l'image appelée Region Of Interest (ROI). Pour ce faire, nous avons besoin d'une opération qui modifie les coordonnées spatiales de l'image, appelée recadrage.

Le processus de recadrage d'une image consiste à sélectionner une petite partie de l'image, une sous-image et à la couper du reste de l'image. La figure 1.6 montre un exemple d'une image recadrée aux points $P_1(35, 35)$, $P_2(110, 110)$.



(a) Image originale (b) Image recadrée

FIGURE 1.6 – Exemple d'une image recadrée

1.4.3 Redimensionnement d'une image

Le redimensionnement des images fait référence à la mise à l'échelle d'une image ou à la déformation de la grille de pixels originale à une autre. Cette opération est

utile dans de nombreuses applications de traitement d'images et d'apprentissage automatique. Elle permet de réduire le nombre de pixels d'une image. Le redimensionnement permet également de zoomer dans les images pour répondre aux exigences de l'application utilisée.

Le redimensionnement se fait par une opération d'interpolation d'image bidimensionnelle où différentes techniques sont employées.

L'image la plus grossière obtenue provient de l'interpolation du plus proche voisin. Cette méthode d'interpolation se contente de regarder la valeur de la fonction la plus proche d'un point donné et lui attribue la même valeur, ce qui conduit à une image très pixélisée. L'interpolation bilinéaire rapproche la fonction en utilisant une approximation linéaire, ce qui permet d'obtenir une meilleure image. L'interpolation bicubique rapproche la fonction en utilisant l'approximation cubique, ce qui permet d'obtenir une image encore meilleure.

la figure 1.7 montre un exemple d'une image avant et après le redimensionnement.



(a) Image originale (256 × 256) (b) Image réduite (64 × 64) (c) Interpolation du plus proche voisin (d)



(d) Interpolation bilinéaire (e) Interpolation cubique

FIGURE 1.7 – Une image redimensionnée avec différents types d'interpolation

1.5 La vidéo

L'une des plus grandes avancées technologiques de cette ère a été l'émergence de la vidéo numérique comme élément fondamental de la vie quotidienne. Une séquence vidéo contient plus d'informations visuelles qu'une seule image. Cela est dû en partie au fait que le mouvement est capturé dans une séquence d'images, plutôt que dans une seule image.

1.5.1 Définition

La vidéo est la technologie qui permet de capturer des images en mouvement par voie électronique. Ces images en mouvement sont en réalité juste une série d'images fixes qui changent si vite qu'on dirait que l'image bouge. La fluidité (vitesse) d'une vidéo est caractérisée par le nombre d'images par secondes (en anglais Frames per second (FPS)).

Une image numérique dans un flux vidéo est représentée par une matrice de valeurs, où chaque valeur est une fonction de l'information entourant le point correspondant dans l'image. Un élément unique dans une matrice d'image comprend des informations pour toutes les composantes de couleur.

Pour stocker ou transmettre une vidéo, une grande capacité de stockage et un taux de transfert rapide sont nécessaires. L'un des facteurs les plus importants de la compression vidéo est le nombre d'images par seconde et la résolution.

1.5.2 Nombre d'images par seconde

Le nombre d'images par seconde (FPS) est la vitesse à laquelle des images successives, appelées trames, apparaissent sur un écran et forment des images en mouvement. Le contenu vidéo que nous consommons quotidiennement n'est pas réellement en mouvement. En fait, il s'agit d'images fixes qui s'enchaînent les unes après les autres. Si une vidéo est tournée à 24 images par seconde, cela signifie que 24 images individuelles sont lues en une seconde (voir la figure 1.8). Elles changent à un rythme différent selon les supports, en fonction de nombreux autres facteurs. [FPS,]

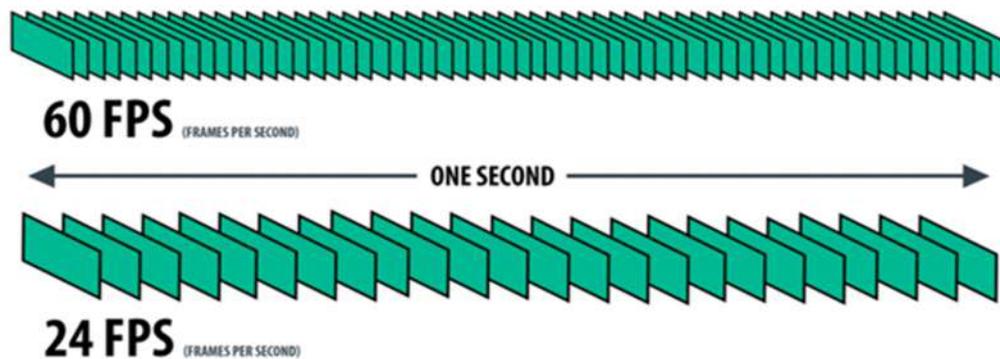


FIGURE 1.8 – Illustration de la notion de FPS

1.5.3 La résolution

La résolution d'un dispositif d'affichage est définie comme le nombre de pixels qui peuvent être affichées. Elle est généralement exprimée par des nombres pour l'horizontale et la verticale : 640 par 480 signifie 640 pixels de largeur, par 480 pixels de hauteur donnant 307.200 pixels. Il existe de nombreux formats de résolution tels que le SD, HD, le Full HD, etc. La figure 1.9 donne un comparatif entre les différentes résolutions.

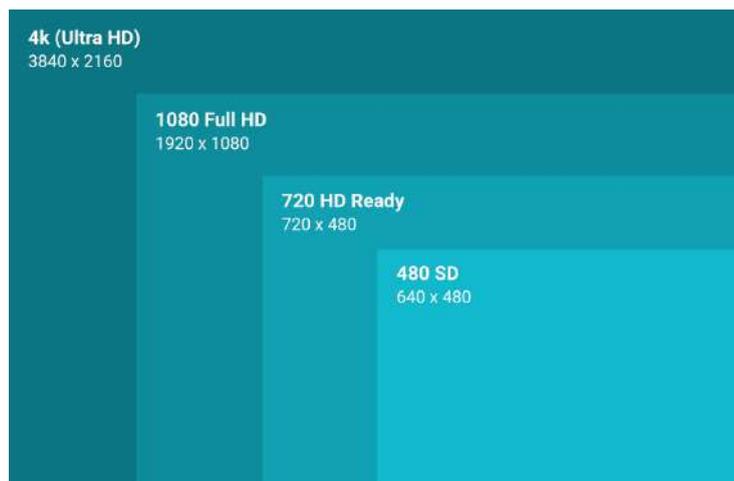


FIGURE 1.9 – Comparatif des différentes résolutions

1.5.4 Domaine d'utilisation de la vidéo

Au cours des deux dernières décennies, on a assisté à un nombre croissant d'applications de la vidéo dans de nombreux domaines, notamment les communications, l'éducation, la médecine et le divertissement. Ces applications vidéo améliorent la communication, favorisent une compréhension plus rapide d'idées complexes et offrent des capacités d'accès accrues à l'information.

Différentes tâches de vidéo-surveillance telles que la détection de mouvement, le contrôle des scènes intérieures et extérieures comme les aéroports, gares, parkings, autoroutes, magasins, etc. sont de plus en plus demandées.

1.6 Conclusion

Les concepts qui servent de base à la compréhension des différentes techniques de traitement de l'image ont été présentés dans ce chapitre. Les caractéristiques des vidéos ont été également examinées. Les techniques de détection de visages et de masques basées sur le traitement d'images peuvent maintenant être abordées dans les chapitres suivants.

Chapitre 2

Intelligence Artificielle

Sommaire

2.1	Introduction	20
2.2	L'intelligence Artificielle (IA)	20
2.3	Machine Learning (ML)	21
2.3.1	L'apprentissage supervisé	22
2.3.2	L'apprentissage non-supervisé	22
2.3.3	L'apprentissage semi-supervisé	22
2.4	Les Principaux Algorithmes du ML	22
2.4.1	K-Nearest Neighbor (K-NN)	22
2.4.2	Support Vector Machine (SVM)	23
2.5	Deep Learning (DL)	23
2.5.1	Cerveau humain et réseaux neuronaux artificiels	24
2.5.2	Les fonctions d'activation	26
2.5.3	Convolutional Neural Network	27
2.5.4	Quelques architectures de Deep Learning	32
2.6	Conclusion	33

2.1 Introduction

Tout ce qui nous entoure est connecté à une source de données où tout est enregistré numériquement ; par exemple, les données de santé, les données COVID-19, et bien d'autres encore. L'intelligence artificielle, et le Machine Learning ont été très développés ces dernières années dans le contexte de l'analyse des données et l'extraction d'informations à partir de ces données. Ce chapitre concerne principalement l'intelligence artificielle et les principaux algorithmes du Machine Learning utilisés.

2.2 L'intelligence Artificielle (IA)

Parfois appelée intelligence des machines l'IA est l'intelligence démontrée par les machines, par opposition à l'intelligence naturelle dont font preuve les humains et d'autres animaux, comme "l'apprentissage" et "la résolution de problèmes".

Il existe de nombreuses définitions de l'IA dont :

- L'IA est "une branche de l'informatique traitant de la simulation du comportement intelligent dans un ordinateur".
- L'IA est "la capacité d'une machine à imiter un comportement humain intelligent".
- L'IA est "un système informatique capable d'effectuer des tâches qui exigent normalement l'intelligence humaine.

L'IA a connu un essor croissant durant la dernière décennie et trouve des applications dans divers domaines (voir la figure 2.1) telles que la reconnaissance des images, la compréhension du langage, etc.

L'intelligence artificielle est le domaine où l'électronique et l'informatique se rencontrent. Elle vise la conception et le prototypage de cartes et de dispositifs électroniques avancés pour leur intégration dans les produits d'équipement, la télémédecine, les systèmes intelligents pour l'agriculture de haute précision, les villes intelligentes.

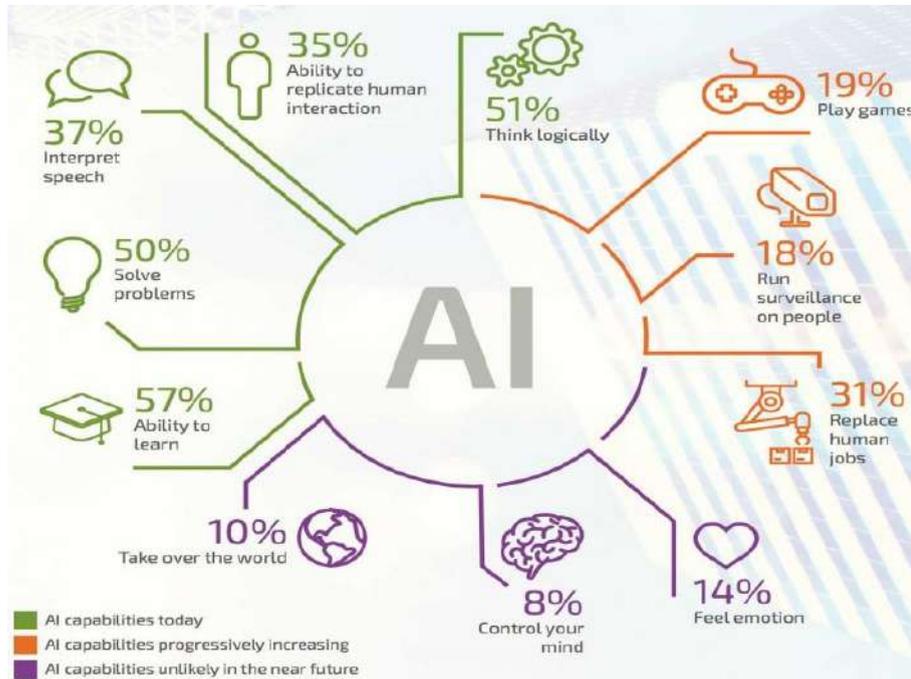


FIGURE 2.1 – Les application de l’IA
[Saini et al., 2019]

2.3 Machine Learning (ML)

Le ML est un sous-ensemble de l’intelligence artificielle qui traite de la conception d’algorithmes afin de construire des modèles pour traiter les données avec plus de précision. Ils ont été récemment utilisés pour résoudre de manière optimale une variété de problèmes techniques et scientifiques.

L’apprentissage automatique consiste donc à faire en sorte que les ordinateurs modifient ou adaptent leurs actions (tel que le contrôle d’un robot) de manière à ce que ces actions deviennent plus précises [Bishop and Nasrabadi, 2006].

Dans le ML les données sont utilisées pour entraîner le modèle selon deux situations différentes : la régression et la classification :

- La régression : problème dans lequel la variable de sortie est une valeur réelle.
- la classification : problème dans lequel la sortie est une étiquette discrète (classe).

En général, les algorithmes basés sur le ML peuvent être classés selon leurs méthodes d’apprentissage comme suit :

2.3.1 L'apprentissage supervisé

Un ensemble d'apprentissage d'exemples avec les réponses correctes (cibles) est fourni et, sur la base de cet ensemble d'apprentissage, l'algorithme se généralise pour répondre correctement à toutes les entrées possibles. Cette méthode est également appelée "apprentissage à partir d'exemples" [Bishop and Nasrabadi, 2006].

2.3.2 L'apprentissage non-supervisé

Les réponses correctes (cibles) ne sont pas fournies, mais l'algorithme tente d'identifier les similitudes entre les entrées de sorte que les entrées qui ont quelque chose en commun soient classées ensemble.

2.3.3 L'apprentissage semi-supervisé

Comme son nom l'indique, ce type d'apprentissage consiste à produire une règle de prédiction à partir de réponses correctes (cibles) et de données supplémentaires sans réponse. L'apprentissage supervisé est utilisé lorsque les données étiquetées sont coûteuses et difficiles à obtenir, car elles nécessitent généralement des annotateurs humains, mais les données non étiquetées sont abondantes et faciles à obtenir [Bishop and Nasrabadi, 2006].

2.4 Les Principaux Algorithmes du ML

De nombreux algorithmes ont été proposés dans la littérature sur l'apprentissage automatique et la science des données. Dans ce qui suit, nous résumons les méthodes les plus courantes et les plus populaires qui sont utilisées dans différents domaines d'application [Bishop and Nasrabadi, 2006].

2.4.1 K-Nearest Neighbor (K-NN)

K-NN est l'un des plus simples algorithmes du ML basé sur une technique d'apprentissage supervisé.

Proposé par [Fix and Hodges, 1951], puis modifié par [Cortes and Vapnik, 1995], le K-NN est utilisé aussi bien pour la régression que pour la classification, mais il est surtout utilisé pour les problèmes de classification.

Dans le NN chaque échantillon est affecté à la classe de son plus proche voisin, ou à la classe la plus commune parmi ses k plus proches voisins dans la variante K-NN. Le NN pur est utilisé lorsque $K = 1$, mais souvent $K > 1$ où un vote majoritaire a lieu. L'algorithme utilise la distance euclidienne entre un échantillon de test x_i et un échantillon d'apprentissage y_i :

$$d(x_i, y_i) = \|x_i - y_i\|_2^2 \quad (2.1)$$

2.4.2 Support Vector Machine (SVM)

Le SVM est l'un des algorithmes les plus populaires de la littérature sur l'apprentissage supervisé [Cortes and Vapnik, 1995].

Étant donné un ensemble de données de classification binaire $(x_i, y_i) \quad i = 1 \dots N$, où $x_i \in \mathbb{R}^n$ est le vecteur de caractéristiques représentant le $i^{\text{ème}}$ exemple d'apprentissage et $y^{(i)} \in \{-1, 1\}$ est l'étiquette de classe correspondant au $i^{\text{ème}}$ exemple d'apprentissage. L'algorithme SVM trouve les paramètres (w, b) de l'hyperplan de séparation linéaire à marge optimale comme l'indique la figure 2.2.

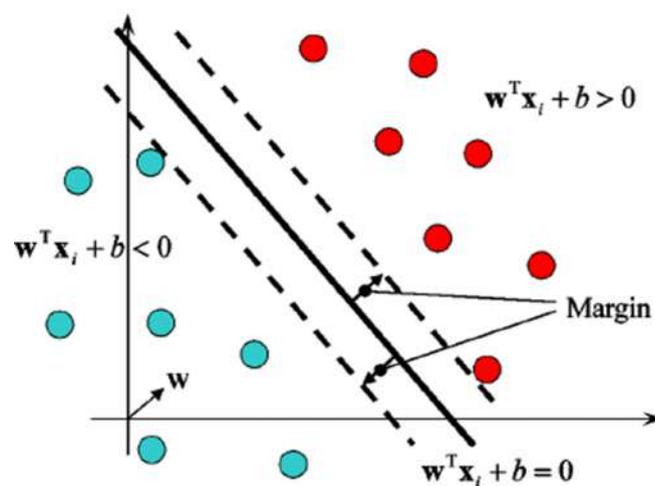


FIGURE 2.2 – Marge optimale de l'hyperplan de séparation linéaire

2.5 Deep Learning (DL)

L'apprentissage profond est un type particulier d'apprentissage automatique. Il s'est imposé et se développe rapidement comme l'un des domaines les plus importants de la science. Le DL se caractérise par un réseau de neurones, qui contient de

nombreuses couches cachées composées de milliers de neurones qui effectuent chacun des opérations petites et simples.

Les résultats d'une première couche de neurones servent d'entrée aux calculs d'une deuxième couche et ainsi de suite.

Une illustration d'un réseau de neurones simple et d'un réseau de neurones profond est donnée dans la figure 2.3.

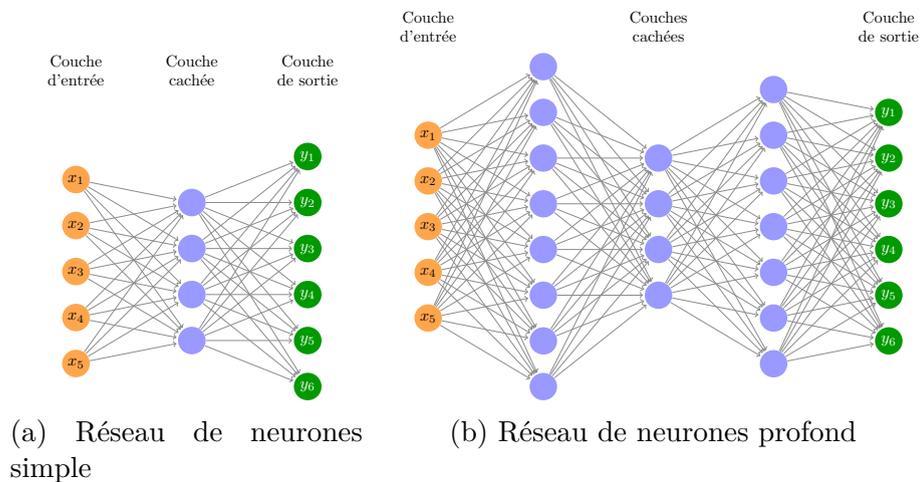


FIGURE 2.3 – Illustration de réseau de neurones

2.5.1 Cerveau humain et réseaux neuronaux artificiels

Les scientifiques et les chercheurs tentent en vain de reproduire l'activité du cerveau humain afin de construire des machines qui apprennent et pensent comme les humains.

Un neurone biologique, comme l'illustre la figure 2.4, comporte plusieurs composants : les dendrites, le corps cellulaire ou soma, l'axone et les terminaisons axonales [Finger, 2001]. Les dendrites servent à recevoir des signaux provenant d'autres neurones en amont. En fonction de ces signaux qui lui parviennent, le soma déclenchera ou non un signal électrique appelé influx nerveux.

Cet influx nerveux va voyager le long de l'axone jusqu'aux terminaisons axonales, où il sera communiqué à d'autres neurones en aval par l'intermédiaire de synapses.

Ce principe est repris pour les neurones formels. Un neurone artificiel se comporte comme une fonction mathématique réelle avec plusieurs variables et des valeurs

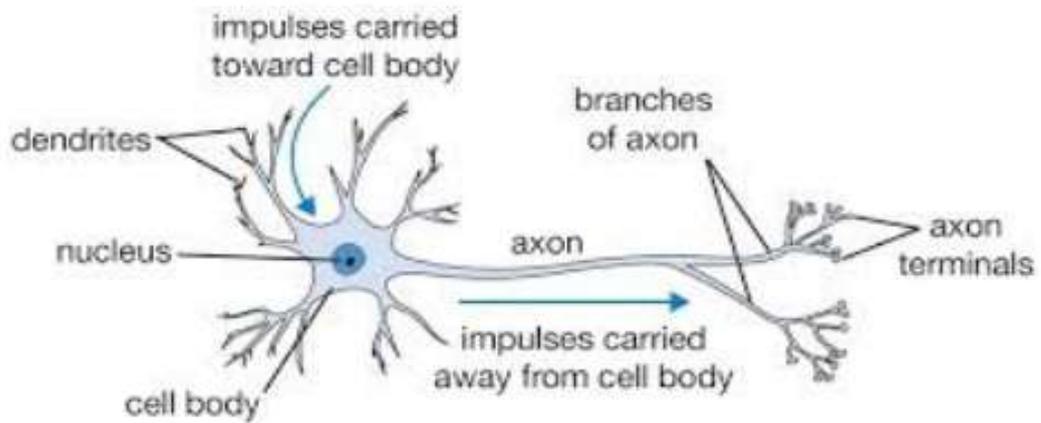


FIGURE 2.4 – Architecture d'un neurone biologique

réelles. Il possède donc plusieurs entrées et une sortie qui correspondent respectivement aux dendrites et à l'axone. Le modèle le plus simple appelé perceptron a été inventé en 1957 par F. Rosenblatt (voir figure 2.5) et est défini par différents éléments.

Un neurone j est défini par :

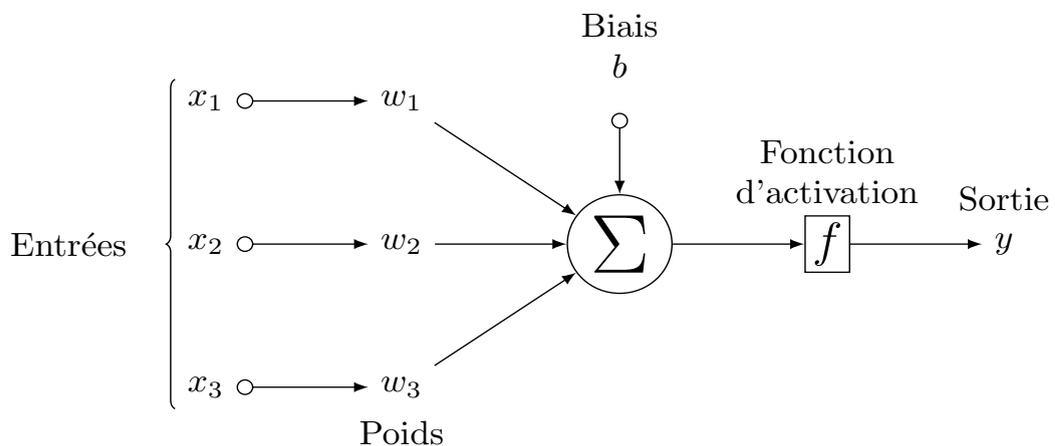


FIGURE 2.5 – Architecture d'un neurone artificiel (perceptron)

- des entrées : x_1, x_2, \dots, x_n
- des poids : w_1, w_2, \dots, w_n qui pondèrent les entrées
- un biais : b_j qui régule l'activation du neurone.
- une fonction d'activation $\Phi()$: détaillée ci-dessous

- une sortie : y_j

2.5.2 Les fonctions d'activation

Les fonctions d'activation sont un élément crucial du DL. Elles déterminent si un neurone se déclenche ou non. Pour apprendre une fonction complexe, la fonction d'activation doit être non linéaire. Dans ce qui suit, nous allons décrire les différentes fonctions d'activation.

Fonction sigmoïde

Le premier but de cette fonction est de réduire la valeur d'entrée entre 0 et 1. Elle est définie comme suit :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$

Fonction Tanh

Est aussi appelée "tangente hyperbolique" produit un résultat entre -1 et 1. Elle est définie comme suit :

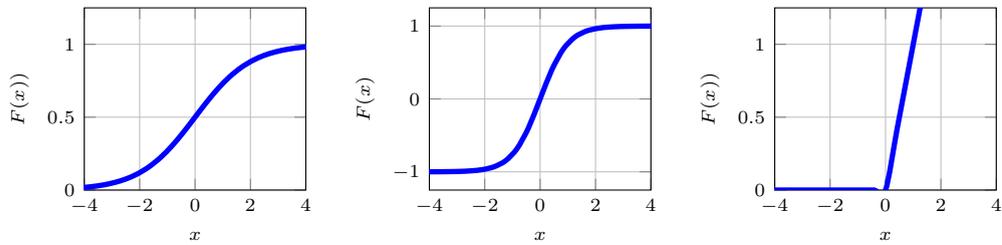
$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.3)$$

Fonction Rectified Linear Unit (ReLU)

est la fonction la plus couramment utilisée en DL. Elle augmente les propriétés de la fonction de décision et de l'ensemble du réseau sans affecter les champs réceptifs de la couche convolutive. Elle est définie comme suit :

$$f(x) = \max(0, x) \quad (2.4)$$

Une représentation graphique des fonctions d'activations suscitée est donnée dans la figure [2.6](#)



(a) Sigmoid activation function (b) Hyperbolic tangent activation (c) Rectified linear unit activation

FIGURE 2.6 – fonction d’activation

2.5.3 Convolutional Neural Network

Le Convolutional Neural Network (CNN), ou plus communément ConvNets, est un type de modèle de DL inspiré par la façon dont le cortex visuel est supposé fonctionner. Son objectif est d’extraire les caractéristiques clés de l’image d’entrée en faisant balayer une matrice appelée *kernel* sur une image pixel par pixel.

Le CNN se compose généralement de trois types de couches : les couches de convolution, de pooling et les couches entièrement connectées. Les deux premières effectuent l’extraction de caractéristiques, tandis que la troisième transforme les caractéristiques extraites en résultats finaux, tels que la classification [LeCun et al., 1995].

Dans ce qui suit, une description détaillée des différents types de couches est donnée.

2.5.3.1 La couche de convolution

La couche convolutive est le bloc de construction central d’un CNN. Les paramètres de la couche consistent en un ensemble de filtres (ou noyaux), qui ont un petit champ réceptif, mais qui s’étendent sur toute la profondeur de l’entrée. Chaque filtre est convolué sur la largeur et la hauteur de l’image d’entrée, en calculant le produit scalaire entre les entrées du filtre et l’entrée. L’empilement des cartes d’activation de tous les filtres le long de la dimension de la profondeur forme le volume de sortie complet de la couche de convolution. Chaque entrée du volume de sortie peut donc également être interprétée comme la sortie d’un neurone qui examine une petite région de l’entrée et partage des paramètres avec les neurones de la même carte d’activation.

La figure 2.7 ci-dessus montre un schéma simple de la couche convolutionnelle.

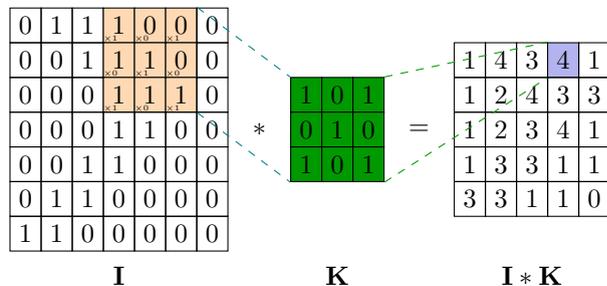


FIGURE 2.7 – La couche de convolution

2.5.3.2 La couche de pooling

Une couche de pooling fournit une opération typique de sous-échantillonnage qui réduit la dimensionnalité dans le plan des cartes de caractéristiques afin d'introduire une invariance de translation à de petits décalages et distorsions, et de diminuer le nombre de paramètres à apprendre. Il est à noter qu'il n'y a pas de paramètre à apprendre dans aucune des couches de pooling. Deux types de pooling existent :

- **Max pooling**

est la forme la plus populaire d'opération de pooling est le max pooling, qui extrait des patches des cartes de caractéristiques d'entrée, produit la valeur maximale dans chaque patch et rejette toutes les autres valeurs. Un max pooling avec un filtre de taille 2×2 avec un stride de 2 est couramment utilisé dans la pratique.

- **Average pooling**

est une opération de pooling qui calcule la valeur moyenne des patches d'une carte de caractéristiques et l'utilise pour créer une carte de caractéristiques sous-échantillonnée. Elle est généralement utilisée après une couche convolutive.

Le average pooling était souvent utilisé dans le passé, mais il est récemment tombé en abandon par rapport au pooling max, qui est généralement plus performant dans la pratique [Scherer et al., 2010].

La figure 2.8 ci-dessus montre un exemple d'opération de max pooling et de average pooling avec une taille de filtre de 2×2 pixels à partir d'une entrée de 4×4 pixels. Lors du max pooling, chaque filtre prend la valeur maximale, puis est arrangé en une nouvelle sortie d'une taille de 2×2 pixels. Le average pooling, quant à lui, prend la valeur moyenne de la taille du filtre.

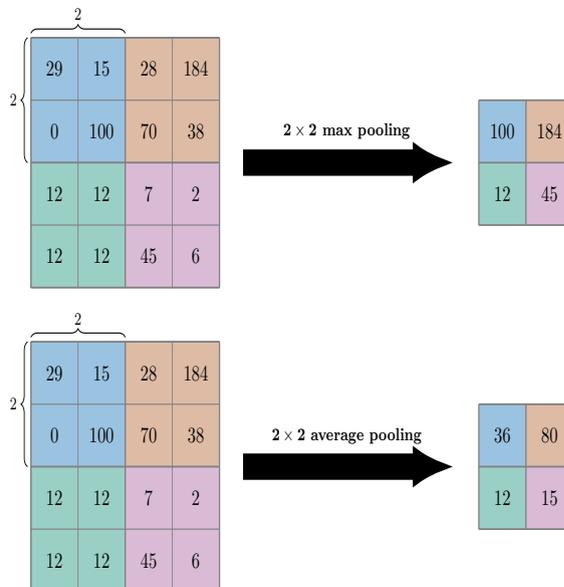


FIGURE 2.8 – Illustration des différents types de pooling

2.5.3.3 La couche flatten

Après avoir terminé les deux étapes précédentes, une carte sous-échantillonnée est obtenue. Comme le nom de cette étape l'indique, l'aplatissement de cette carte en une colonne est obligatoire comme dans la figure 2.9. La raison de cette opération est que nous allons devoir insérer ces données dans un réseau neuronal artificiel par la suite.

La couche dropout

Une autre caractéristique typique des CNNs est la couche Dropout. La couche Dropout est un masque qui annule la contribution de certains neurones à la couche suivante et laisse tous les autres inchangés (figure 2.10). Nous pouvons appliquer une couche Dropout au vecteur d'entrée et également l'appliquer à une couche cachée. Les couches de Dropout sont importantes pour l'apprentissage des CNN car elles empêchent le sur-apprentissage des données d'apprentissage. Si elles ne sont pas présentes, le premier lot d'échantillons d'apprentissage influence l'apprentissage de manière disproportionnée. Ceci, à son tour, empêcherait l'apprentissage de caractéristiques qui n'apparaissent que dans les échantillons ou lots ultérieurs.

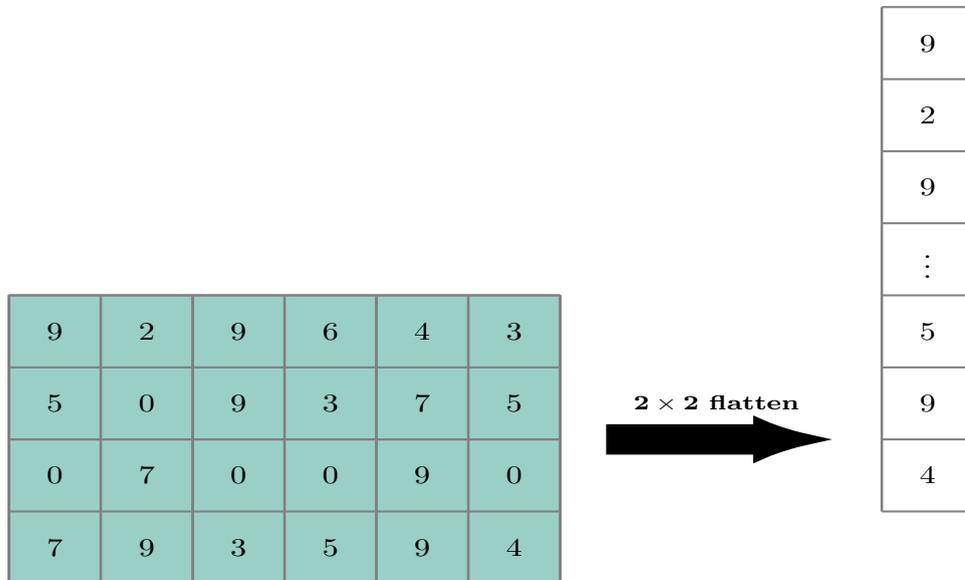


FIGURE 2.9 – Illustration de la couche flatten

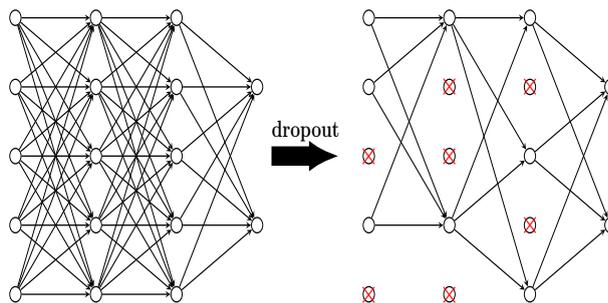


FIGURE 2.10 – La couche dropout layer

Réseaux neuronaux entièrement connectés (Fully Connected) :

Afin de résoudre des problèmes plus complexes, une architecture plus efficace est créée en assemblant plusieurs neurones. Le système neuronal entièrement connecté (FCN) est une structure composée de plusieurs couches cachées de neurones, où tous les nœuds contenus sont connectés à tous les nœuds de la couche suivante. Les FCNs sont couramment utilisés dans les réseaux neuronaux convolutifs et les réseaux neuronaux récurrents. La figure 2.11 donne un exemple d'une couche entièrement connectée.

Lors de la création du réseau neuronal, les poids sont générés de manière arbi-

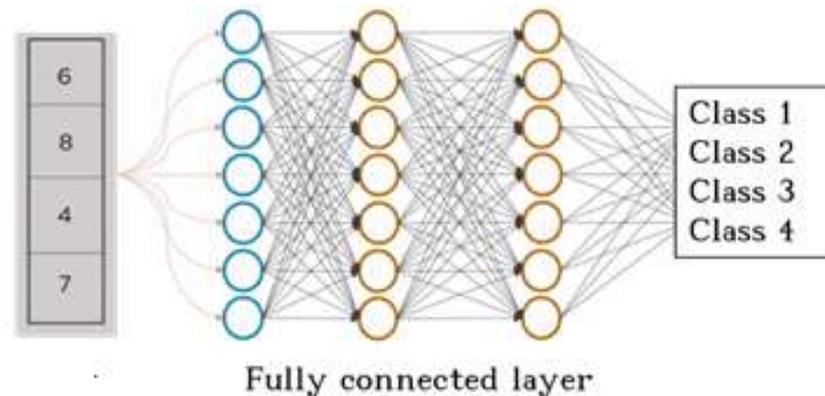


FIGURE 2.11 – Exemple d’une couche entièrement connectée

traire. Il est donc nécessaire de mettre à jour ses poids pour minimiser la valeur de la fonction de coût. Ce processus est appelé "apprentissage" et s’effectue en deux étapes : Forward et Backward.

L’étape Forward consiste à déterminer une prédiction faite par le neurone et à calculer l’erreur par rapport à une valeur de référence indiquée par les données d’apprentissage. L’étape Backward met à jour les poids des neurones en fonction de l’erreur effectuée par l’algorithme de gradient Backpropagation .

Différentes fonctions de coût sont définies telles que :

- **Erreur quadratique moyenne :**

$$L = \frac{1}{n} \left(\sum_{i=1}^n (y^i - Y^i)^2 \right) \quad (2.5)$$

- **Entropie croisée :**

$$L = \frac{1}{n} \sum_{i=1}^n [y^i \log(Y^i) + (1 - y^i) \log(1 - Y^i)] \quad (2.6)$$

où Y est la prédiction du neurone, y est la valeur de référence des données d’apprentissage et n est le nombre de prédictions.

La descente de gradient est un algorithme d’optimisation utilisé pour minimiser (ou maximiser) une fonction objectif. En Machine Learning, la fonction objectif correspond à l’erreur d’apprentissage de dimension d où d, nombre de paramètres du modèle.

L'utilisation de la descente du gradient n'est pas réalisable lorsque la quantité de données est importante.

Pour surmonter ce problème, l'ensemble des données est divisé en mini-batches. Pour chaque mini-batch, la perte et le gradient sont calculés et les paramètres sont mis à jour.

Cette stratégie appelée Stochastic Gradient descent (SGD) est la plus utilisée aujourd'hui en raison de ses performances. Elle offre un bon compromis entre la robustesse et l'évolution rapide du réseau. En plus de l'algorithme SGD, il existe plusieurs autres algorithmes utilisés pour mettre à jour les paramètres et ainsi minimiser la perte tels que l'algorithme ADAM et l'algorithme RMSprop.

Un réseau neuronal peut être entraîné des dizaines, des centaines, voire des milliers d'itérations pour améliorer sa précision. Chaque passage sur l'ensemble des données pour former le réseau neuronal est appelé une époque.

2.5.4 Quelques architectures de Deep Learning

2.5.4.1 LeNet

L'architecture LeNet est une excellente "première architecture" pour les CNN. L'architecture de LeNet consiste en deux ensembles de couches convolutionnelles, d'activation et de pooling, suivis d'une couche entièrement connectée, d'une activation, d'une autre entièrement connectée et enfin d'un classificateur softmax [LeCun et al., 1989].

2.5.4.2 AlexNet

L'Alexnet possède huit couches avec des paramètres apprenables. Le modèle se compose de cinq couches avec une combinaison de max pooling suivi de 3 couches entièrement connectées et ils utilisent l'activation Relu dans chacune de ces couches sauf la couche de sortie [Krizhevsky et al., 2012].

2.5.4.3 VGG

VGG est l'abréviation de Visual Geometry Group (groupe de géométrie visuelle) et se compose de blocs, où chaque bloc est composé de couches de convolution 2D et

de Max Pooling. Il existe en deux modèles VGG16 et VGG19 avec 16 et 19 couches [Simonyan and Zisserman, 2014].

2.5.4.4 ResNet

Les réseaux résiduels profonds (ResNets), tels que le modèle populaire ResNet-50, sont un autre type d'architecture de réseau neuronal convolutif (CNN) d'une profondeur de 50 couches [He et al., 2016]. Un réseau neuronal résiduel utilise l'insertion de connexions raccourcies pour transformer un réseau ordinaire en son homologue résiduel. Par rapport aux VGGNets, les ResNets sont moins complexes car ils comportent moins de filtres.

2.5.4.5 MobilNet

MobileNet est une classe de CNN qui a été mise en libre accès par Google, ce qui nous donne un excellent point de départ pour l'apprentissage de nos classificateurs qui sont extrêmement petits et extrêmement rapides [Howard et al., 2017]. Ils sont basés sur une architecture simplifiée qui utilise des convolutions séparables en profondeur pour construire des réseaux neuronaux profonds légers qui peuvent avoir une faible latence pour les dispositifs mobiles et embarqués.

2.6 Conclusion

Dans ce chapitre, nous avons présenté l'intelligence artificielle et ses applications. Par la suite, nous avons donné une idée générale sur le Machine Learning et ses catégories, commençant par l'apprentissage supervisé, non supervisé, et par renforcement. Nous avons donné une comparaison entre cerveau humain et les réseaux neuronaux artificiels. Après, nous avons présenté les fonction d'activation et le CNN. Nous avons terminé par donné quelques architectures de DL.

Chapitre 3

La Vidéo-surveillance dans le Contexte du COVID-19 par Deep Learning :

Sommaire

3.1	Introduction	35
3.2	Contexte et études connexes	35
3.2.1	Détection de visages par la couleur de peau	35
3.2.2	Détection de visages à l'aide de Eigenface	36
3.2.3	Détection robuste des visages en utilisant la distance de Hausdorff	37
3.2.4	Le détecteur de visage Viola-Jones	38
3.3	Détection de visages et de masques	38
3.3.1	Détection de personnes	38
3.3.2	Single Shot MultiBox Detector (SSD)	40
3.3.3	Détection de masques	42
3.3.4	Mesure de distanciation	47
3.4	Conclusion	48

3.1 Introduction

La protection contre les coronavirus est une contre-mesure obligatoire, selon WHO [Loey et al., 2021]. En effet, le port d'un masque est une méthode efficace pour bloquer 80 des infections respiratoires [Wang et al., 2021]. De plus, WHO recommande de pratiquer la distanciation physique pour limiter la propagation du virus. la vidéo-surveillance joue un rôle majeur afin de s'assurer que les gens portent des masques et maintiennent une distance sociale dans les lieux publics.

3.2 Contexte et études connexes

La détection automatique des visages est la pierre angulaire de toutes les applications qui tournent autour de l'analyse automatique des images faciales telles que la reconnaissance et la vérification des visages, le suivi des visages pour la surveillance, l'analyse du comportement facial, la reconnaissance des attributs faciaux (c'est-à-dire la reconnaissance du genre, la race et de l'âge et le morphing des visages.

La détection des visages est un processus de segmentation dans une scène donnée. Pour les humains, cette tâche n'est pas difficile à réaliser. Mais pour les machines, c'est une tâche très difficile à cause des difficultés dues aux expressions faciales, poses et conditions d'éclairage.

Dans la littérature, diverses techniques de détection de visages existent. Un aperçu de quelques techniques importantes et techniques récentes es donnée dans ce qui suit.

3.2.1 Détection de visages par la couleur de peau

La couleur de la peau humaine a été utilisée pour détecter les visages et il a été démontré qu'il s'agit d'une caractéristique spécifique au visage. Cette technique utilise les informations de couleur pour distinguer les pixels de peau de ceux qui ne le sont pas. Un espace couleur est utilisé pour coder chaque pixel dans une image couleur (par exemple RGB ou YCrCb, ...). Chaque étape de cette procédure peut être résumée comme suit :

- le prétraitement de l'image
- La sélection d'un espace chromatique
- Le seuillage des couleurs et la segmentation de la couleur de la peau



FIGURE 3.1 – Détection du visage en utilisant la couleur de la peau [Kovac et al., 2003]

La rapidité du traitement et la simplicité de la décision sont les caractéristiques de cette méthode de détection du visage en utilisant la couleur de la peau. En effet, le principe est simple et se concentre uniquement sur la couleur de la peau, sans tenir compte de facteurs tels que l'échelle ou la position. La méthode peut produire des faux positifs et poser des problèmes au niveau des informations de fond. Un exemple de la détection du visage en utilisant la couleur de la peau est donné dans la figure 3.1.

3.2.2 Détection de visages à l'aide de Eigenface

La méthode d'utilisation de Eigenface a été introduite par Sirovich et Kirby (1987) et utilisée par Matthew Turk et Alex Pentland dans la classification des visages. Lorsque une entrée est fournie à la machine, elle contient un certain motif qui apparaît dans chaque image. Ces motifs sont des caractéristiques de base qui sont présentes dans chaque image de visage. Ces caractéristiques ou modèles sont appelées "visages propres" ou "composantes principales". On peut extraire ces caractéristiques en utilisant une méthode mathématique connue sous le nom de Analyse en Composantes Principales (ACP). Il s'agit d'une méthode mathématique qui permet d'extraire et d'utiliser les composantes principales. Son fonctionnement est comme suit.

Tout d'abord, un ensemble d'images d'apprentissage est donné pour permettre d'extraire les visages propres. Ensuite, le poids est calculé en utilisant l'image et stocké dans la galerie. Après cela, si une nouvelle image arrive, ses poids sont calculés et comparés avec les poids présents dans la galerie. Si une correspondance la

plus proche est présente dans la galerie, on dit que cette image est appariée. Pour comparaison, on utilise la distance Euclidienne.

3.2.3 Détection robuste des visages en utilisant la distance de Hausdorff

Il s'agit d'un algorithme basé sur un modèle qui fonctionne sur les images fixes en niveau de gris. Il utilise la distance de Hausdorff ce qui le rend efficace et robuste pour les applications en temps réel [Jesorsky et al., 2001]. La distance de Hausdorff est utilisée comme une mesure de similarité entre l'image dans laquelle l'objet doit être détecté et un échantillon de visage. Cet algorithme fonctionne en deux étapes :

1. Détection grossière
2. Raffinement

Détection grossière : tout d'abord, la zone d'intérêt est déterminée pour chaque image entrante avant d'appliquer une quelconque méthode. Cette région est rééchantillonnée à une taille fixe.

Segmentation : A partir de l'image rééchantillonnée, une image d'intensité des bords est calculée avec l'opérateur de Sobel. Il en résulte des points d'arête binaires.

Localisation : Avec la représentation binaire obtenue par l'étape de segmentation et un modèle de visage modèle de visage , une localisation du visage dans l'image peut être calculée.

La figure 3.2 résume le processus en deux étapes :

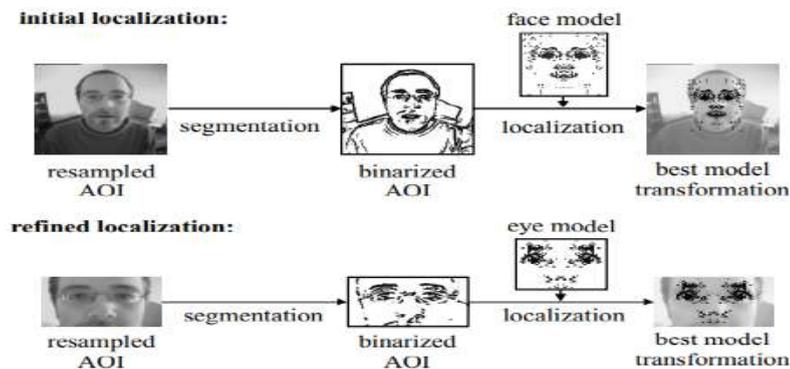


FIGURE 3.2 – Détection de visages en utilisant la distance de Hausdorff [Jesorsky et al., 2001]

3.2.4 Le détecteur de visage Viola-Jones

Viola Jones est une technique de détection des visages proposé par Paul Viola et Michael Jones en 2001. Cet algorithme scanne une image avec une fenêtre à la recherche des caractéristiques d'un visage humain. Si ces caractéristiques sont trouvées et ont une valeur particulière, alors la fenêtre de l'image est considérée être un visage. Pour résoudre le cas de visages de tailles différentes, la fenêtre est mise à l'échelle avec le processus répété pour chaque image [Viola and Jones, 2001].

Il permet d'atteindre un taux de détection élevé avec 15 images par seconde à travers quatre étapes :

- Sélection des caractéristiques Haar
- Création d'une image intégrale
- Apprentissage AdaBoost
- Classificateurs en cascade

3.3 Détection de visages et de masques

la communauté scientifique travaillent à l'élaboration de méthodes automatiques permettant d'identifier et de faire respecter le port de masques faciaux dans les lieux publics. Le problème peut être résumé comme suit : étant donné une image provenant d'un flux vidéo, le visage est détecté puis donné à un modèle de classification pour catégoriser cette image dans une tâche de détection de masque en utilisant le modèle de classification.

Les principales étapes de la technique que nous proposons sont illustrées dans la figure 3.3.

3.3.1 Détection de personnes

La détection des visages constitue une tâche essentielle dans les applications de surveillance et de sécurité. Dans cette toute première étape, l'objectif principal est d'identifier la partie de l'image ou d'une vidéo donnée qui représente un visage ainsi que de reconnaître l'emplacement de ces visages. Ensuite, le résultat de cette étape peut transformer les données en fenêtres qui contiennent chaque visage comme image d'entrée. Il s'agit essentiellement d'un problème de segmentation basé sur des caractéristiques extraites de points de repère faciaux. La plupart des systèmes de

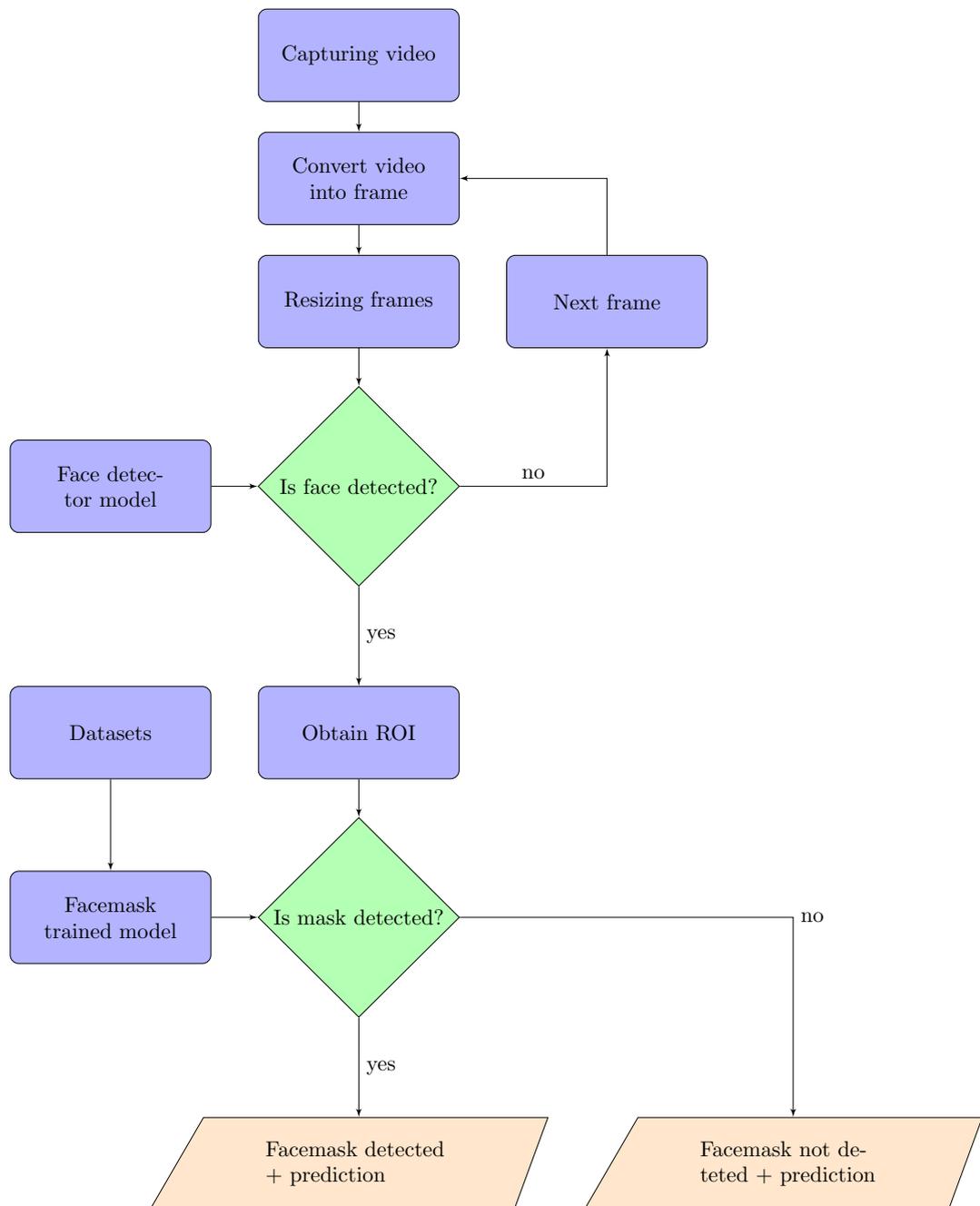


FIGURE 3.3 – Diagramme de la méthode utilisée

détection de visages tentent d'extraire une fraction du visage entier, ce qui éliminant ainsi la plupart de l'arrière-plan et d'autres zones de la tête d'un individu, comme les cheveux, qui ne sont pas nécessaires à la tâche de reconnaissance du visage. La figure

3.4 montre Une détection de visage réussie dans une image avec une vue frontale d'un visage humain.

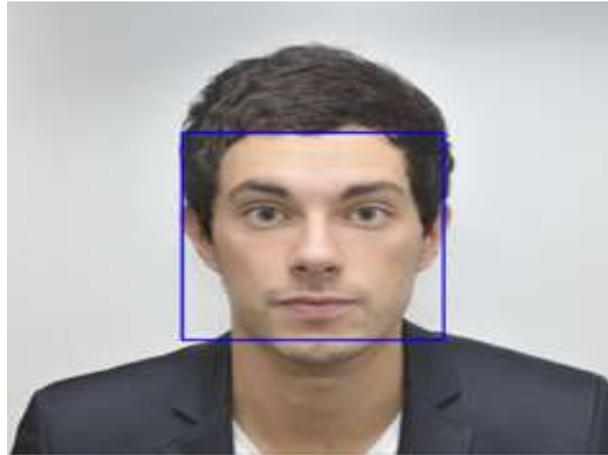


FIGURE 3.4 – Détection de visage dans une image

3.3.2 Single Shot MultiBox Detector (SSD)

Le SSD a été développé par Liu et al. et publié en 2015 sous le nom de Single Shot MultiBox Detector (SSD). Rapide et précis, le SSD signifie aux composants du nom [Liu et al., 2016].

- **Single Shot** : les tâches de classification et de localisation des objets sont effectuées en une seule passe avant du réseau
- **MultiBox** : c'est le nom d'une technique de régression par boîte englobante
- **Detector** : le réseau est un détecteur d'objets qui classe également les objets détectés.

Le modèle du réseau SSD se compose de trois parties.

Tout d'abord, l'image d'entrée est introduite dans un réseau classifieur, appelé ici réseau de base. Dans ce travail, nous utilisons le réseau VGG 16 à cette fin. Les couches entièrement connectées de l'architecture standard sont remplacées par des couches de convolution.

La sortie du réseau de base modifié est introduite dans une série de couches de caractéristiques supplémentaires. Ces couches diminuent en taille et permettent la prédiction de détections à plusieurs échelles.

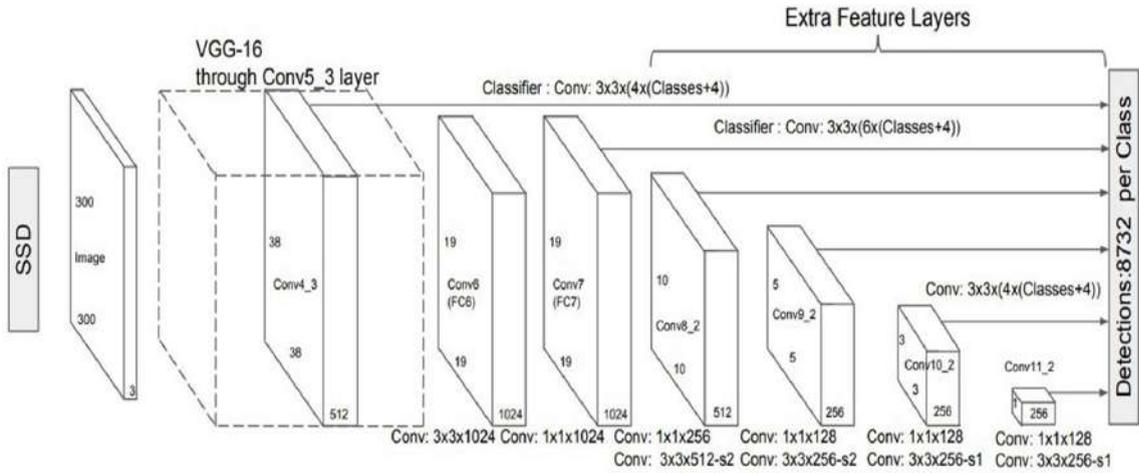


FIGURE 3.5 – Architecture du réseau SSD 300

Enfin, la sortie des couches de caractéristiques supplémentaires et une couche sélectionnée du réseau de base sont utilisées comme prédicteurs de détection. Chacune de ces couches peut produire un ensemble fixe de prédictions en utilisant un nouvel ensemble de couches convolutives. ‘

MultiBox démarre avec un ensemble fixe de boîtes d’ancrage par défaut en tant que références pour les prédictions et tente d’approcher les boîtes de délimitation de la vérité du sol en utilisant les références.

Pour chaque couche de caractéristiques, nous calculons des boîtes d’ancrage par défaut pour un ensemble donné de rapports d’aspect et d’échelles. Le nombre de boîtes d’ancrage pour chaque couche est donnée dans la table 3.1.

TABLE 3.1 – Nombre de boites pour le réseau SSD 300

Couche	Height	Width	# boites
conv4_3	38	38	4
conv7	19	19	6
conv8_2	10	10	6
conv9_2	5	5	6
conv10_2	3	3	4
conv11_2	1	1	4

Cela donne un total de 8732 boîtes d’ancrage.

3.3.3 Détection de masques

Ce travail a pour objectif de concevoir un système permettant de savoir si une personne utilise un masque ou non. Une fois les visages détectés, les images faciales sont extraites puis utilisées pour identifier le masque sur le visage en utilisant l'algorithme d'apprentissage Convolutional Neural Network (CNN).

3.3.3.1 Le CNN pour la détection

La figure 3.6 montre une vue d'ensemble du réseau de neurones convolutifs que nous utilisons. Il peut être divisé en 11 couches décrites comme suite :

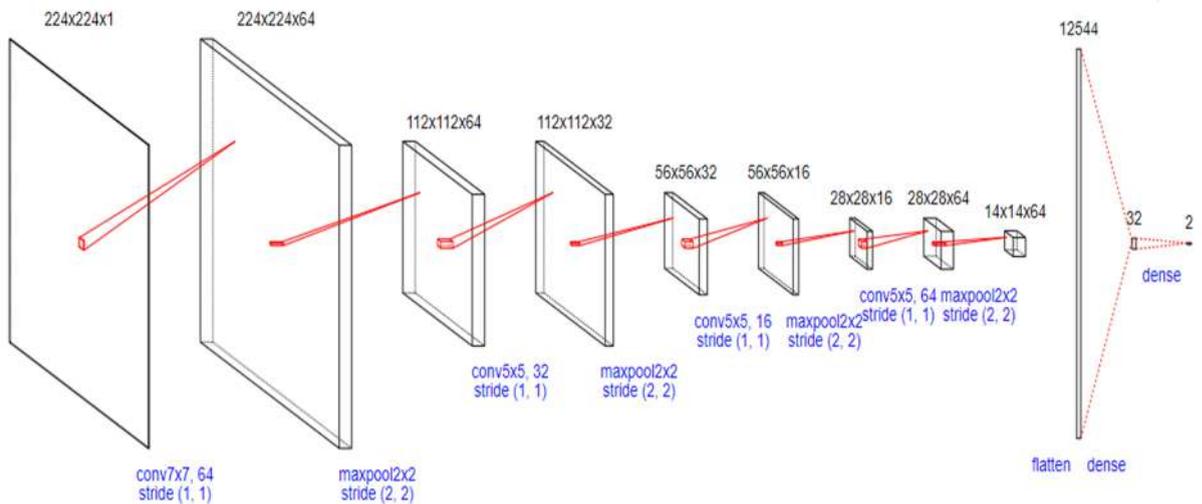


FIGURE 3.6 – Architecture du modèle de détection de masques

- La couche 1 est constituée d'une couche convolutive avec fonction d'activation ReLu. Cette couche reçoit une image de taille 224×224 . La taille du filtre convolutif est de 7×7 , le padding est same est le stride est de 1 et le nombre de filtre est de 64.
- La couche 2 est la couche max pooling. Cette couche reçoit une entrée de taille $64@224 \times 224$ de la couche précédente. La taille du pooling= 2×2 , le padding = 0 et le stride= 2 Cette couche n'a pas de fonction d'activation.
- La couche 3 est constituée d'une couche convolutive avec fonction d'activation ReLu. Cette couche reçoit une entrée de taille $64@112 \times 112$. La taille du filtre convolutif est de 5×5 , le padding est same est le stride est de 1 et le nombre de filtre est de 32.

- La couche 4 est la couche max pooling. Cette couche reçoit une entrée de taille $32 \times 112 \times 112$ de la couche précédente. La taille du pooling = 2×2 , le padding = 0 et le stride = 2.
- Le vecteur de caractéristiques issu des convolutions a une dimension de 12544 obtenu par la couche flatten.
- L'avant dernière couche du réseau est une couche entièrement connectée composée de 32 noeuds avec fonction d'activation ReLu.
- La dernière couche du réseau est une couche entièrement connectée composée de 2 noeuds avec fonction d'activation Softmax. Cette couche calcule les scores des classes, résultant en un vecteur de taille 2, où chacun correspond à un score de classe avec masque et sans masques respectivement.

3.3.3.2 Environnement de travail

Il est nécessaire de mettre en place l'environnement de travail du projet. Pour ce projet, nous proposons Google Colab.

Google Colab est un service en cloud gratuit qui prend désormais en charge le GPU gratuit, qui permet d'évaluer les compétences en langage de programmation Python. Il s'agit d'un environnement de carnet de notes Jupyter composé de cellules pouvant contenir du code, du texte, des figures et ne nécessitant aucune installation ou configuration.

3.3.3.3 Base de données et protocole d'évaluation

Le développement du modèle de détection des masques de visage commence par la collecte des données. Une base de données est constituée d'images de personnes qui portent des masques et celles qui n'en portent pas.

Dans ce travail, 1915 images de sujets avec masque et 1759 de sujets sans masque sont d'abord collecter pour former la base de données. L'étape suivante consiste à étiqueter ces images en deux groupes : avec et sans masque. La figure 3.8 montre quelques exemple de visages avec et sans masque. L'ensemble des données est divisé selon un ratio 80 : 20, ce qui signifie que 80% des données sont prises pour l'apprentissage et 20% pour le test (voir le listing 3.1).

```
1 from sklearn.model_selection import train_test_split
```



FIGURE 3.7 – Colaboratory (Exemple)



FIGURE 3.8 – Exemple de visages avec et sans masque

```

2 X_train,X_test,y_train,y_test= train_test_split(X,y,test_size =
  0.2,random_state = 42)
3 print(X_train.shape)
4 print(y_train.shape)
5 print(X_test.shape)
6 print(y_test.shape)

```

Listing 3.1 – Répartition de la base de données

une fois le modèle créé et la base de données répartie, des paramètres d'initia-

lisation tels que la taille du batch, le nombre d'époques, le taux d'apprentissage, doivent être ajustés pour obtenir de meilleures performances. performance. La taille du batch est le nombre d'images dans un lot (batch). Le nombre d'époques se définit comme étant le nombre de fois que toutes les données d'apprentissage sont passées par le réseau. Le taux d'apprentissage est un paramètre très sensible qui pousse le modèle vers la convergence. La recherche de sa meilleure valeur se fait par un processus expérimental. Dans ce travail, l'étape d'apprentissage est effectuée avec des hyper paramètres définis comme `Batch_size = 32` et `Epochs = 20` (voir le listing 3.2). Enfin, le modèle est compilé en appelant la fonction `compile()`, qui a comme entrées :

- **optimiser** utilisé pour optimiser le réseau neuronal, tel que Adagrad et Adam.
- **loss** qui fait référence à la valeur que le modèle doit minimiser, comme l'entropie croisée et l'erreur quadratique moyenne.
- **Metrics** qui évaluent la précision des problèmes de classification.

```

1 model.compile(loss='categorical_crossentropy',
2               optimizer='adam',
3               metrics=['accuracy'])
4 history=model.fit(X_train,y_train,batch_size=32,epochs=20,verbose
                    =1)

```

Listing 3.2 – Hyper paramètres et compilation du modèle

Le modèle est entraîné à l'aide de la méthode `fit()`. Cette méthode permet au modèle d'itérer sur les données et de trouver le réseau neuronal le plus optimal pour ces données. Une matrice de confusion est un outil qui permet de visualiser les performances d'un algorithme utilisé dans l'apprentissage supervisé. Dans un problème à deux classes, le modèle de classification prédit l'une des deux classes qui sont généralement désignées sous le nom de classes positive et négative. Étant donné un classificateur et un exemple à classer, quatre situations sont envisageables. Si l'exemple est positif et qu'il est classé comme positif, il compte comme un vrai positif *TP* ; s'il est classé négatif, il compte comme un faux négatif *FN*. Si l'exemple est négatif et qu'il est classé comme négatif, il compte comme un vrai négatif *TN* ; s'il est classé positif, il compte comme un faux positif *FP*. Les sorties produites par le classificateur peuvent être représentées par une matrice de confusion de 2×2 . La Table 3.2 montre la matrice de confusion pour un problème à deux classes où les

lignes indiquent la classe réelle et les colonnes indiquent la classe prédite.

TABLE 3.2 – Matrice de confusion d’un problème à deux classes

		Classe prédite	
		Positive	Negative
Classe réelle	Positive	TP	FN
	Negative	FP	TN

Le taux de bonne classification est la mesure de performance la plus simple et qui est à la base de tout autre critère. Le taux de classification calcule le pourcentage d’échantillons correctement classés par rapport au nombre total d’échantillons. À partir des valeurs de la matrice de confusion (voir la Table 3.2), le taux de classification est donné par l’équation suivante :

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

3.3.3.4 Résultat obtenus

Parmi les images de test, le modèle proposé atteint un taux de classification de 97.84%. Les performances du classifieur sont analysées à l’aide de la matrice illustré dans la Figures 3.9.

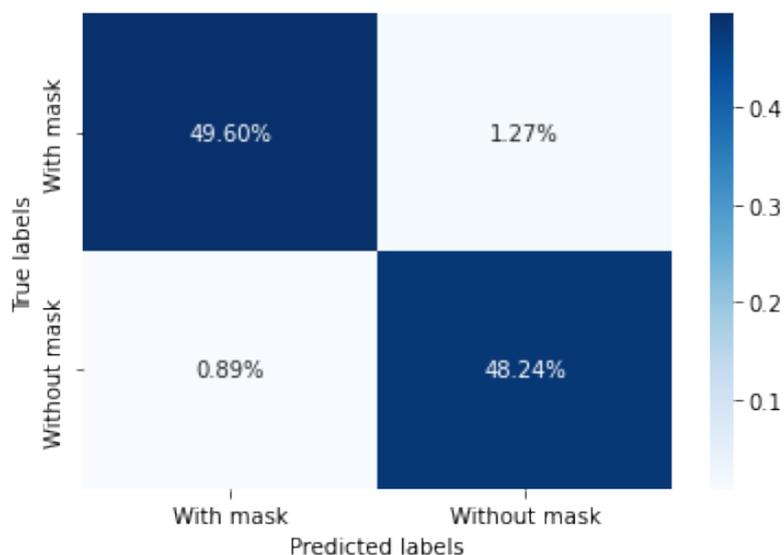


FIGURE 3.9 – La matrice de confusion

On remarque que le modèle basé sur le CNN a très peu d’erreurs de classification.

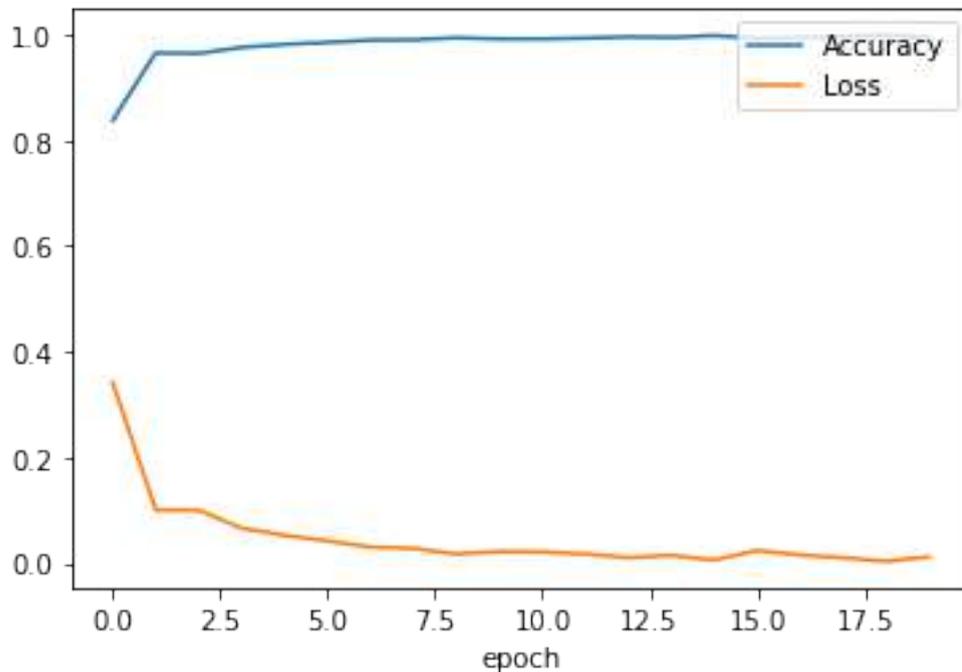


FIGURE 3.10 – Le taux d’apprentissage et de l’erreur en fonction du nombre d’époque

Ce modèle classe les images avec masque comme sans masque avec un taux de 1.27% et classe les images sans masque comme avec masque avec un taux de 0.89%.

Puisque le modèle a été entièrement entraînés puis testé ; il peut être sauvegardé dans un fichier h5 afin d’être utiliser dans une étape ultérieure. Notons que ce fichier contient à la fois la définition du modèle et les poids calculés dans la phase de d’apprentissage.

L’évolution du taux d’apprentissage et de l’erreur en fonction du nombre d’époque est illustré dans la figure fig :performance.

3.3.4 Mesure de distanciation

Également appelée distanciation physique, l’idée principale est que si les gens ne sont pas proches les uns des autres, ils ne peuvent pas propager les germes, les virus et les infections. En se référant au site officiel de WHO, la distanciation sociale autorisée pour limiter la propagation de l’infection par le COVID-19 est d’un mètre au minimum [Albayati et al., 2021], comme le montre la figure 3.11. Pour mesurer

la non-adoption du protocole de distanciation sociale, le terme violation est utilisé.

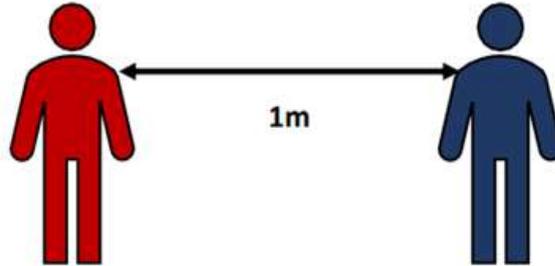


FIGURE 3.11 – Distance Sociale

Après la détection des visages, la distance Euclidienne entre chaque paire de centroïdes détectée est calculée à l'aide de la boîte de délimitation détectée et de ses informations sur les centroïdes. Un seuil minimal prédéfini de violation de la distance sociale est spécifié en utilisant des hypothèses de distance par pixel. Pour vérifier si la distance calculée est inférieure ou non au seuil de violation, les informations estimées sont comparées au seuil de violation. La couleur de la boîte englobante est initialisée en vert ; si la boîte englobante se trouve sous le seuil de violation, sa couleur est mise à jour en rouge.

3.4 Conclusion

Dans ce chapitre, nous avons présenté un bref état de l'art sur les méthodes de détection de visage. Par la suite, le cadre basé sur le deep learning pour la détection de visage et du masque facial a été introduit. Pour entraîner le modèle, des données d'images étiquetées sont utilisées. Le système proposé détecte un masque facial précision de 97.35%. Le système proposé dans cette étude servira d'outil pour l'implémentation dans un Raspberry Pi.

Deuxième partie

Partie Pratique

Chapitre 4

Environnement de développement

Sommaire

4.1	Introduction	51
4.2	Environnement hardware	51
4.2.1	Présentation du Raspberry pi	51
4.2.2	Ecran TFT LCD 7	55
4.2.3	Une caméra Pi	56
4.2.4	Carte Micro SD	57
4.3	Environnement software	58
4.3.1	Raspbian	58
4.3.2	Langage Python	58
4.3.3	Bibliothèques utilisées	59
4.3.4	L'IDE Thonny	63
4.3.5	VNC Viewer	64
4.4	Conclusion	64

4.1 Introduction

Le but de notre travail est la détection par vidéo-surveillance. Dans le chapitre précédent, on a fait une étude théorique. dans le présent chapitre, les outils hardware et software seront utilisés pour l'implémentions de notre système.

4.2 Environnement hardware

4.2.1 Présentation du Raspberry pi

Le Raspberry Pi est un mini-ordinateur mono-carte Single Board Computer (SBC) peu coûteux, de la taille d'une carte de crédit qui se branche sur un écran d'ordinateur ou un téléviseur et qui utilise un clavier et une souris standard. C'est un petit dispositif performant qui permet aux personnes de tous âges d'explorer l'informatique et d'apprendre à programmer dans des langages comme Scratch et Python. Il est capable de faire tout ce que l'on attend d'un ordinateur de bureau, de la navigation sur Internet à la lecture de vidéos haute définition, en passant par la création de feuilles de calcul, le traitement de texte et les jeux ; la programmation.

4.2.1.1 Historique

Le Raspberry Pi est conçu et fabriqué au Royaume-Uni dans le but de fournir un dispositif informatique bon marché pour l'éducation. Cependant, depuis sa sortie, il s'est développé bien au-delà de la sphère académique. Ses origines se trouvent dans le laboratoire informatique de l'université de Cambridge en 2006. L'informaticien Eben Upton, ainsi que Rob Mullins, Jack Lang et Alan Mycroft, s'inquiétaient du fait que les étudiants de premier cycle en informatique étaient de plus en plus éloignés des aspects techniques de l'informatique. Cela était dû en grande partie aux programmes scolaires qui mettaient l'accent sur l'utilisation des ordinateurs plutôt que sur leur compréhension. C'est à partir de cette préoccupation initiale que la fondation Raspberry Pi a été créée. Au cours des six années suivantes, le premier lancement commercial du Raspberry Pi a eu lieu en Février 2012.

La carte a subi un certain nombre de révisions et est disponible en deux modèles, à savoir le modèle A et le modèle B.

Le modèle A est le moins cher et le plus simple des deux ordinateurs, tandis que le modèle B est le plus puissant et prend notamment en charge la connectivité Ether-

net.[Dennis, 2016]

La figure 4.1 présente les différentes générations du Raspberry Pi.

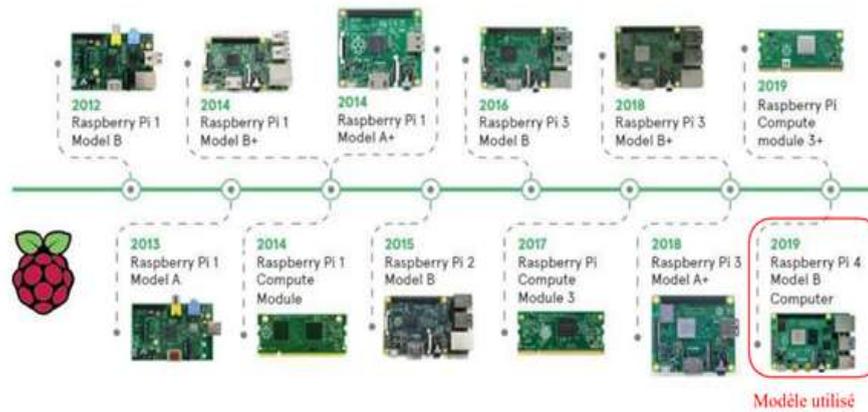


FIGURE 4.1 – Evolution du Raspberry Pi

Une comparaison entre les différents modèles de Raspberry Pi les plus utilisées dans le Tableau /reftab :my-table

4.2.1.2 Composant de base

Le Raspberry Pi 4 Model B (voir la figure 4.2) est le dernier produit de la populaire gamme d'ordinateurs Raspberry Pi.



FIGURE 4.2 – Raspberry Pi 4 model B

Modèle	PRIX (DOLLARS USD)	CPU	RAM	Wireless	Ports d'E/S
Raspberry Pi 4 B (8GB)	75	1.5-GHz, 4-core Broadcom BCM2711 (Cortex-A72)	8GB	802.11ac / Bluetooth 5.0	2x USB 3.0 2x USB 2.0 1x Gigabit Ethernet 2x micro HDMI
Raspberry Pi 4 B (4GB)	55	1.5-GHz, 4-core Broadcom BCM2711 (Cortex-A72)	4GB	802.11ac / Bluetooth 5.0	2x USB 3.0 2x USB 2.0 1x Gigabit Ethernet 2x micro HDMI
Raspberry Pi 4 B (2GB)	35	1.5-GHz, 4-core Broadcom BCM2711 (Cortex-A72)	2GB	802.11ac / Bluetooth 5.0	2x USB 3.0 2x USB 2.0 1x Gigabit Ethernet 2x micro HDMI
Raspberry Pi 3 B+	35	1.4-GHz, 4-core Broadcom BCM2837B0 (Cortex-A53)	1GB	802.11ac / Bluetooth 4.2/ Ethernet	4 x USB 2.0 HDMI 3.5mm audio
Raspberry Pi Zero W	10	1-GHz, 1-core Broadcom BCM2835 (ARM1176JZF-S)	512MB	802.11n / Bluetooth 4.1	1x micro USB 1x mini HDMI
Raspberry Pi Zero WH	17	1-GHz, 1-core Broadcom BCM2835 (ARM1176JZF-S)	512MB	802.11n / Bluetooth 4.1	1x micro USB 1x mini HDMI
Raspberry Pi Zero	5	1-GHz, 1-core Broadcom BCM2835 (ARM1176JZF-S)	512MB	N/A	1x micro USB 1x mini HDMI

TABLE 4.1 – Comparaison entre différents modèles de Raspberry

Il offre une augmentation révolutionnaire de la vitesse du processeur, des performances multimédia, de la mémoire et de la connectivité par rapport à la génération précédente Raspberry Pi 3 Model B+, tout en conservant une compatibilité ascendante et une consommation d'énergie similaire.

Le Raspberry Pi 4 Model B offre des performances de bureau comparables à celles des PC x86 d'entrée de gamme.

4.2.1.3 Spécifications

- **processeur** : Broadcom BCM2711, SoC quadri-cœur Cortex-A72 (ARM v8) 64-bit @ 1.5GHz
- **Mémoire** : 4 Go
- **Connectivité** :
 - LAN sans fil IEEE 802.11b/g/n/ac à 2,4 GHz et 5,0 GHz.
 - LAN, Bluetooth 5.0, BLE
 - Gigabit Ethernet
 - 2 ports USB 3.0
 - 2 ports USB 2.0
- **GPIO** : : connecteur GPIO standard à 40 broches (entièrement rétro compatible avec les cartes précédentes).
- **Vidéo et son** :
 - 2 × ports micro HDMI (jusqu'à 4Kp60 supporté)
 - Port d'affichage MIPI DSI à 2 voies
 - Port caméra MIPI CSI à 2 voies
 - Port audio stéréo 4 pôles et port vidéo composite
- **Multimédia** :
 - H.265 (décodage 4Kp60)
 - H.264 (décodage 1080p60, codage 1080p30)
 - Graphisme OpenGL ES, 3.0
- **Support de carte SD** : Fente pour carte Micro SD pour le chargement du système d'exploitation et le stockage des données.
- **Alimentation en entrée** :

- 5V DC via le connecteur USB-C (minimum 3A)
- 5V DC via l'embase GPIO (minimum 3A)
- Compatible avec l'alimentation par Ethernet (PoE) (nécessite un adaptateur PoE séparé).
- **Environnement** : Température de fonctionnement : 0-50°Cv.[Pi, 2019]

La figure 4.3 donne une description détaillée des différents composants d'une carte Raspberry Pi 4 model B.

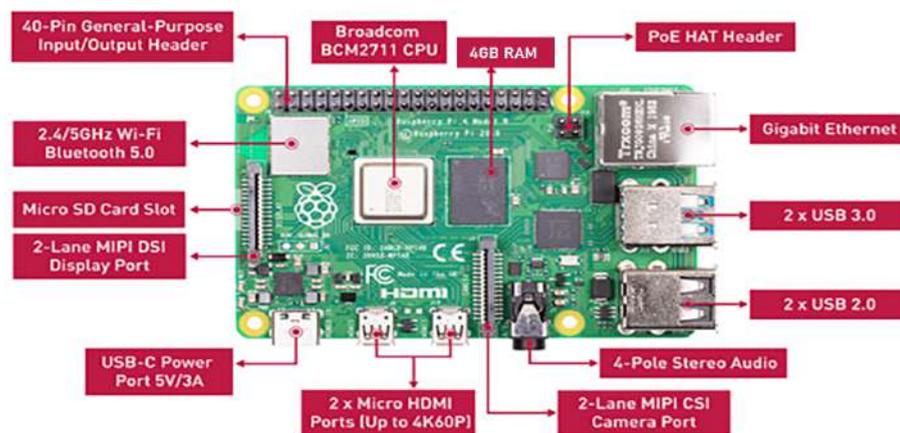


FIGURE 4.3 – Raspberry Pi 4 model B

4.2.2 Ecran TFT LCD 7

Le Thin Film Transistor (TFT) est un écran Liquid Crystal Display (LCD) (comme le montre la figure 4.4) qui utilise une fine couche de semi-conducteur déposée sur le panneau, qui permet de contrôler activement l'intensité de la couleur de chaque pixel. Un écran tactile de 7 pouces avec une résolution 1024x600 pixels avec des entrées Universal Serial Bus (USB) et High-Definition Multimedia Interface (HDMI). L'écran Fonctionne avec le mini-ordinateur Raspberry Pi. Il est compatible avec les systèmes d'exploitation suivants : Raspbian, Ubuntu, Win10 IoT et Kali.[Editor CNN, 2011]



FIGURE 4.4 – Ecran TFT LCD 7" pour Raspberry

4.2.3 Une caméra Pi

Le module caméra haute définition, compatible avec tous les modèles Raspberry Pi, fournit une haute sensibilité, une capture d'image à faible bruit dans un design ultra petit et léger. Le module caméra se connecte à la carte Raspberry Pi via le connecteur Camera Serial Interface (CSI) conçu spécifiquement pour l'interface avec les caméras qui est capable de débits de données extrêmement élevés, et il transporte exclusivement les données de pixels vers le processeur Comme le montre la figure 4.5.

4.2.3.1 Spécifications

- Résolution 8 mégapixels
- Résolution des images fixes 3280 x 2464
- Taux de transfert d'image maximum 1080p : 30fps (codage et décodage)
720p : 60 images par seconde
- Connexion au Raspberry Pi Câble plat 15 broches, à l'interface série dédiée MIPI 15 broches pour caméra
- Balance des blancs automatique
- Filtre de bande automatique
- Détection automatique de la luminance 50/60 Hz



FIGURE 4.5 – Raspberry Pi Camera Module 2

- Calibrage automatique du niveau de noir
- Plage de température de fonctionnement : -20° à 60°
- Image stable : De -20° à 60°
- Taille de l'objectif 1/4"
- Dimensions 23,86 x 25 x 9mm
- Poids 3g .[cam,]

4.2.4 Carte Micro SD

Une carte Micro SD (Figure) est un élément clé du Raspberry Pi, elle fournit le stockage initial pour le système d'exploitation et les fichiers. Il est préférable qu'elle soit de haute capacité (au moins 16 GByte).

4.3 Environnement software

4.3.1 Raspbian

Raspberry Pi OS (précédemment appelé Raspbian) est un système d'exploitation libre basé sur Debian, optimisé pour le matériel Raspberry Pi, et est le système d'exploitation recommandé pour une utilisation normale sur un Raspberry Pi. 35 000 paquets qui sont des logiciels regroupés et précompilés contenu dans le système d'exploitation pour l'installation du Raspberry Pi. [[ras](#),]

Il existe plusieurs versions de Raspberry Pi OS :

- Wheezy (basée sur Debian 7)
- Jessie (basée sur Debian 8)
- Stretch (basée sur Debian 9)
- Buster (debian 10)
- Bullseye (debian 11) [[sys](#),]

4.3.2 Langage Python

Python est un langage de programmation de haut niveau à usage général largement utilisé (voir la figure 4.6). Il a été initialement conçu par Guido van Rossum en 1991 et développé par la Python Software Foundation. Les avantages principaux sont la lisibilité de la syntaxe et du code qui vont permettre aux programmeurs de montrer des idées au moins de lignes de code. [[his](#),]

De nombreuses raisons expliquent la popularité de Python parmi les développeurs,



FIGURE 4.6 – Logo du langage Python

notamment le fait qu'il dispose d'une collection étonnamment vaste de bibliothèques

avec lesquelles les utilisateurs peuvent travailler.

Voici quelques raisons importantes qui expliquent la popularité de Python :

- Python dispose d'une énorme collection de bibliothèques
- Du développement au déploiement et à la maintenance, Python veut que ses développeurs soient plus productifs
- Parce qu'il est simple et facile, il peut être utilisé par les débutants
- La portabilité est une autre raison de la grande popularité de Python
- La syntaxe de programmation de Python est simple à apprendre et est de haut niveau si on la compare à C, Java et C++
- Il suffit donc de quelques lignes de code pour créer de nouvelles applications

La simplicité de Python a incité de nombreux développeurs à concevoir de nouvelles bibliothèques pour le machine learning.[\[lib, \]](#)

4.3.3 Bibliothèques utilisées

4.3.3.1 Math

Le module mathématique est un module standard de Python et est toujours disponible. Pour utiliser les fonctions mathématiques de ce module, On utilise `import math` [\[mat, b\]](#)

4.3.3.2 Numpy

NumPy est le paquetage fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que les tableaux masqués et les matrices), ainsi pour effectuer des opérations rapides sur les tableaux, notamment des opérations mathématiques, logiques, de manipulation de formes, de tri, de sélection, d'E/S, de transformées de Fourier discrètes, d'algèbre linéaire de base, d'opérations statistiques de base, de simulation aléatoire et bien plus encore.[\[num, \]](#)

Caractéristiques de Numpy

- Interactif : Numpy est très interactif et facile à utiliser
- Mathématiques : Rend les opérations mathématiques complexes très simples[\[lib, \]](#)

On utilise `import numpy`

4.3.3.3 Mathplotlib.pyplot

matplotlib.pyplot est une bibliothèque de dessin utilisée pour les graphiques 2D dans le langage de programmation python. Elle peut être utilisée dans les scripts python, les shells, les serveurs d'applications web et autres boîtes à outils d'interface utilisateur graphique. Matplotlib ne fait pas partie des bibliothèques standard qui sont installées par défaut dans Python. Il existe plusieurs boîtes à outils qui étendent les fonctionnalités de Matplotlib dans Python. Certaines d'entre elles sont téléchargées séparément, d'autres peuvent être livrées avec le code source de matplotlib mais ont des dépendances externes.[\[mat, a\]](#)

Plusieurs graphiques sont conçus par python matplotlib. Certains d'entre eux sont énumérés dans la figure 4.7

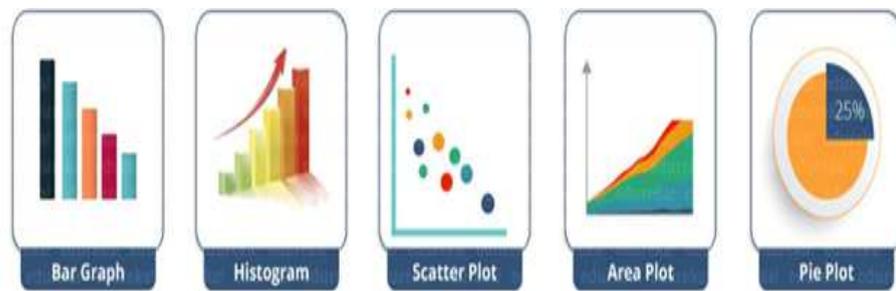


FIGURE 4.7 – Types de tracés utilisés par Matplotlib
[\[mat, a\]](#)

4.3.3.4 OpenCV

Open Source Computer Vision Library (OpenCV) est une bibliothèque logicielle de vision par ordinateur et d'apprentissage automatique (voir la figure 4.8).

OpenCV a été construit pour fournir une infrastructure commune pour les applications de vision par ordinateur et pour accélérer l'utilisation de la perception artificielle dans les produits commerciaux.

OpenCV est publié sous une licence BSD (un type de licence à faible restriction pour les logiciels libres qui n'impose pas de conditions de redistribution).

Il possède des interfaces C++, C, Python et Java et supporte Windows, Linux, Mac OS, iOS et Android.

OpenCV a été conçu pour l'efficacité des calculs et avec un fort accent sur les applications en temps réel. La bibliothèque compte plus de 2500 algorithmes optimisés, ce

qui inclut un ensemble complet d'algorithmes de vision par ordinateur et d'apprentissage automatique classiques et de pointe. Ces algorithmes peuvent être utilisés pour :

- Détection et identification des objets et visages
- Classification du comportement humain en vidéo
- Suivi du mouvement de caméra et d'objets
- Assemblage d'images pour générer une image haute résolution de toute la scène
- Décèlement des images similaires dans une base de données d'images
- Suppression des yeux rouges des images prises au flash
- Suivi des mouvements des yeux,
- Reconnaissance des paysages
- établissement des marqueurs en vue de leur superposition à la réalité augmentée, etc.[[ope](#), [b](#)]



FIGURE 4.8 – Logo de la bibliothèque OpenCV

Fonctionnalités

La bibliothèque OpenCV fournit de nombreuses fonctions très diverses, permettant de créer des programmes à partir de données brutes, jusqu'à la création d'interfaces graphiques de base.

Traitement d'images

Elle propose la plupart des opérations classiques en traitement bas niveau des images :

- Lecture, écriture et affichage d'une image
- Calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs
- Lissage, filtrage
- Seuillage d'image (méthode d'Otsu, seuillage adaptatif)
- Segmentation (composantes connexes, GrabCut)
- Morphologie mathématique

Traitement vidéos

La bibliothèque s'est imposée comme une référence dans le domaine de la recherche, car elle fournit un grand nombre d'outils de pointe en vision par ordinateur, tels que :

- Lecture, écriture et affichage d'une vidéo
- Détection de mouvement
- Détection des lignes
- Détection des visages[[ope](#), [a](#)]

4.3.3.5 TensorFlow

TensorFlow est une bibliothèque open-source développée par Google principalement pour les applications d'apprentissage profond. Elle prend également en charge l'apprentissage automatique traditionnel. TensorFlow a été développé à l'origine pour les grands calculs numériques, sans penser à l'apprentissage profond. Cependant, il s'est avéré très utile pour le développement de l'apprentissage profond, et c'est pourquoi Google l'a mis en libre accès.

TensorFlow accepte les données sous la forme de tableaux multidimensionnels de dimensions supérieures appelés tenseurs. Les tableaux multidimensionnels sont très pratiques pour traiter de grandes quantités de données, il fonctionne sur la base de graphes de flux de données comportant des nœuds et des arêtes.[[ten](#),]

4.3.3.6 Keras

Keras est considéré comme l'une des bibliothèques d'apprentissage automatique les plus cool de Python. Elle fournit un mécanisme plus facile pour exprimer les

réseaux neuronaux. Keras fournit également certains des meilleurs utilitaires pour la compilation des modèles, le traitement des ensembles de données, la visualisation des graphiques, et bien plus encore.

En arrière-plan, Keras utilise Theano ou TensorFlow en interne. Keras est comparativement lent lorsque nous le comparons à d'autres bibliothèques d'apprentissage automatique. En effet, il crée un graphe de calcul en utilisant infrastructure back-end, puis l'utilise pour effectuer des opérations.

Caractéristiques de Keras

- Il fonctionne sans problème à la fois sur le CPU et le GPU
- Keras prend en charge presque tous les modèles d'un réseau neuronal - entièrement connecté, convolutif, pooling, récurrent, embedding, etc. En outre, ces modèles peuvent être combinés pour construire des modèles plus complexes.
- Keras par sa nature modulaire est incroyablement expressif, flexible et adapté à la recherche innovante.[\[lib, \]](#)

4.3.3.7 Time

Le module time de Python est utilisé pour calculer le temps écoulé en secondes. En utilisant le module temps, il est possible de calculer le temps d'exécution d'une partie du code.[\[tim, \]](#) En utilisant la syntaxe import time

4.3.4 L'IDE Thonny

An integrated development environment (IDE) est un outil de développement open-source pour la programmation python qui est utilisé pour écrire et télécharger des programmes sur différentes cartes de développement telles que Raspberry Pi.

Voici quelques-unes des principales caractéristiques de Thonny :

- Il auto complète le code
- Il inspecte le code pour fournir une correspondance entre les parenthèses et mettre en évidence les erreurs
- Il est facile à utiliser car son installateur installe également Python 3.7
- Son débogueur est simple à utiliser car aucune connaissance des points d'arrêt n'est requise

- Il permet aux utilisateurs d'entrer dans un appel de fonction en fournissant des détails sur les variables locales et en affichant le pointeur de code
- Il dispose d'une interface facile pour installer des paquets. Il convient donc parfaitement aux débutants
- Il est préinstallé dans Raspian OS, un système d'exploitation pour Raspberry Pi. Il est disponible pour être installé sur Windows, Linux et Mac OS [tho,]

4.3.5 VNC Viewer

VNC est un système de partage de bureau graphique qui vous permet de contrôler à distance l'interface de bureau d'un ordinateur (exécutant VNC Server) depuis un autre ordinateur ou un appareil mobile (exécutant VNC Viewer) : VNC Viewer transmet les événements du clavier et de la souris ou du toucher à VNC Server, et reçoit en retour des mises à jour de l'écran.

Vous verrez le bureau du Raspberry Pi dans une fenêtre sur votre ordinateur ou votre appareil mobile. Vous pourrez le contrôler comme si vous travailliez sur le Raspberry Pi lui-même.

VNC Connect de RealVNC est inclus dans Raspberry Pi OS. Il comprend à la fois VNC Server, qui vous permet de contrôler votre Raspberry Pi à distance, et VNC Viewer, qui vous permet de contrôler les ordinateurs de bureau à distance depuis votre Raspberry Pi si vous le souhaitez.

4.4 Conclusion

Dans ce chapitre, Nous avons présenté la Raspberry Pi4. Nous avons donné un petit historique et les différents modèles de Raspberry pi. Puis, Nous avons donné les avantages de notre modèle pi4 par rapport aux autres modèles. Par la suite, nous avons présenté tout le matériel utilisé. Enfin, la présentation de tout l'environnement software, logiciels et la définition des différentes bibliothèques utilisés.

Chapitre 5

Implémentation su Raspberry Pi 4

Sommaire

5.1	Introduction	66
5.2	Configuration matérielle	66
5.2.1	Téléchargement de l'image de distribution	66
5.2.2	Enregistrement du contenu de l'image sur la carte SD	67
5.2.3	Configuration initiale	67
5.3	Conception du système	70
5.3.1	Détection de visage	71
5.3.2	Détection de masque	72
5.3.3	Mesure de distanciation	73
5.4	Conclusion	73

5.1 Introduction

Jusqu'à présent, nous nous sommes attachés à décrire le cadre théorique ainsi que le cadre de programmation de notre travail de projet. La matérialisation de notre système et les résultats que nous avons obtenus constituent le cœur de ce dernier chapitre.

Nous avons choisi de diviser ce chapitre en deux parties : la présentation et la configuration matérielle et l'implémentation sur le raspberry Pi 4.

5.2 Configuration matérielle

Dans ce projet, nous avons opté pour un Raspberry pi 4, afin d'essayer de réduire le prix, le niveau de complexité et de rendre possible les fonctionnalités de vidéo surveillance. Compte tenu de ses caractéristique, que nous avons mentionnées précédemment, elles se résument comme suit :

- posséder des dimensions adéquates pour l'installation de dispositifs d'entrée et de sortie
- une faible consommation d'énergie
- pourrait être relié à un écran faible coût
- facilement accessible sur le marché

Le Raspberry Pi 4 utilise une carte mémoire microSD pour démarrer et charger le système d'exploitation. Dans ce qui suit, nous allons décrire les étapes à suivre pour installer et mettre à jour le système d'exploitation.

5.2.1 Téléchargement de l'image de distribution

L'image de la distribution Raspberry Pi OS, obtenue sur le site web officiel¹ du fabricant Raspberry Pi, est d'abord téléchargée librement sur notre ordinateur. Deux versions de la distribution du système d'exploitation Raspberry Pi existent : Raspberry Pi OS (64-bit) avec bureau et Raspberry Pi OS (64-bit) Lite. Nous avons choisi de télécharger la première version.

1. <https://www.raspberrypi.com/software/operating-systems/>

5.2.2 Enregistrement du contenu de l'image sur la carte SD

Pour enregistrer le contenu de l'image sur la carte SD, nous avons besoin d'un logiciel spécifiquement créé pour cette tâche. L'application Raspberry Pi Imager est utilisée pour garantir une adéquation optimale.

En activant le "bouton" CHOOSE OS, nous avons indiqué le dossier où se trouvait l'image du système d'exploitation du Raspberry Pi et plus précisément le fichier en question. Grâce au "bouton" CHOOSE STORAGE, nous avons indiqué l'unité où se trouvait la carte SD que nous voulions utiliser. Enfin, sur le "bouton" WRITE, nous avons activé la poursuite de la tâche. À la fin de ce processus, un message apparaît indiquant que le processus a été effectué correctement et que la carte SD peut être éjectée et insérée dans son emplacement prévu à cet effet dans le Raspberry Pi.

5.2.3 Configuration initiale

Par défaut, le nom d'hôte du Raspberry Pi est raspberrypi. Ce nom d'hôte sera visible sur le réseau. Si on dispose de deux ou plusieurs appareils Raspberry Pi, Il faudrait peut-être modifier le nom d'hôte par défaut afin de différencier les appareils et d'éviter les conflits de noms.

Comme première activité, nous avons renommé notre Raspberry Pi. Pour ce faire, il faut comme l'indique la figure 5.1 :

1. Utiliser CTRL+ALT+T pour vous connecter au terminal
2. Exécuter "sudo raspi-config"
3. Sélectionner "System Options"
4. Sélectionner Hostname et appuyez sur Entrée ou sélectionnez Ok si vous recevez un avertissement vous indiquant de ne pas utiliser de caractères spéciaux dans le nom d'hôte du Raspberry Pi.
5. Entrez le nouveau nom d'hôte et appuyez sur ok

Notre Raspberry Pi4 porte le nom InstrumentationUAT2022

Afin de protéger le Raspberry Pi, SSH est installé par défaut, mais est désactivé pour des raisons de sécurité. Sur notre Raspberry Pi, la seconde étape consiste à activer SSH comme suit (voir la figure 5.2).

1. Utiliser CTRL+ALT+T pour vous connecter au terminal
2. Exécuter "sudo raspi-config"

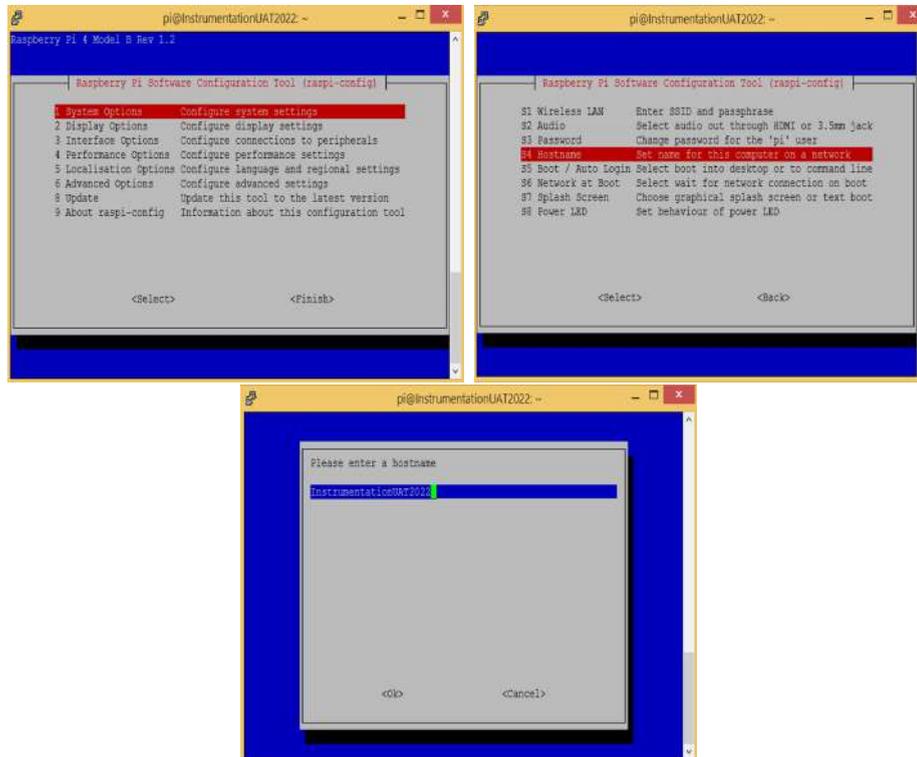


FIGURE 5.1 – Modification du nom d'hôte

3. Sélectionner "Interface Options" Choisissir "SSH" dans le menu déroulant.
4. Cliquer sur "Enable" pour activer la fonctionnalité.
5. Le bouton "Ok" doit être utilisé pour confirmer le message.
6. Le bouton "Finish" permet de terminer la configuration

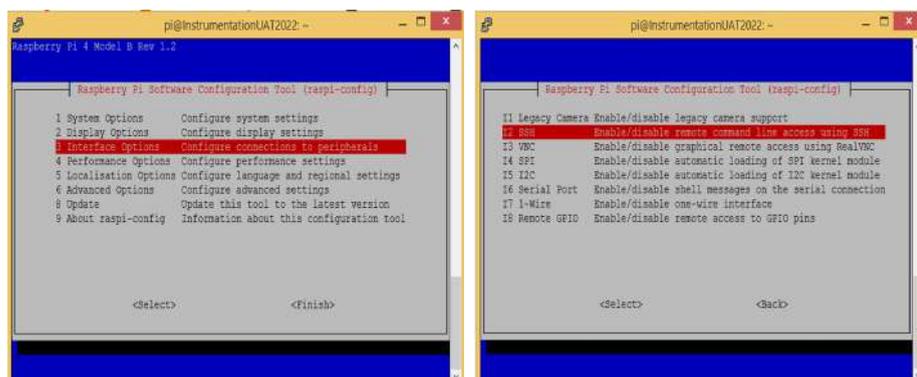


FIGURE 5.2 – Activation du SSH

Il est nécessaire de télécharger et d'exécuter le logiciel client SSH sur notre ordinateur après l'avoir activé sur notre Raspberry Pi. Comme notre ordinateur fonctionne sous Windows, la solution suggérée est de télécharger et d'installer PuTTY. PuTTY est un logiciel open source créé et maintenu par une communauté d'utilisateurs. PuTTY peut être téléchargé à partir de cette page².

La particularité de VNC est de permettre de contrôler un ordinateur distant tout en visualisant son bureau. On peut désormais regarder ce qui se passe sur notre Raspberry Pi en temps réel sans avoir à le connecter à un écran. Comme le montre la figure 5.3, l'activation de VNC se fait de la même façon que le SSH sauf en choisissant VNC dans le menu "Interface Options".

VNC peut être téléchargé à partir de cette page³.

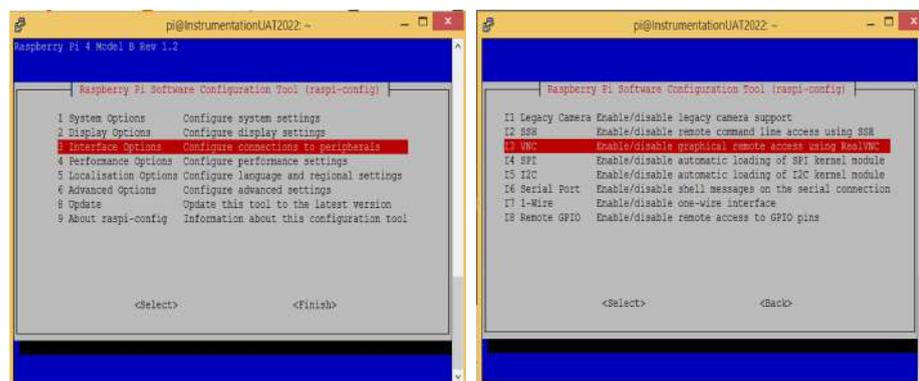


FIGURE 5.3 – Activation du VNC

Le dernière étape consiste à activer la Raspicam comme suit (voir la figure 5.4).

1. Utiliser CTRL+ALT+T pour vous connecter au terminal
2. Exécuter "sudo raspi-config"
3. Sélectionner "Interface Options"
4. Choisir "Legacy Camera" dans le menu déroulant.
5. Cliquer sur "Enable" pour activer la fonctionnalité.
6. Le bouton "Ok" doit être utilisé pour confirmer le message.
7. Le bouton "Finish" permet de terminer la configuration

2. <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

3. <https://www.realvnc.com/fr/connect/download/vnc/>

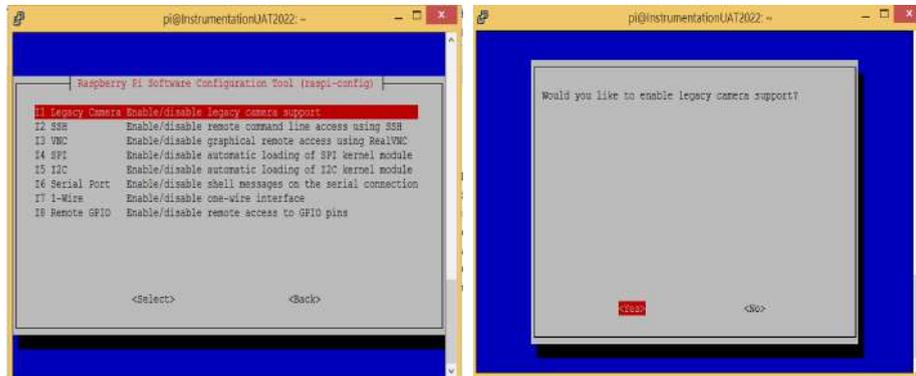


FIGURE 5.4 – Activation de le caméra

5.3 Conception du système

Le Raspberry Pi 4 et la caméra constituent la base du hardware proposé. La conception d'un système de vidéo-surveillance basé sur la détection du visage, des masques et de la mesure de distanciation sociale sur le Raspberry Pi 4 peut être illustré dans la figure 5.5



FIGURE 5.5 – Prototype de vidéo-surveillance proposé

Une vue de près du prototype est présentée dans la figure 5.6.

Le système proposé fonctionne en trois étapes. Le système capture une image et vérifie la présence d'un visage dans l'image. Les visages détectés sont recadrés puis donnés à un modèle pour vérifier si les personnes portent un masque ou pas.



FIGURE 5.6 – Vue de près du prototype de vidéo-surveillance proposé

Nous commençons le programme en important les modules spécifiés. Le module `cv2` est déployé pour le traitement des images et `Numpy` est utilisé pour convertir les images en équivalents mathématiques. Les modèles de détection du visage et du masque sont également transférés du cloud Colab vers le Raspberry Pi 4. Un code nommé `face_mask.py` est créé, comme illustré dans la figure 5.7

```
1 import numpy as np
2 import cv2
3 from tensorflow.keras.preprocessing.image import img_to_array
4 from keras.models import load_model
5 from math import pow
6 import time
7 model = load_model('mask_recog.h5')
```

FIGURE 5.7 – Code de détection du visage et du masque

5.3.1 Détection de visage

Dans la figure 5.8, quelques résultats de la détection sont affichés. Comme on peut le voir, le détecteur SSD donne d'excellents résultats même avec une occlusion

(lunettes).

L'un des principaux défis rencontrés par les techniques actuelles de reconnaissance des visages réside dans les difficultés à gérer des poses différentes. La dernière ligne de la figure 5.8, la détection avec un visage profil est obtenue grâce au SSD. Une telle situation est généralement rencontrée dans systèmes de vidéo-surveillance traitant de sujets non coopératifs.

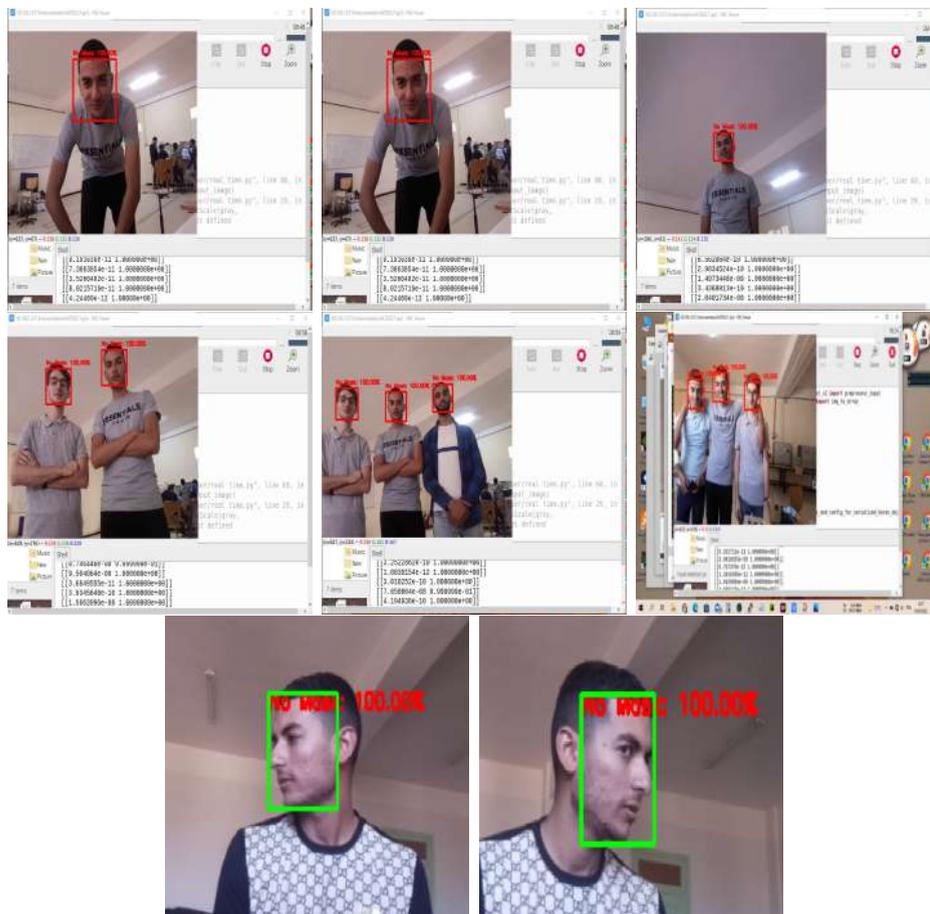


FIGURE 5.8 – Quelques résultats de la détection du visage

5.3.2 Détection de masque

Le modèle de détection de masque a été testé sur diverses images/vidéos en temps réel de personnes portant un masque et sans masque. Le modèle a été capable de détecter avec succès des visages avec et sans masque sur les images suivantes de la figure 5.9.

Le modèle basé sur le CNN a donné d'excellents résultats sur des visages portant des masques colorés avec des variations de pose.

5.3.3 Mesure de distanciation

Les distances entre les centres d'un visage détecté et un autre visage est mesurée ; si la distance est dépassée, les personnes sont dites en violation des mesures de sécurité.

La figure 5.10 montre un exemple de notre représentation des visages détectés dans une scène ainsi que de multiples cas de violation de la distance sociale.

5.4 Conclusion

Dans ce chapitre, nous avons proposé un système automatisé pour couvrir plusieurs aspects pertinents de la vidéo-surveillance tels que la détection de visages, la détection automatique de masques et la mesure de distanciations.

Notre système peut prendre en entrée des images de la vie réelle prises à partir d'une Raspicam. Ensuite, il peut déterminer si la personne sur l'image porte un masque ou non. Une mesure de distanciation peut être faite pour détecter le nombre de violation des mesures sanitaires.

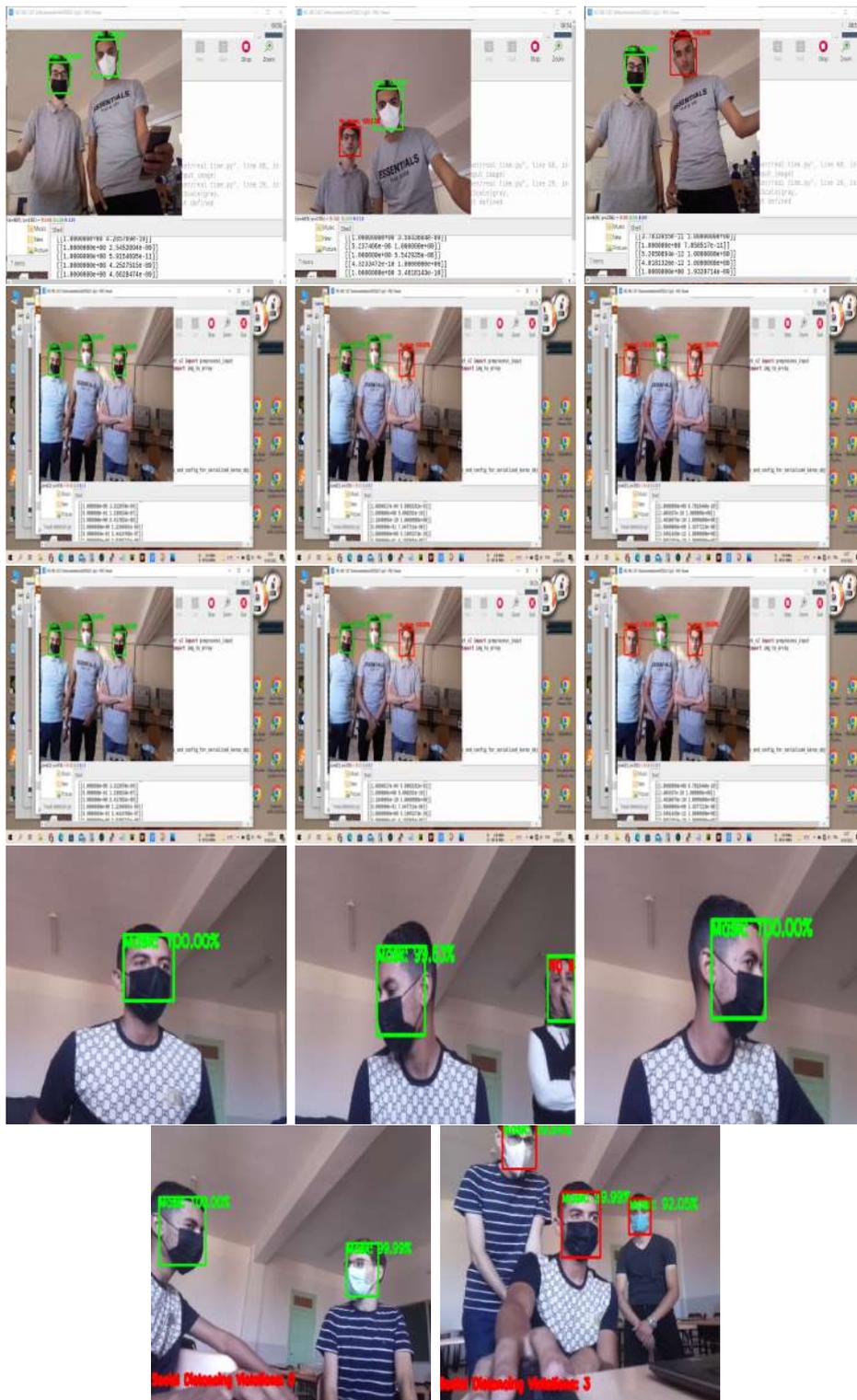


FIGURE 5.9 – Quelques résultats de la détection du masque

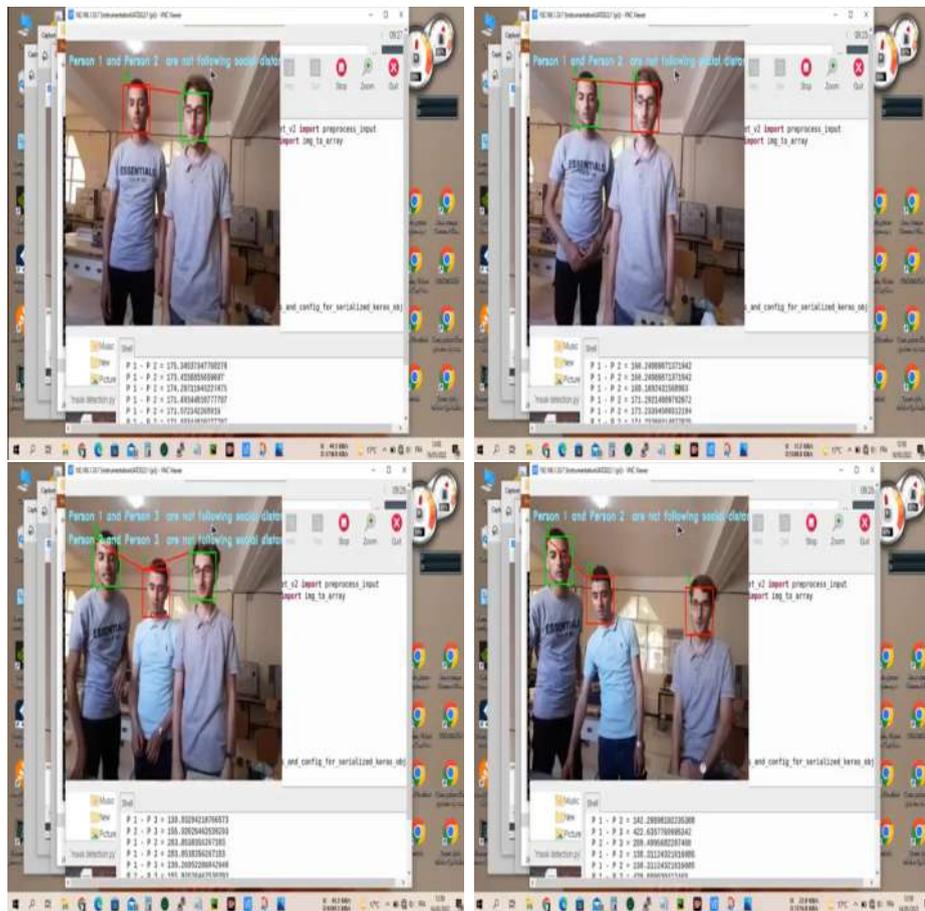


FIGURE 5.10 – Quelques résultats de la mesure de distanciation

Conclusion Générale

La transmission universelle de la pandémie de COVID-19 (Coronavirus) entraîne un besoin immédiat d'engagement dans la lutte à travers toute la population humaine. Les urgences en matière de santé humaine sont limitées pour cette épidémie brutale et cet environnement abandonné. Dans cette situation, l'automatisation inventive comme la computer vision (machine learning, Deep Learning, intelligence artificielle), a développé des solutions encourageantes contre le COVID-19. Ces derniers mois, différentes techniques utilisant le traitement d'image sont réalisées par divers chercheurs. L'IA a également contribué à la lutte contre la pandémie de coronavirus (COVID-19), qui sévit dans le monde entier. dans ce travail nous avons vu qu'il sera important de disposer d'outils permettant d'identifier rapidement les personnes les plus exposées au risque de morbidité et celles qui ne se soucient pas des mesures de précaution, c'est la raison pour laquelle nous avons développé le système susmentionné, Ce dernier est un système de surveillance intelligent permet de traite les personnes qui respectent les mesures de précaution (la distanciation et le masque) et celles qui ne le font pas. Nous proposant deux nouveaux modèle : Le premier pour la détection de visage, et le second pour la détection de masque, ces modèles sont réalisé à l'aide des algorithmes de classification d'images de vision par ordinateur CNN basé sur le Deep Learning

Bibliographie

- [tho,] getting started with raspberry pi pico using thonny ide. <https://microdigisoft.com/getting-started-with-raspberry-pi-pico-using-thonny-ide/>.
- [his,] history of python. <https://www.geeksforgeeks.org/history-of-python/>.
- [mat, a] matplotlib tutorial. <https://www.edureka.co/blog/python-matplotlib-tutorial/#matplotlib>.
- [mat, b] modules math. <https://www.programiz.com/python-programming/modules/math>.
- [num,] Numpy. <https://numpy.org/doc/stable/>.
- [ope, a] opencv. <https://fr.wikipedia.org/wiki/OpenCV>.
- [ope, b] opencv about. <https://opencv.org/about/>.
- [lib,] python librairies. <https://www.edureka.co/blog/python-libraries/>.
- [sys,] raspberry guide. <https://www.raspberrypi-france.fr/guide/systeme-exploitation-raspberry-pi/>.
- [cam,] Raspberry pi camera module. <https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/RPiCamMod2.pdf>.
- [ras,] raspbian systeme. <https://www.raspbian.org/>.
- [FPS,] What does fps mean? <https://clipchamp.com/en/definition/what-does-fps-mean-frame-rates-explained/>.
- [ten,] what is tensorflow. <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflow>.
- [tim,] what is the python time module. <https://www.educative.io/edpresso/what-is-the-python-time-module>.

- [Albayati et al., 2021] Albayati, A. Q., Alkhalid, F. F., and Hussain, R. K. (2021). Online multi-layers social distance detection in iraqi schools. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(6) :2103–2108.
- [Bishop and Nasrabadi, 2006] Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.
- [Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3) :273–297.
- [Dennis, 2016] Dennis, A. K. (2016). *Raspberry pi computer architecture essentials*. Packt Publishing Ltd.
- [Editor CNN, 2011] Editor CNN (2011). Illinois Supreme Court keeps Emanuel on ballot. <http://edition.cnn.com/2011/POLITICS/01/27/emanuel.ballot/>. Last checked on Nov 01, 2013.
- [Finger, 2001] Finger, S. (2001). *Origins of neuroscience : a history of explorations into brain function*. Oxford University Press, USA.
- [Fix and Hodges, 1951] Fix, E. and Hodges, J. L. (1951). Nonparametric discrimination : consistency properties. *Randolph Field, Texas, Project*, pages 21–49.
- [He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [Howard et al., 2017] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*.
- [Jesorsky et al., 2001] Jesorsky, O., Kirchberg, K. J., and Frischholz, R. W. (2001). Robust face detection using the hausdorff distance. In *International conference on audio-and video-based biometric person authentication*, pages 90–95. Springer.
- [Kovac et al., 2003] Kovac, J., Peer, P., and Solina, F. (2003). *Human skin color clustering for face detection*, volume 2. IEEE.
- [Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

- [LeCun et al., 1995] LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10) :1995.
- [LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4) :541–551.
- [Liu et al., 2016] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd : Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.
- [Loey et al., 2021] Loey, M., Manogaran, G., Taha, M. H. N., and Khalifa, N. E. M. (2021). A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the covid-19 pandemic. *Measurement*, 167 :108288.
- [Pi, 2019] Pi, R. (2019). Raspberry pi 4 computer-model b : Product brief. *Raspberry Pi Trading Ltd., Cambridge*.
- [Saini et al., 2019] Saini, R., Prakash, C., and Dua, A. (2019). Current trends in information technology a survey on artificial intelligence techniques for cybersecurity. 9 :5–13.
- [Scherer et al., 2010] Scherer, D., Müller, A., and Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. In *International conference on artificial neural networks*, pages 92–101. Springer.
- [Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*.
- [Thompson, 1981] Thompson, M. (1981). Digital image processing by rafael c. gonzalez and paul wintz. *Leonardo*, 14(3) :256–257.
- [Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee.
- [Wang et al., 2021] Wang, Z., Wang, P., Louis, P. C., Wheless, L. E., and Huo, Y. (2021). Wearmask : Fast in-browser face mask detection with serverless edge computing for covid-19. *arXiv preprint arXiv :2101.00784*.

