République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Faculté des Sciences Exactes

Département d'Informatique

Mémoire de Fin de Cycle
En vue de l'obtention du diplôme de Master en Informatique
Option : Réseaux et Sécurité

Thème :

Détection d'intrusion et contrôle d'accès dans les réseaux IoT

Réalisé par:

M^elle **BAZIZI Anya**

Encadré par:

YAZID Mohand	Encadrant	Professeur	Université de Béjaia.
MAMMERI Souhila	Encadrant	M.C.B	Université de Béjaia.

Soutenu devant le jury composé de:

MOKETFI Mohand	Président	M.C.B	Université de Béjaia.
HOCINI Kenza	Examinateur	M.A.B	Université de Béjaia.

Promotion 2023 – 2024

Remerciements

En tout premier lieu, je souhaite adresser mes remerciements les plus sincères ainsi qu'une immense gratitude à mon encadrante Mme MAMMERI, pour son précieux soutien, pour ses conseils avisés, pour sa présence tout au long de ce mémoire ainsi que tout ce qu'elle a fait pour moi, c'est grâce à elle si ce mémoire a abouti. Je tiens également à chalereusement remercier mon encadrant Mr YAZID pour ses sages conseils ainsi que son soutien tout au long de ce projet de fin d'études. Je suis reconnaissance du temps qu'ils ont consacré afin de m'aider à réaliser ce mémoire.

Ensuite, je souhaite remercier le président du jury Mr MOKTEFI ainsi que l'examinatrice Mme HOCINI pour leur présence ainsi que leur évaluation de notre travail.

Un énorme merci à mes très chers parents, à ma très chère petite soeur, ils sont ma motivation et ma plus grande fierté.

Dédicaces

Je dédie mon mémoire à ma très précieuse et regrettée grand-mère Ndjima qui nous a malheureusement quittés ce mois de mai. Tu resteras à jamais dans mon cœur, toi qui m'as tout appris. J'espère que tu es fière de moi et j'espère être à la hauteur des attentes que tu avais de moi. Je ne t'oublierai jamais.

À mes parents Abdelhafid et Nassima, qui ont toujours cru en moi et à qui je dois tout. Vous avez toujours été mes piliers dans la vie, mon moteur. Je vous serai à jamais reconnaissante pour tous les sacrifices que vous avez faits pour moi afin que je puisse toujours me consacrer à mes études.

À ma petite sœur Mélissa, que j'affectionne tout particulièrement et pour qui je souhaite être un modèle.

À mes adorables tantes Chafika et Nabila, que je considère comme mes deuxièmes mamans.

À mes oncles Nassim, Nabil et Moumen, qui m'ont vue grandir et sur qui j'ai toujours su compter.

À mes sœurs de cœur Taous, Célia, Lydia, Imène,et Mina sur qui je peux toujours m'appuyer,

À mes cousines que j'affectionne énormément, en particulier Chanez, ma confidente et ma grande soeur,

À mes cousins que je considère comme mes petits frères,

Avec amour et gratitude,
Anya.

Table des matières

Liste des tableaux

Liste des équations

Liste des figures

Liste des abréviations

CEI	**C**ommision **E**léctronique **I**nternationale
CNN	**C**onvolutionnal **N**eural **N**etwork
CSV	**C**oma **S**eparated **V**alues
CTJ1	**C**ommité **T**echnique **J**oint 1
FBI	**F**ederal **B**ureau of **I**nvestigation
GPU	**G**raphic **P**rocessing **U**nit
GUI	**G**raphical **U**ser **I**nterface
HDR	**H**igh **D**ynamic **R**ange
HD	**H**igh **D**efinition
HTTPS	**H**yper **T**ext **T**ransfer **S**ecure
IA	**I**ntelligence **A**rtificielle
IoT	**I**nternet **o**f **T**hings
IP	**I**nternet **P**rotocol
kNN	**k-N**earest **N**eighbor algorithm
LAN	**L**ocal **A**rea **N**etwork
LBP	**L**ocal **B**inary **P**attern algorithm
LBPH	**L**ocal **B**inary **P**attern **H**istograms
LDA	**L**inear **D**iscriminant **A**nalysis
LoRaWAN	**Lo**w **Ra**nge **W**ide **A**ea **N**etwork
LTE	**L**ong **T**erm **E**volution
ML	**M**achine **L**earning
NFC	**N**ear **F**ield **C**ommunication
NumPy	**N**umerical **P**ython
OCR	**O**ptical **C**haracter **R**ecognition
OpenCV	**O**pen **C**omputer **V**ision library
OS	**O**perating **S**ystem
OSI	**O**rganisation **I**nternationale de **S**tandardisation
PCA	**P**rincipal **C**omponent **A**nalysis
RFID	**R**adio **F**requency **ID**entification
RMSE	**R**oot **M**ean **S**quare **E**rror
SciPy	**S**cientific **P**ython
SIFT	**S**hift-**I**nvariant **eature** **T**ransform
SQL	**S**tructured **Q**uery **L**anguage

SSL	**S**ecure **S**ockets **L**ayer
SURF	**S**peed-**U**p **R**obust **F**eatures
SVM	**S**upport **M**achine **V**ector
TIC	**T**echniques de l'**I**nformation et de la **C**ommunication
TLS	**T**ransport **L**ayer **S**ecurity
UIT	**U**nion **I**nternationale des **T**élécommunications
Wi-Fi	**Wi**reless **Fi**delity
WSN	**W**ireless **S**ensor **N**etwork

Introduction Générale

La planète intelligente est un monde dans lequel des objets connectés à Internet, sont capables de communiquer avec des humains et intéragir intelligemment avec d'autres objets, c'est l'IoT (Internet Of Things). Une maison intelligente ou smart home est l'un des domaines d'application de l'IoT. L'IoT a été rendu possible grâce à deux principaux facteurs, à savoir la variété des technologies de communication (Wi-Fi, 4G/5G, LoRa, etc) et la disponibilité de plateformes de Cloud, d'intelligence artificielle, etc. Parmi les caractéristiques de l'IoT, nous citons le nombre énorme d'objets, les données sensibles collectées des utilisateurs, la grande quantité de données ou Big data, les ressources restreintes des objets, le changement dynamique dans l'environnement des utilisateurs, le traitement en temps réel pour certaines applications, etc.

Avec l'IoT, les objets physiques deviennent intelligents grâce à l'intégration de capteurs et d'actionneurs, de logiciels et d'autres technologies qui permettent l'échange de données en temps réel via Internet. Le réseau connecté ouvre la voie à des perspectives innovantes dans divers secteurs, qu'il s'agisse de la maison intelligente, de la santé intelligente, de l'agriculture intelligente, de l'industrie intelligente, etc. Cependant, afin de rendre un système intelligent, les fonctionnalités de la portabilité, l'interopérabilité, la mobilité et notamment la sécurité doivent être prises en compte. La sécurité dans l'IoT ne se résume pas à prémunir contre la fuite des données des utilisateurs, mais au contrôle d'accès des utilisateurs, leur sécurité physique et notamment la détection d'intrusions.

Dans ce mémoire, nous abordons la sécurité d'une personne dans une maison intelligente, nous nous inétressions principalement à la vidéosurveillance et son mode de fonctionnement ainsi que les diverses approches utilisées afin de détecter une intrusion. On cite par exemple la détection de mouvement, la détection de la silhouette et la détection du visage et donc la reconnaissance faciale sur laquelle nous basons notre travail. L'utilisation de la reconnaissance faciale avec des caméras de surveillance ou de sécurité et des systèmes de contrôle d'accès garantit la sécurité dans les espaces publics ou privés en identifiant rapidement les personnes grâce à la surveillance, permettant ainsi de réagir de manière appropriée. L'IoT favorise ainsi une automatisation accrue et une prise de décision rapide, basée sur des données précises, ce qui améliore l'efficacité, la productivité et la qualité de vie.

L'objectif principal de ce projet est d'implémenter deux algorithmes de Machine Learning afin qu'ils puissent faire de la reconnaissance faciale et qu'ils puissent faire la différence entre une personne autorisée et un intrus, nous voulons également proposer une solution de système de

détection d'intrusion dans une maison intelligente qui utiliserait les algorithmes que nous avons implémentés. Ce système intervient lorsqu'un habitant (une personne autorisée) est absente, elle recevra donc un message d'alerte qui lui notifiera qu'une intrusion a été détectée.

Nous allons tout d'abord pré-traiter les données du Dataset avec l'algorithme de réduction de dimensionnalité PCA (Principal Component Analysis) puis nous allons décider de comparer entre deux algorithmes supervisés SVM (Support Machine Vector) et LDA (LinearDiscriminant Analysis) afin de savoir quel est le meilleur algorithme à utiliser en terme de précision dans le cas d'un système de détection d'intrusion dans une maison intelligente.

Ce mémoire est constitué d'une introduction générale, de trois chapitres et d'une conclusion générale avec perspectives. Les chapitres sont structurés comme suit :

- Le chapitre 1 est intitulé **Les réseaux IoT et la sécurité** : Dans ce chapitre, nous allons définir le paradigme IoT de manière générale, les objets connectés et l'architecture IoT de manière détaillée avec les différents domaines d'application, et enfin la sécurité dans ce contexte.

- Le chapitre 2 est intitulé **Vidéosurveillance dans les maisons intelligentes** : Dans ce chapitre, on choisit les maisons intelligentes comme contexte d'étude et on définit la vidéosurveillance comme moyen de détection de personnes physiques, puis on réalise un état de l'art sur la détection d'intrusion dans une maison intelligente par moyen de vidéosurvaillance.

- Le chapitre 3 est intitulé **Implémentation d'algorithmes de Machine Learning pour la détection d'intrusion dans une maison intelligente** : Dans ce chapitre, nous allons, tout d'abord, définir les techniques d'intelligence artificielle et les algorithmes de Machines Learning, notamment PCA, SVM et LDA. Puis, nous allons proposer une approche de détection d'intrusion utilisant la vidéosurveillance, par reconnaissance faciale, dans le cas d'une maison intelligente. Ensuite, nous allons évaluer et analyser les performances de l'approche en fonction du nombre de caractristiques des images capturées depuis des vidéos et les différentes dimensions des images prises en considération.

Chapitre 1

Les réseaux IoT et la sécurité

1.1 Introduction

Depuis à présent quelques années, nous assistons à l'essor fulgurant de l'Internet des Objets (IoT). Ce nouveau paradigme représente une révolution d'Internet qui peut connecter les objets de notre quotidien à Internet et entre eux. Il est considéré comme une innovation technologique majeure dans l'industrie des nouvelles technologies de l'information et de la communication. L'IoT n'a pas une définition unique mais d'une manière générale, il est défini comme étant une extension de l'Internet actuel à tous les objets pouvant communiquer de manière directe ou indirecte avec des éuipements électroniques, eux-même connectés à Internet [44].

Grâce à cette interconnexion, l'IoT offre un potentiel énorme dans de nombreux domaines, tels que la santé, la domotique, l'industrie, et bien d'autres encore. Par exemple, des dispositifs de sécurité connectés tels que les caméras de surveillance peuvent surveiller l'intrusion dans les domiciles intelligents et envoyer des alertes en temps réel en cas d'intrusion.

Dans ce chapitre, nous allons aborder les notions élémentaires de l'IoT, la définition des objets connectés, l'architecture IoT ainsi que la sécurité dans le contexte de l'IoT.

1.2 Un aperçu sur l'IoT

1.2.1 Définition

L'Internet des objets ou l'IoT est l'acronyme de "Internet Of Things" en anglais, introduit en 1999 pour la première fois par K.Ashton.

Les différentes visions et technologies de l'IoT rendent sa définition universelle une tâche difficile. L'Union internationale des télécommunications (UIT) à travers son groupe de travail SG20, a défini l'IoT comme étant "une infrastructure mondiale qui interconnecte des dispositifs physiques et virtuels en utilisant les technologies de l'information et de la communication et qui permet d'offrir des services évolués" (ITU 2012) [48].

L'Organisation internationale de standardisation (ISO) et la commission éléctrotechnique internationale (CEI) ont, à travers le comité technique joint 1 (CTJ1), défini l'IoT comment "une intfrastructure d'entités (dispositifs, personnes, systèmes) et des ressources d'informations interconnectées ainsi que des services intelligents leur permettant de traiter les informations du

monde physique et virtuel et de réagir" (ISO/IEC JTC 1 2014) [48].

La figure 1.1 ci-dessous montre la fléxibilité de l'IoT, qu'on peut l'utiliser partout, à tout moment.

Figure 1.1: L'IoT

1.2.2 Les objets connectés

La définition d'un objet connecté, également appelé "smart-device" est sujet à controverse. Dans un premier temps, il est important de replacer le contexte d'utilisation. La communauté scientifique propose plusieurs définitions, ainsi Dorsemaine et al. [29] ont défini les objets connectés comme ce qui suit :

"Les objets connectés sont des capteurs et/ou actionneurs exerçant une fonction spécifique avec des capacités de communication vers d'autres équipements. Ils appartiennent à une infrastructure permettant le transport, le stockage, le traitement et l'accès aux données générées par les utilisateurs ou d'autres systèmes" [29]

Tandis que Zhang et al [51] proposent une définition similaire en affirmant que les objets connectés sont des

"objets physiques ou virtuels qui se connectent à l'internet et qui ont la capacité de communiquer avec des utilisateurs humains ou d'autres objets" [51]

Dans leur définition, les auteurs introduisent la possibilité qu'un objet connecté soit virtuel. Cela impliquerait alors que ce qui peut être considéré comme un seul objet physique soit finalement constitué d'une association de plusieurs objets virtuels. Le smartphone serait alors

le parfait exemple d'une telle structure avec un unique appareil physique qui regroupe une multitude d'objets virtuels via ses nombreux capteurs. [32]

Jérome Gorin [34] considère quant à lui qu'on object connecté "dispose d'une interface de communication standard (Wi-Fi, Ethernet, Zigbee, Bluetooth, etc.) et de capacités de traitement de données (capteur de température, humidité, etc.) [34]

Ces trois définitions d'objet connecté définissent toutes le fait qu'un objet connecté soit constitué de capteurs et/ou d'actionneur et soit en mesure de transmettre de l'information à d'autres entités (utilisateurs ou machines). [32]

1.2.3 Les types d'objets connectés

D'après plusieurs définitions de plusieurs scientifiques, nous avons conclu dans la section précédente que l'objet connecté peut être physique ou virtuel, nous avons notamment cité le smartphone en exemple, le fait qu'il soit un objet physique constitué de plusieurs objets connectés virtuels.

1.2.3.1 Objet connecté physique

Un objet connecté physique appartient au monde réel. Ces objets peuvent être détectés, contrôlés et connectés. L'environnement qui nous entoure, les robots industriels, les biens et les équipements électriques sont tous des exemples d'objets physiques. [26]

Ces objets physiques peuvent être intégrés dans des réseaux informatiques et interagir avec d'autres objets connectés, des systèmes informatiques et des utilisateurs, ce qui leur permet de collecter, de traiter et de partager des données pour fournir des fonctionnalités et des services avancés.

1.2.3.2 Objet connecté virtuel

Un objet connecté virtuel appartient au monde de l'information, il existe uniquement sous forme numérique, sans existence physique. Il peut être stocké, traité et récupéré. Ces objets sont, par exemple, des contenus multimédias ou des logiciels. [26]

Par exemple, un assistant virtuel comme Siri d'Apple ou Alexa d'Amazon est un objet connecté virtuel qui peut répondre à des commandes vocales, fournir des informations et exécuter des tâches sans posséder de forme physique tangible.

1.2.4 Les composants d'un objet connecté physique

L'objet connecté physique a pour fonction de collecter des données de capteurs, de traiter ces données et de les communiquer à l'aide de d'une fonction de connectivité et de recevoir des instructions pour exécuter une action. Généralement ces fonctions de l'objet connecté nécessitent une source d'énergie, surtout quand les données sont prétraitées directement dans l'objet [6].

Afin d'assurer le bon fonctionnent d'un objet connecté, plusieurs composants sont requis, nous allons les citer dans ce qui suit : [6]

a) **Les capteurs :** Les capteurs sont des dispositifs permettant de transformer une grandeur physique observée (température, luminosité, etc. . .) en signaux analogiques ou numériques utilisables par des logiciels. [6] Ces signaux sont ensuite utilisés par l'objet connecté pour prendre des décisions, déclencher des actions ou transmettre les données à d'autres appareils ou à des serveurs distants. Exemples de capteurs : lumière, présence, déplacement...

b) **Les sources d'énérgie :** Les sources d'énergie sont de quatre types : [6]

- Alimentation filaire pour les objets ayant accès à une prise de courant.
- Piles ou batteries pour ceux qui n'y ont pas accès ou de manière occasionnelle (recharge).
- Capteurs d'énergie (photovoltaïque, thermoélectrique, . . .) comme les panneaux solaires pour rallonger la durée de vie des objets à très faible consommation.
- Objets passifs sans piles qui sont alimentés par les ondes électromagnétique des lecteurs (RFID pour Radio Frequency Identification, NFC pour Near Field Communication,...)

c) **Les actionneurs :** Les actionneurs sont des dispositifs qui transforment une donnée digitale en phénomène physique pour créer une action, ils sont en quelque sorte l'inverse du capteur. Exemples d'aactionneurs : Afficheurs, Alarmes, Caméras, Haut-parleurs, Interrupteurs, Lampes, Moteurs... [6]

d) **La connectivité :** La connectivité de l'objet est assurée par une petite antenne radio-fréquence qui va permettre la communication de l'objet vers un ou plusieurs réseaux. [6] Les objets pourront d'une part remonter des informations telles que leur identité, leur état, une alerte ou les données de capteurs, et d'autre part recevoir des informations telles que des commandes d'action et des données. [6]

La figure 1.2 résume les composants d'un objet connecté physique :

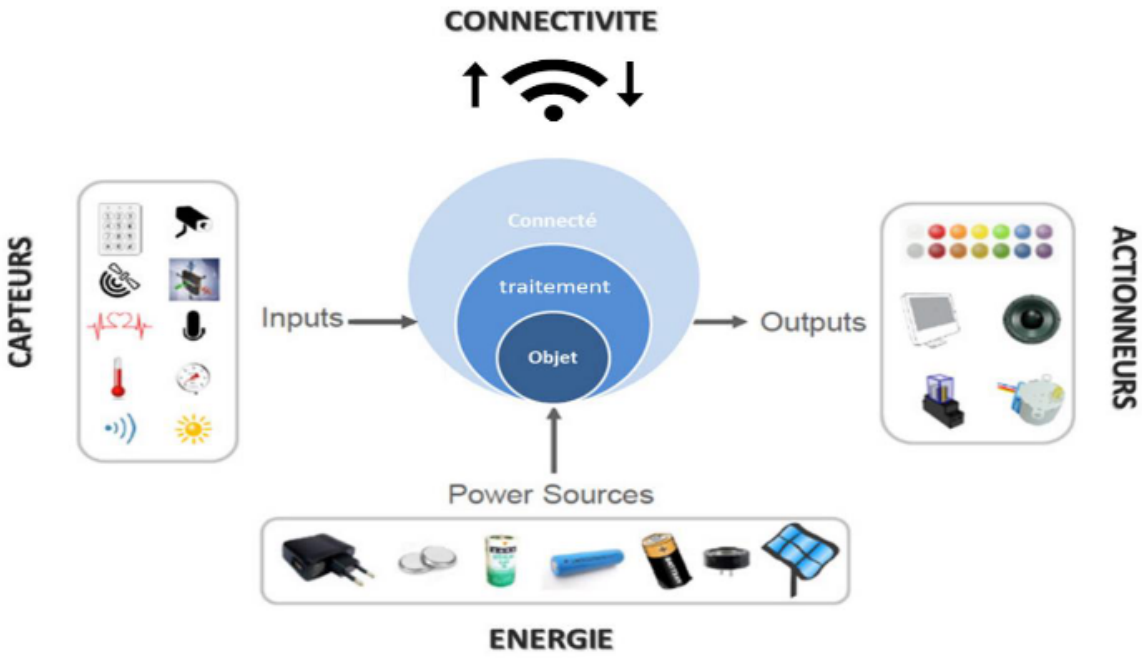


Figure 1.2: Les composants d'un objet connecté physique

1.2.5 Architecture de l'IoT

Au vu de l'objectif principal de l'IoT qui est de connecter un nombre astronomique d'objets sur Internet, il est clair qu'il faille une architecture pour organiser et structurer l'IoT afin qu'il fonctionne correctement. Actuellement, il n'y a pas d'architecture universelle de l'IoT qui mette la communauté scientifique d'accord, plusieurs ont été proposées mais celle que nous choisissons dans ce mémoire est l'architecture à trois couches, composée de la couche de perception, de la couche réseau et de la couche application.

Nous allons détailler chacune de ces couches dans ce qui suit.

1.2.5.1 La couche de perception

Cette couche est constituée d'objets physiques composés de capteurs, de dispositifs RFID, de mobiles, etc. Le rôle principal de la couche perception est essentiellement d'identifier et de collecter les informations spécifiques aux objets. Selon le type d'objets physiques, les informations peuvent être des informations de type localisation, humidité, température, qualité de l'air, etc... [44].

Les informations ainsi collectées sont ensuite transmises à la couche réseau pour leurs transmissions via des canaux sécurisés. La grande quantité de données générées par l'IoT provient de cette couche. Le coût des objets connectés et leur autonomie sont les contraintes de cette couche, en dehors des données massives qu'ils génèrent [44].

La couche perception utilise plusieurs technologies lui permettant de collecter les données depuis l'environnement extérieur. Voici les plus utilisées :

a) **RFID:** est une méthode permettant de mémoriser et récupérer des données à distance.

Le système est activé par un transfert d'énergie électromagnétique entre une étiquette radio et un émetteur RFID. L'étiquette radio composée d'une puce électronique et d'une antenne reçoit le signal radio émis par le lecteur lui aussi équipé d'une technologie RFID. Les composants permettent à la fois de lire et de répondre aux signaux [21].

b) **NFC:** La NFC est une technologie de communication sans fil à courte portée qui permet l'échange de données entre des dispositifs compatibles NFC situés à proximité les uns des autres.

1.2.5.2 La couche réseau

La couche réseau est responsable de la transmission des données vers la couche perception et la couche application, elle gère la connectivité réseau des dispositifs (contrôles d'accès, authentification...). Cette couche englobe les technologies et les protocoles qui permettent la communication entre les appareils IoT et les systèmes de gestion des données, ce qui assure la collecte, le traitement et l'analyse des données IoT.

Cette couche utilise plusieurs technologies telles que Wi-Fi, Bluetooth, Zigbee... Voici quelques exemples :

a) **Wi-FI :** Le Wi-Fi est une technologie de réseau sans fil qui permet aux périphériques tels que des ordinateurs (portables et fixes), des périphériques mobiles (téléphones intelligents et dispositifs portables), et d'autres équipements (imprimantes et caméras vidéo) d'accéder à Internet. Il permet à ces appareils, et à de nombreux autres, d'échanger des renseignements entre eux, ce qui crée un réseau [3].

Il peut prendre en charge de nombreux appareils simultanément. Il est notamment utilisé dans les maisons intelligentes, les villes intelligentes et dans le domaine de l'industrie.

b) **WSN ou Wireless Sensor Network :** Un WSN, ou Wireless Sensor Network, est un réseau de capteurs sans fil composé de nombreux nœuds capteurs distribués. Ces nœuds sont équipés de capteurs pour mesurer des données en temps réel telles que la température, l'humidité, la lumière, etc.

Ils sont utilisés dans des applications telles que la surveillance de l'environnement, la gestion des ressources, la surveillance de la santé et la domotique.

c) **Bluetooth :** Bluetooth désigne une norme de communication sans fil par ondes radio capable de transmettre des données et de la voix entre deux appareils électroniques compatibles. Le Bluetooth est notamment très répandu sur les téléphones mobiles, les écouteurs et casques sans fil ou encore les enceintes nomades. Il fonctionne sur les bandes de fréquence 2,4 GHz avec une portée maximale qui varie de 10 à 100 mètres. [10]

d) **Zigbee :** Zigbee est un protocole sans fil conçu pour les réseaux IoT, offrant une communication à faible puissance et à courte portée. Il fonctionne sur la bande de fréquences

de 2,4 GHz et est particulièrement adapté aux applications nécessitant une faible consommation d'énergie et une longue durée de vie de la batterie. Zigbee prend en charge le maillage de réseau, permettant une communication fiable entre les appareils via des nœuds intermédiaires. Il est très souvent utilisé dans les applications domotiques.

e) **LoRaWAN :** LoRaWAN est un protocole de communication radio qui constitue l'architecture du système et qui permet d'avoir une transmission de données bas débit mais surtout longue portée pour des objets connectés (exemple : capteur IoT communicant) [16].

f) **Les réseaux cellulaires :** sont des réseaux de télécommunications sans fil qui utilisent une architecture cellulaire pour fournir une couverture étendue. Ils fournissent une connectivité fiable et étendue pour les appareils IoT à travers des technologies telles que la 2G, la 3G, la 4G (LTE), et la 5G. Ces réseaux permettent aux dispositifs IoT d'être connectés à Internet et de transmettre des données à partir de n'importe où, ce qui les rend idéaux pour les applications nécessitant une couverture large et une connectivité permanente, telles que le suivi à distance, la surveillance de l'environnement et les applications industrielles.

1.2.5.3 La couche application

La couche application permet l'interaction directe avec les utilisateurs finaux. Les applications peuvent être déployées dans différents domaines de la vie quotidienne tels que la santé intelligente, le transport intelligent, les maisons intelligentes, la surveillance routière, l'industrie intelligente etc. Elle comprend également des infrastructures serveur et Cloud partageant du contenu et fournissant des services en temps réel. Le traitement des données et la fourniture de services sont deux des fonctions les plus importantes de cette couche [44].

La couche application fournit une gestion globale du système IoT. Il reçoit des informations de la couche réseau qui permet aux développeurs de créer diverses applications en utilisant une abstraction et des interfaces ouvertes et de haut niveau. Et c'est dans cette couche que toutes les décisions de contrôle, de sécurité et de gestion des applications sont prises [44].

Cette couche offre plusieurs applications dans une multitude de domaines tels que :

a) **Les villes intelligentes :** Plusieurs définitions différentes existent pour le terme "ville intelligente" mais elle peut être définie comme une zone urbaine (englobant différents secteurs de la ville - rue, place, quartier, ou, finalement, une ville entière) qui utilise des capteurs de collectent de données électroniques situées dans les infrastructures, les bâtiments, les véhicules, les institutions et les appareils pour fournir des informations en temps réel sur les principaux systèmes opérationnels des villes [38].

La figure 1.3 est une illustration d'une ville intelligente.

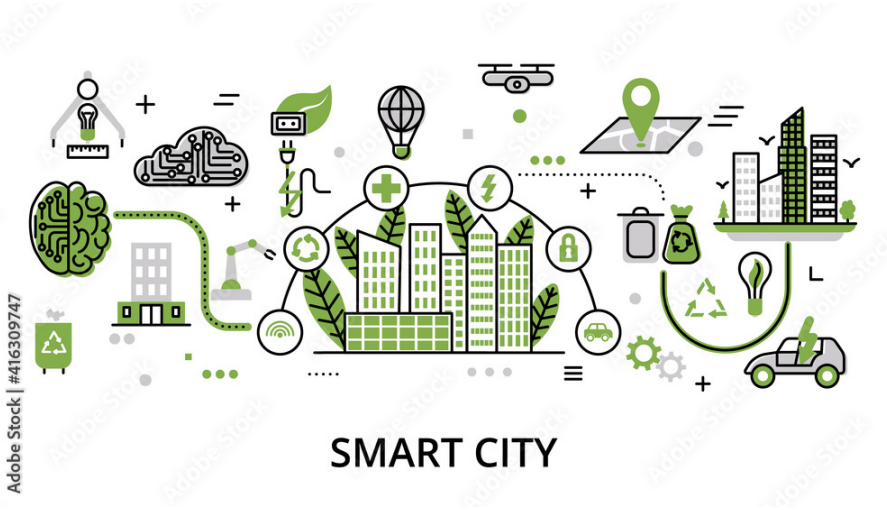


Figure 1.3: Une ville intelligente

b) **La domotique :** Une maison intelligente connectée est une résidence équipée de divers capteurs, de systèmes et de dispositifs qui permettent aux propriétaires de contrôler, surveiller à distance et automatiser les différentes fonctions de leur maison. Cette technologie permet de créer un environnement plus confortable et sécurisé pour les habitants, tout en facilitant leur vie quotidienne tel que : la sécurité, la gestion de l'énergie et le divertissement.

Nous pouvons résumer ce que nous avons expliqué dans la figure 1.4 suivante :

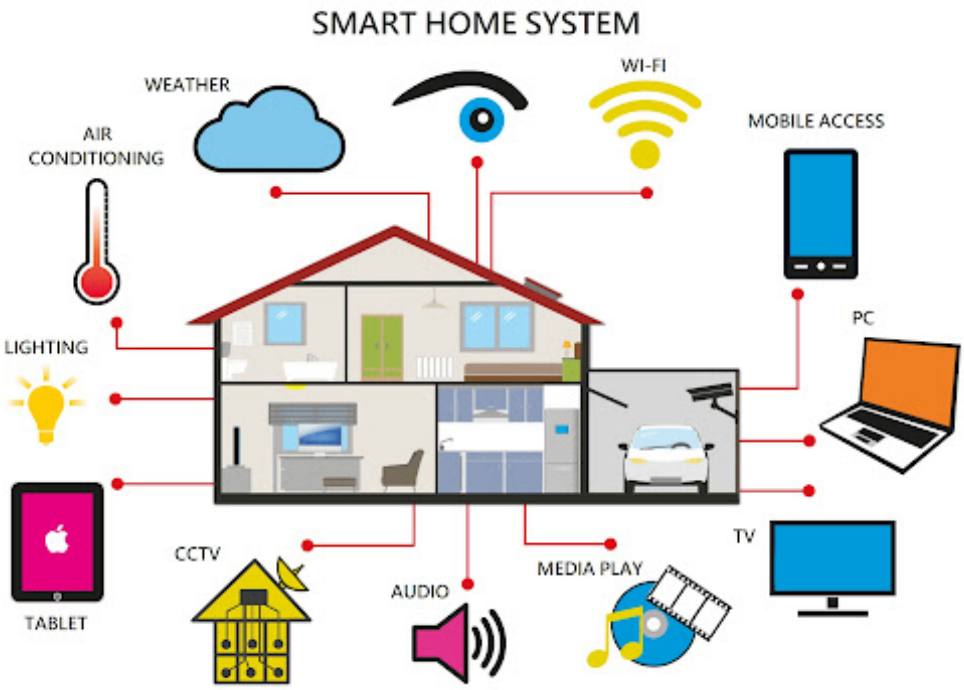


Figure 1.4: Une maison intelligente

c) **La santé intelligente :** La santé intelligente fait référence à l'utilisation de technologies de l'information et de la communication (TIC) pour améliorer la prévention, le diagnostic,

le traitement et la gestion des soins de santé. Cela inclut l'utilisation de dispositifs connectés, de capteurs, d'applications mobiles et de systèmes informatiques pour collecter, surveiller et analyser les données médicales et de bien-être des patients. L'objectif est d'optimiser les soins de santé en permettant une surveillance continue et en temps réel, un suivi personnalisé et une intervention précoce, tout en réduisant les coûts et en améliorant l'efficacité des traitements.

d) **L'agriculture :** L'agriculture intelligente est une pratique de production agricole qui consiste à utiliser des technologies telles que l'IoT pour gérer les exploitations agricoles, augmenter la production et réduire les impacts environnementaux. L'objectif de l'agriculture intelligente est d'augmenter la qualité et la quantité des produits agricoles tout en optimisant le travail humain afin d'obtenir les meilleurs résultats possibles.

Les exploitations et les sites agricoles qui utilisent des techniques d'agriculture intelligente sont appelés "fermes intelligentes". Ces exploitations utilisent généralement de nouveaux outils tels que des capteurs qui surveillent les conditions météorologiques 24 heures sur 24 et des applications d'inspection mobiles qui aident les agriculteurs à améliorer leurs pratiques agricoles afin d'augmenter leur production tout en favorisant la durabilité et en maintenant la qualité des produits. [20]

d) **L'industrie :** Incorporer l'IoT dans le domaine de l'industrie revient à utiliser des capteurs et des objets connectés pour collecter des données en temps réel sur les équipements, les processus de production et ce qui se rapporte aux opérations de logistique. Cela permet d'optimiser la production et la qualité des produits, augmenter les bénéfices et assurer la sécurité des employés.

1.3 Les caractéristiques de l'IoT

1.3.1 Données sensibles

Les données des utilisateurs collectées par les dispositifs IoT sont majoritairement sensibles. Elles doivent donc être traitées afin de protéger la vie privée des utilisateurs [48].

1.3.2 Hétérogénéité

L'IoT est constitué de dispositifs et de systèmes basés sur différentes configurations matérielles, loogicielles et réseau. Ces dispositifs interagissent entre eux et avec différentes plateformmes de services. Cette caractéristique doit être prise en compte dans les applications IoT afin de permettre l'interopérabilité et la modularité [48].

1.3.3 Ressources restreintes

Le faible coût et les environnements dans lesquels les dispositifs doivent être déployés ont un impact sur leurs capacités. Cela se traduit par des ressources (processeur, mémoire vive, énergie, stockage) restreints [48].

1.3.4 Dynamisme

Les applications IoT collectent des données du monde physique par les dispositifs IoT. Les changements dynamiques de l'environnement de l'utilisateur peuvent être perçus à travers les données collectées par les dispositifs. Ce dynamisme permet aux applications IoT de détecter et de s'adapter aux changements du monde physique [48].

1.3.5 Intelligence

L'intelligence est une caractéristique de l'Iot qui consiste en la mise en oeuvre de la connaissance dans l'IoT par une association d'algorithmes et de techniques de traitement de données évolués (Machine Learning, raisonnement...). Elle aide à l'adaptation dynamique et à la prise de décision. Elle permet l'adaptation automatique du niveau de sécurité en fonction des risques et des paramètres de sécurité associés au contexte de l'utilisateur [48].

1.3.6 Temps réel

Le traitement en temps réel des données est nécessaire pour certaines applications IoT critiques comme celle de la télésurveillance des patients. C'est également un besoin pour la mise en oeuvre de l'adaptation dynamique du niveau de sécurité des applications IoT. En effet, l'adaptation du niveau de sécurité doit être effectuée en continu et en temps réel.

1.4 La sécurité au niveau de la couche application

Etant donné l'importance de la couche application dans une architecture Iot, il est tout à fait compréhensible qu'on s'intéresse à la sécurité au niveau de cette couche. En effet, elle intéragit de façon directe avec les utilisateurs finaux et leur fournit les services qu'ils demandent. Par conséquent, elle est particulièrement vulnérable aux attaques et aux violations de sécurité.

La sécurisation de la couche application est essentielle pour garantir la confidentialité, l'intégrité et la disponibilité des données échangées entre les appareils IoT et les applications, ainsi que pour prévenir les attaques et assurer le bon fonctionnement des services IoT et protéger les personnes qui intéragissent dans un environnement où des dispositifs IoT sont déployés comme une maison intelligente, un aéroport intelligent. Dans ce cas, nous parlons de dispositifs de sécurité mis en oeuvre afin d'assurer la protection de ces individus tels que la surveillance vidéo, les contrôle d'accès...

On distingue donc deux types de sécurité dans ce contexte : la sécurité propre aux objets connectés ainsi que la sécurité des personnes qui évoluent dans un environnement IoT que nous allons appeler sécurité physique. Nous allons définir les deux dans ce qui suit.

1.4.1 La sécurité dans le contexte de l'IoT

Cette sécurité fait référence à la mise en place de mesures de sécurité pour protéger les données et les services au sein des applications IoT. Elle vise à garantir la confidentialité, l'intégrité et la disponibilité des données échangées entre les dispositifs IoT et les applications. Cela comprend entre autres :

- La sécurisation des communications en utilisant des protocoles de communication sécurisés tels que HTTPS, TLS/SSL...
- L'authentification des utilisateurs afin de s'assurer que leur identité est légitime avant de leur donner accès aux services IoT. Cela peut se faire à l'aide de mots de passe, de certificats numériques ou autres méthodes d'identification.
- Le contrôle d'accès afin de contrôler les accès aux services IoT et déterminer qui peut accéder, à quoi et quand. Cela peut inclure la définition de politiques d'accès par exemple.

1.4.2 La sécurité physique dans le contexte de l'IoT

La sécurité physique dans le contexte de l'IoT concerne la protection des personnes, infrastructures et environnements contre les menaces physiques dans lesquelles les systèmes IoT sont déployés. Cela englobe la sécurité des utilisateurs, des techniciens et du personnel dans divers environnements tels que les maisons intelligentes, les établissements gouvernementaux, les écoles...

Elle comprend la mise en oeuvre de mesure de sécurité telles que la surveillance vidéo, les contrôles d'accès, les dispositifs de verouillage, les systèmes d'alarmes...

Elle implique également la protection des objets connectés eux-mêmes contre le vol en veillant à ce qu'ils soient installés et entretenus de manière sécurisée. L'objectif est d'utiliser les systèmes IoT afin d'assurer un environnement sûr et sécurisé pour les personnes impliquées dans l'écosystème de l'IoT.

L'illustration suivante à travers la figure 1.5 résume parfaitement ce que nous venons d'expliquer dans cette section à propos de l'architecture IoT à trois couches.

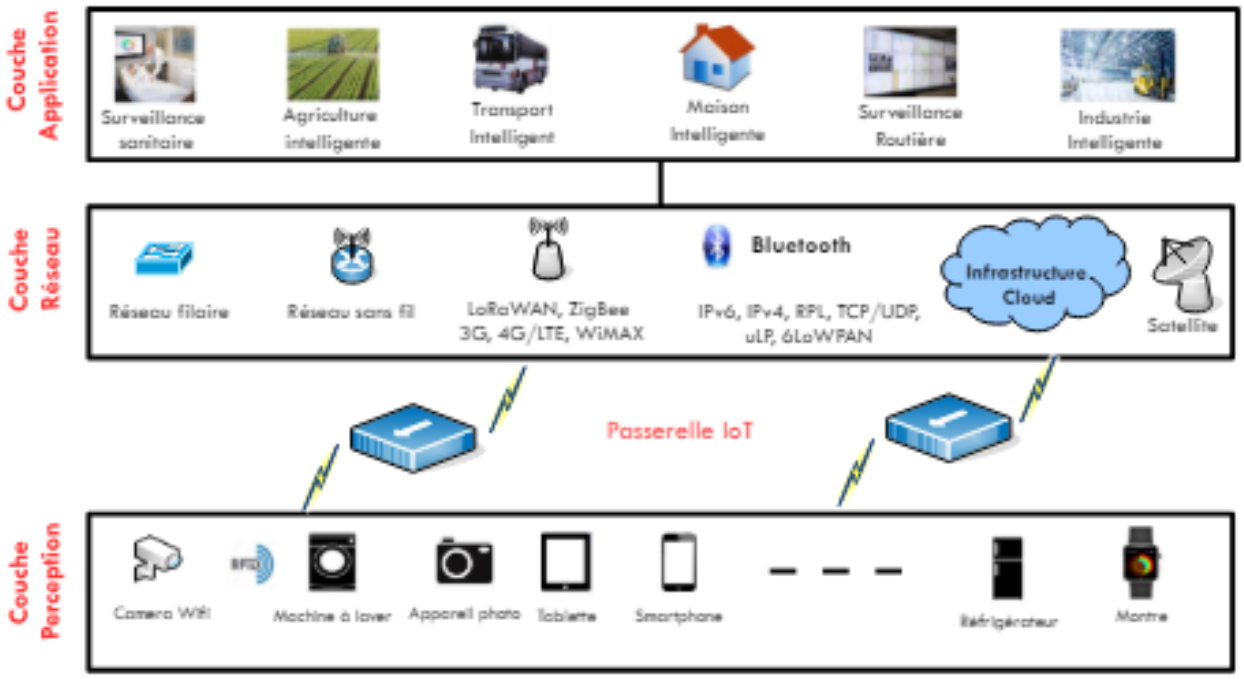


Figure 1.5: Architecture IoT à trois couches

1.5 Les défis de la sécurité dans l'IoT

Malgré les nombreux avantages de l'IoT, les défis qui freinent son développement sont nombreux [47].

La sécurité et la protection de la vie privée sont parmi les défis les plus importants à relever dans l'Internet des Objets. Selon la littérature, plusieurs risques de sécurité dans l'IoT proviennent principalement de la vulnérabilité des objets. Ces vulnérabilités sont essentiellement dûes aux contraintes liées à leurs performances restreintes et à la non prise en charge des questions de sécurité dès leur conception. Ils facilitent les accès non autorisés aux informations personnelles et l'accès à d'autres parties des réseaux. Autrement dit, ils constituent une grande surface d'attaque. Cependant un autre risque lié aux vulnérabilités des objets connectés est le risque d'attaques personnelles, certains objets jouant un rôle vital chez leurs porteurs. Par exemple, un régulateur cardiaque connecté pourrait être la cible d'un attaquant visant l'assassinat de son porteur [47].

La nature dynamique et hétérogène de l'IoT, la connectivié constante entre les dispositifs, ainsi que le manque de normes de sécurité font que les dispositifs classiques de sécurité tels que les pares-feu par exemple sont insuffisants dans le contexte de l'IoT. En raison de ces défis, il est donc nécessaire de développer des solutions de sécurité spécifiquement adaptées aux dispositifs IoT, en tenant compte de leurs contraintes et de leurs caractéristiques uniques.

1.5.1 Contexte sécuritaire de l'IoT

Les objets de l'IoT, en raison du nombre important d'appareils et de la variété des domaines impactés, nécessitent une sécurisation appropriée. Le cryptologue Bruce Schneier décrit en 2016 que "le marché récompense encore largement le sacrifice de la sécurité au profit du prix et du délai de mise sur le marché" [45].

Cette constatation est confirmée par Barcena et Wueest [24] qui, dans leur étude sur 50 des objets les plus commercialisés, ont constaté qu'aucun des ces objets n'imposait de mots de passe suffisamment robustes, n'utilisait d'authentification mutuelle entre clients/serveurs ou ne bénéficiait de protection contre les attaques par force brute [32].

SERMA [31] a constaté que 90% des industriels n'avaient pas connaissance des menaces réelles de l'IoT, que 84% n'évaluaient jamais la sécurité de leurs produits et que près de 70% ne savent pas en évaluer leur robustesse. Quand bien même une évaluation de la sécurité serait demandée, SERMA [31] affirme que près de 90% des consultants en sécurité informatique ne bénéficient pas d'une formation suffisante à la sécurisation des objets de l'IoT [32].

L'ensemble de ces éléments font des objets de l'IoT des cibles de choix pour les attaquants.

1.6 Conclusion

En conclusion, ce premier chapitre nous a permis d'avoir un aperçu de l'IoT car il offre des possibilités infinies dans de nombreux domaines tels que la santé et la domotique.

Nous avons souligné l'importance de la sécurisation des données, des communications et du contrôle d'accès ainsi que la sécurité d'une personne dans le contexte de l'IoT. En examinant ces différents aspects de la sécurité, nous avons identifié les défis uniques auxquels est confrontée la sécurité de l'IoT.

Dans le prochain chapitre, nous allons choisir la maison intelligente comme domaine d'application et vidéosurveillance comme moyen de détection de personnes.

Chapitre 2

Vidéosurveillance dans les maisons intelligentes

2.1 Introduction

Dans un environnement aussi connecté qu'une smart home, aussi dite maison intelligente, la sécurité constitue un aspect non négligeable pour le résident - que ce soit une personne âgée vivant seule, des parents avec des enfants en bas âge... – car avec cette interconnexion accrue vient une vulnérabilité aux intrusions physiques.

Dans ce contexte, la sécurité de l'individu repose non seulement sur des mécanismes classiques tels que les alarmes et les verrous mais également sur des mécanismes plus sophistiqués tels que la vidéosurveillance qui offre un bon nombre d'avantages. Ainsi, la vidéosurveillance n'est plus seulement une fonctionnalité supplémentaire mais un élément clé afin d'assurer la sécurité des résidents.

Dans ce deuxième chapitre, nous allons choisir les maisons intelligentes comme contexte d'étude et comprendre comment fonctionne la vidéosurveillance afin de détécter les intrusions physiques, ainsi qu'un état de l'art sur les techniques de sécurisation d'une maison intelligente en utilisant la vidéosurveillance.

2.2 Définition

La vidéosurveillance dans une maison intelligente est un système intégré de caméras connectées à Internet ainsi que des technologies de traitement d'images pour surveiller en temps réel les activités à l'intérieur et à l'extérieur de la maison. Les systèmes peuvent détecter des mouvements, identifier des individus et alerter instantanément les propriétaires en cas de comportement suspect. Ces systèmes s'intégrent généralement avec d'autres dispositifs IoT pour offrir une meilleure sécurité.

2.3 Fonctionnement de la vidéosurveillance

Le but de la vidéosurveillance dans une maison intelligente est de détecter, reconnaître et suivre des personnes à partir de séquences d'images récoltées et d'essayer de comprendre et reconnaître leurs activités et mouvements. En général, un système de vidéosurveillance interne comprend les modules suivants : (a) détection des objets, (b) suivi d'objets et (c) analyse de mouvement humain. Chacune des sections suivantes explique les techniques les plus couramment utilisées dans chacun des modules.

2.3.1 Détection des objets

La détection des objets en mouvement est la première étape ainsi que la plus importante dans l'analyse d'une vidéo. Cette étape permet de détecter toute activité spécifique dans la zone surveillée comme le mouvement, l'apparition ou la disparition d'objet [43].

Plusieurs méthodes sont utilisés dans la détéction de mouvement, on cite notamment la soustraction d'arrière-plan, la différence temporelle et le flux optique. Ces méthodes tentent de localiser des régions connectées de pixels représentant les objets en mouvement dans la scène. Nous allons expliquer brièvement chacune des techniques dans ce qui suit.

- **La soustraction de l'arrière-plan :** C'est une étape essentielle de tout traitement vidéo afin de détecter l'objet en mouvement. Elle permet de classer les pixels en arrière-plan et en premier plan. Le processus de soustraction d'arrière-plan prend en entrée une image codée dans un espace couleur quelconque et produit à la fin une image en noir et blanc. Le principe de cette technique est de mesurer la différence entre l'image actuelle et l'image d'arrière-plan. Si la différence entre les pixels est supérieure à la valeur seuil (T), on considère qu'il s'agit des pixels de l'objet en mouvement, sinon, il s'agit des pixels de l'arrière-plan [35].

  Cette méthode est assez simple mais est sensible aux changements d'éclairage.

  Voici une exemple du fonctionnement de cette méthode montré dans la figure 2.1 suivante:

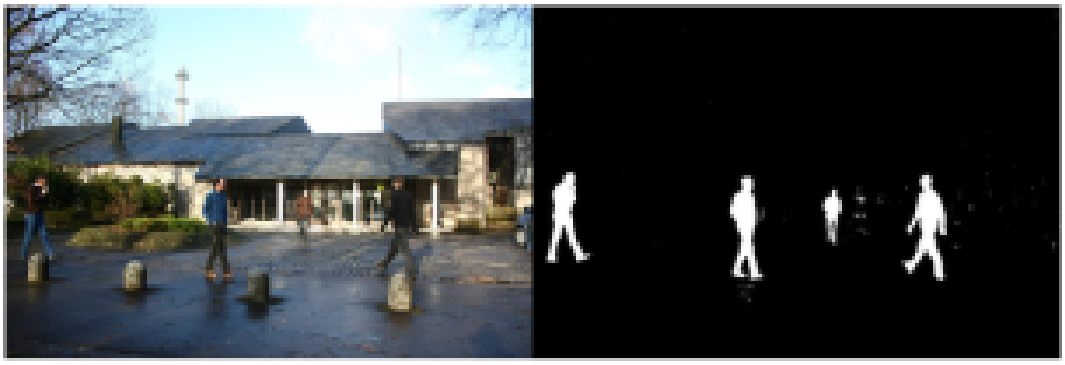


Figure 2.1: Soustraction d'arrière-plan

- **La différence temporelle :** Dans cette méthode, la différence de pixels entre deux images consécutives est calculée, ce qui permet d'identifier l'objet en mouvement [43].

- **Le flux optique :** Cette méthode basée sur un processus de regroupement en fonction de distribution de flux optique de l'image [43]. Cette technique tente de calculer le mouvement entre deux images qui sont prises aux instants t et t + δt. Cette méthode est dite différentielle, car elle est basée sur la dérivée partielle par rapport aux coordonnées spatiales et temporelles.

2.3.2 Suivi d'objets

Le suivi d'objet est l'étape succédant à la détection des objets. Elle a pour but de localiser un objets dans une séquence vidéo en fonction de son emplacement dans la première image.

Les techniques de suivi d'objets sont regroupées principalement en trois catégories : suivi par points, suivi par noyaux et suivi par silhouettes [43].

- **Le suivi par points :** Dans cette méthode, le suivi des objets est effectué en fonction de la représentation des points caractéristiques des objets en mouvement essentiels. Cette méthode présente deux phases importantes : la prédiction qui consiste à utiliser toutes les observations actuelles et à les mettre à jour pour l'étape suivante. La deuxième phase est la correction qui fournit une meilleure estimation de l'étape suivante [43].

- **Le suivi par noyaux :** Le suivi de noyau est typiquement réalisé en calculant le mouvement de l'objet lorsqu'il se déplace d'une image à l'autre. Le mouvement de l'objet se présente généralement sous la forme d'un mouvement paramétrique (translation, conformation, affine, etc.) ou du champ de flux dense calculé dans les images suivantes. Les algorithmes de cette méthode sont classés selon plusieurs critères : nombre des objets suivis, la méthode utilisée pour estimer le mouvement de l'objet et la représentation d'apparence utilisée [50].

- **Le suivi par silhouettes :** Dans la plupart des cas, les objets n'ont pas de contours géométriques simples, comme un corps humain avec des formes géométriques complexes. L'objectif de ce mécanisme de suivi est de trouver la région de l'objet (tel que les mains, la tête, les épaules, etc.) dans chaque image à l'aide d'un modèle d'objet généré par les images précédentes [43]. Le principal défi de création de cette méthode est les formes composites des objets qui ne sont pas facilement définies par des formes géométriques.

2.3.3 Analyse du mouvement humain

L'analyse comportementale, considérée comme une tâche de niveau supérieur, implique d'interpréter les comportements et interactions des objets à partir de toutes les informations recueillies lors du processus de suivi. Cette tâche requiert une analyse sémantique, parfois complexe, adaptée au contexte de l'application et de l'événement détecté [46].

2.4 Les caméras utilisées dans la vidéosurveillance

Dans une maison intelligente, plusieurs types de caméras peuvent être utilisés, nous allons citer les plus connues.

a) **Les caméras de sécurité connectées au Wi-Fi :** Ces caméras se connectent au réseau Wi-Fi domestique et peuvent être contrôlées via une application mobile. Elles sont souvent utilisées pour la surveillance interne et externe. L'avantage de ces modèles est qu'elles sont dépourvues de branchements filaires, elles peuvent être disposées n'importe où dans la maison, pourvu qu'elles soient à portée suffisante de la box Internet. On peut citer comme exemple Arlo Pro 3 qui offre une résolution vidéo 2K HDR, la vision nocturne en couleur ainsi que la détection de mouvement avancée, il y a aussi Ring Stick Up Cam, Google Nest Cam IQ Indoor...etc

b) **Les caméras IP :** Une caméra IP est un dispositif de surveillance vidéo qui se connecte à un réseau informatique, comme Internet ou un réseau locan (LAN), pour transmettre des données audio et vidéo. Contrairement à une caméra analogique, la caméra IP numérise les images directement dans l'appareil et le transmet sous forme de flux de données via un cable Ethernet ou sans fil. Elles offrent souvent des fonctionnalités avancées telles que la détection de moouvement, la vision nocturne et la communication audio bidirectionnelle. On cite Axis M3045-V, Amcrest ProHD et Reolink RLC-511W comme exemples de caméras IP.

c) **Les caméras de surveillance intelligentes :** Ces caméras utilisent des algorithmes d'intelligence artificielle pour analyser les images en temps réel et détecter automatiquement les événements suspects, comme les intrusions ou les mouvements inhabituels. Voici quelques exemples de caméras intelligentes : Blink Mini, Wyze Cam Pan, Ring Indoor Cam...etc

2.5 Les avantages de la vidéosurveillance dans les maison intelligentes

La vidéosurveillance dans une maison intelligente offre un nombre considérable d'avantages, parmi lesquels on peut notamment citer :

a) **Détection précoce des intrusions :**

Les caméras de surveillance permettent de détecter les intrusions en temps réel, souvent avant que des dommages significatifs ne se produisent. Les systèmes intelligents peuvent envoyer des alertes instantanées aux propriétaires et aux services de sécurité et garantir une surveillance continue même lorsque le résident est absent.

b) **Surveillance à distance :**

Les propriétaires peuvent accéder aux flux vidéo en direct et aux enregistrements via des applications mobiles, leur permettant de surveiller leur maison de n'importe où dans le monde, ils peuvent également recevoir des notifications en temps réel lors de la détection d'activité suspecte.

c) **Intégration avec d'autres systèmes de sécurité :**

Les systèmes de vidéosurveillance peuvent être intégrés avec d'autres dispositifs IoT, tels que les alarmes, les éclairages intelligents, et les serrures électroniques, pour créer une solution de sécurité automatisée.

d) **Preuves en cas d'incident :** Les enregistrements vidéo fournissent des preuves précieuses en cas de vol ou de vandalisme. Ces vidéos peuvent être utilisées par la police et les assurances pour enquêter sur les incidents et traiter les réclamations.

d) **Protection des proches :** Les systèmes de vidéosurveillance aident également à surveiller les enfants, les personnes âgées, ou les animaux de compagnie, assurant leur sécurité et bien-être.

2.6 Les moyens d'authentification des personnes

Un système de vidéosurveillance domestique, pour être efficace, se doit de vérifier l'identité des résidents et des visiteurs tout en offrant une surveillance continue pour détecter toute activité suspecte. Pour ce faire, le système utilise plusieurs algorithmes de l'apprentissage automatique, ce qui lui permet d'apprendre et de s'améliorer automatiquement à partir de l'expérience sans être explicitement programmé. Il s'agit de classification, de régression, de regroupement et d'optimisation.

Voici quelques techniques d'authentification utilisées dans ce contexte :

2.6.1 La reconnaissance faciale

La reconnaissance faciale est une technique qui permet d'identifier ou d'authentifier une personne à partir des traits de son visage [4]. Elle fonctionne en analysant et en comparant les caractéristiques du visage. Plusieurs méthodes sont utilisées, voici les plus courantes :

2.6.1.1 Méthodes basées sur la géométrie

Utilisée depuis les années 1970, les méthodes basées sur la géométrie reposent sur des arêtes et des contours spécialisés du visage, à partir desquels les points de repère faciaux sont détectés, puis leur position et leur distance sont mesurées. Il y a certaines méthodes où l'image faciale est transformée en primitives géométriques et des caractéristiques distinctives telles que les yeux, le nez et la bouche sont localisées et leur position est notée et d'autres où les caractéristiques

basées sur l'apparence sont utilisées par ces méthodes qui qui enregistrent les valeurs statistiques des pixels de l'image faciale et les compare [30].

2.6.1.2 Méthodes holistiques

Dans les méthodes holistiques, les visages sont représentés sous forme de matrices 2D plutôt que d'une géométrie 3D. Elles décomposent une image d'un visage en Eigenfaces qui sont prises comme composants de base pour l'entraînement des systèmes de reconnaissance faciale. Une nouvelle image est projetée dans le sous-espace engendré par les Eigenfaces et comparée avec la position de l'individu connu dans l'espace facial pour la reconnaissance [30].

Ces méthodes utilisent des techniques telles que l'analyse en composantes principales PCA (Principal Component Analysis) et l'analyse discriminante linéaire LDA (Linear Discriminant Analysis) pour identifier des modèles dans les images de visage.

2.6.1.3 Méthodes basées sur les caractéristiques

Les méthodes d'extraction de caractéristiques impliquent l'extraction de caractéristiques discriminantes plutôt que le calcul de leur géométrie, ce qui les rend plus robustes dans la représentation des variations faciales par rapport aux méthodes holistiques.

Ces méthodes utilisent des techniques telles le descripteur SIFT (Shift-Invariant Feature Transform) qui est utilisé en vision par ordinateur pour détecter des points importants sur les visages, tandis que SURF(Speeded-Up Robust Features) se base sur la détection automatique de l'échelle qui se concentre sur les coins des objets, ce qui est non seulement plus rapide mais aussi plus distinctif.

Les caractéristiques de type Haar qui sont des motifs lumineux qui aident à détecter des formes sur les visages, comme les yeux ou la bouche, et l'algorithme AdaBoost qui permet de sélectionner les meilleures caractéristiques sont également des méthodes importantes pour extraire des caractéristiques distinctives des visages [30].

2.6.1.4 Méthodes de deep learning

Les CNN (Convolutional Neural Networks) sont les réseaux neuronaux profonds les plus largement utilisés, capables d'extraire automatiquement des caractéristiques représentatives de haut niveau à partir de vastes ensembles de données, et ils sont invariants à la variation de l'éclairage, de la luminosité, de l'âge et de l'orientation faciale. Les performances peuvent être améliorées en utilisant des ensembles de données diversifiés et volumineux [30].

2.6.2 Reconnaissance d'actions basées sur l'apparence

Les approches de reconnaissance d'actions basées sur l'apparence humaine exploitent des données 2D ou 3D relatives aux parties du corps, telles que leurs positions et leurs mouvements. Cela permet de se concentrer sur ces mouvements, indépendamment du contexte environnant. Ces méthodes se divisent généralement en deux catégories : l'analyse de la forme du sujet à

travers sa silhouette, et l'analyse du mouvement du sujet en identifiant ses membres (mains, tête, jambes,... etc). Dans ce qui suit, nous allons donner quelques brèves explications.

2.6.2.1 Analyse de la silhouette

Dans les analyses de la forme par des silhouettes, une première étape est nécessaire, celle de l'extraction des formes de la personne dans les images. Cela se fait en supprimant généralement l'arrière plan. Parfois après suppression d'arrière plan, l'image résultante comporte la personne et d'autres objets. Dans ce cas, une modélisation plus complexe peut être mise en œuvre, par exemple, l'identification du personnage par un algorithme de détection d'humain. Une fois la silhouette extraite, une représentation de la séquence vidéo est nécessaire et enfin vient l'étape de la classification [28].

2.6.2.2 Analyse du squelette ou des parties du corps humain

Certains algorithmes de reconnaissance d'actions se concentrent sur la description des parties du corps humain dans l'espace 2D ou 3D. Ces méthodes utilisent des points lumineux pour décrire les mouvements des principales articulations du corps. D'autres approches se concentrent sur l'utilisation des extrémités du corps, telles que la tête, les mains et les pieds pour modéliser l'action. Ces différentes approches permettent de capturer efficacement les mouvements du corps humain, offrant ainsi une base solide pour la reconnaissance précise des actions dans divers contextes [28].

2.7 État de l'art des travaux de recherche de la littérature

2.7.1 Realtime Intrusion Detection System using OpenCV (2023)

A.S.Teja et al. [49] ont crée et implémenté un système de surveillance basé sur la reconnaissance et la détection du visage qui fournit la surveillance pour les zones restreintes et confidentielles quand un intrus ou une personne non autorisée pénètre dans la zone. La détection d'intrusion se fait par un système d'alarme sonore en temps réel. L'objectif de ce travail est de déployer un système de surveillance intelligent qui comporte cinq principales étapes comme illustré dans la figure 2.2 ci-dessous :

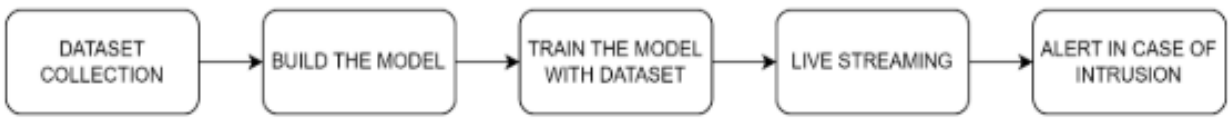


Figure 2.2: L'architecure du système proposé

L'algorithme Haar-Cascade pour la collecte des données comprenant l'identification des visages des personnes capturées depuis des vidéos, plus précisemment l'algorithme LBP (Local

Binary Pattern Algorithm) est utilisé sous le langage de programmation Pyhton pour s'organiser dans la bibliothèque OpenCV (Open Computer Vision Library).

Pour l'évaluation, ils ont pris en considération deux paramètres : la valeur de confiance qui représente la distance entre le visage détecté et le visage le plus proche dans l'ensemble des données du Dataset utilisé et la précision de reconnaissance du visage.

L'organigramme résumant les étapes du système de détection proposé est illustré dans la figure 2.3 suivante :

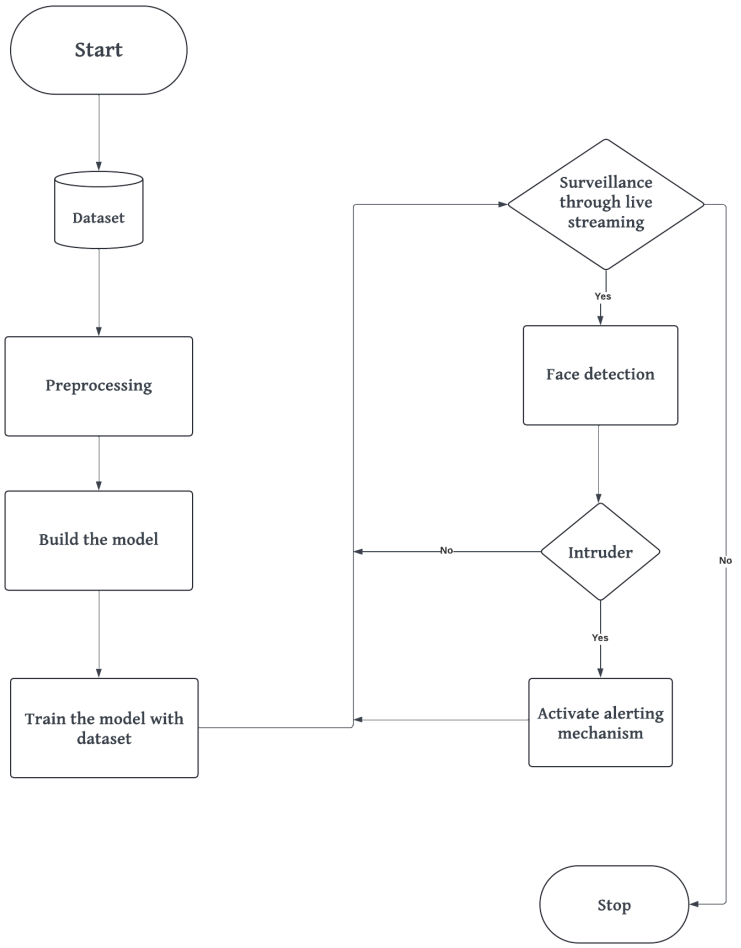


Figure 2.3: L'organigramme du système de détection proposé

2.7.2 Smart surveillance monitoring system using Machine Learning and Raspberry Pi (2022)

Dr. S.Choudhary et al. [27] ont visé à améliorer l'environnement de surveillance vidéo en utilisant des technologies de reconnaissance basées sur des systèmes embarqués et de vision par ordinateur.

Pour l'implémentation de leur projet, les auteurs utilisent Raspberry Pi 4 qui vise à utiliser des techniques de vision par ordinateur telles que la détection de mouvement, la détection d'objets, la détéction de visage...etc, pour séparer la région d'intérêt (personne ou objet) de la vidéo enregistrée. Les auteurs ont prouvé que le système proposé est performant par rapport au

système de surveillance traditionnel dans la mesure où il a la capacité de détecter des évenements suspects concernant les personnes comme les objets et cela sans intéraction ni intervention humaine dans la salle de contrôle. Pour la détection d'intrusion, la solution proposée se base sur l'envoi d'alertes par mail au propriétaire dès qu'un mouvement est détectée.

Dans le cas de détection de personnes, la caméra capture une image ou une vidéo, et OpenCV utilise un algorithme de détection de personne, plus précisemment le package Frontal Face de l'algorithme Haar Cascade.

L'organigramme de la solution proposée est donné dans la Figure 2.4 ci-dessous :

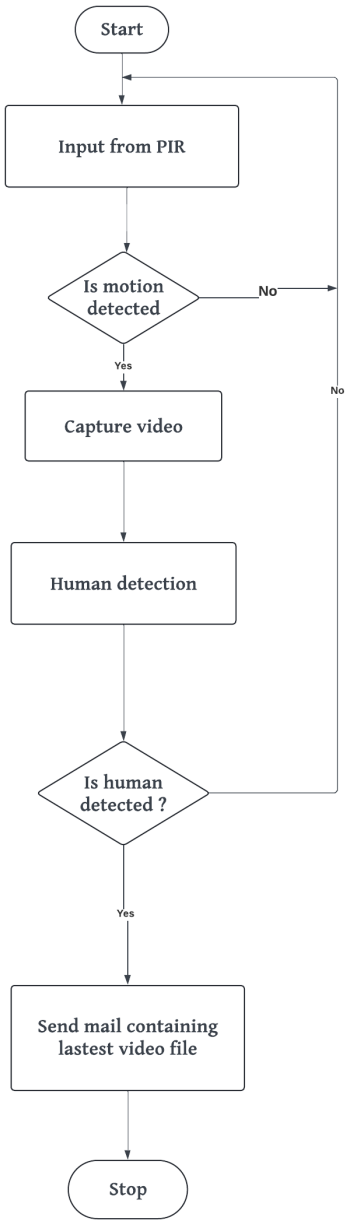


Figure 2.4: La solution proposée

2.7.3 Integration of a Video Surveillance System Into a Smart Home Using the Home Assistant Platform (2022)

Z.Nurlan et al. [41] proposent une solution intégrée pour le contrôle d'accès et la vidéosurveillance utilisant Home Assistant, qui est une plateforme open source flexible et modifiable pour contrôler divers appareils dans une maison intelligente, et MotionEyeOS qui est un système d'exploitation installé sur une caméra Raspberry Pi 4 pour transformer ce dernier en un système de vidéosurveillance et qui permet de capturer des images via le GPU (Graphic Processing Unit) et de les diffuser en MJPEG, avec la possibilité de gérer plusieurs caméras IP et de recevoir des notifications par e-mail.

Les auteurs ont proposé une solution pour la vidéosurveillance basée sur le Raspberry Pi 4 et les microcontrôleurs ESP8266 qui sont des puces Wi-Fi à faible coût et hautement intégrées. Tout d'abord, ils ont utilisé la Raspberry Pi comme base pour le système de vidéosurveillance en raison de sa polyvalence, de sa disponibilité et de son coût abordable. Ensuite, ils ont intégré des microcontrôleurs ESP8266 pour étendre les capacités du système, notamment pour la connectivité sans fil et le contrôle à distance.

Ils ont donc utilisé des caméras pour la capture d'images et des ESP8266 pour la détection de mouvement, le tout intégré et automatisé via Home Assistant pour créer un système de vidéosurveillance efficace et réactif. La solution proposée est montrée dans la figure 2.5 suivante:

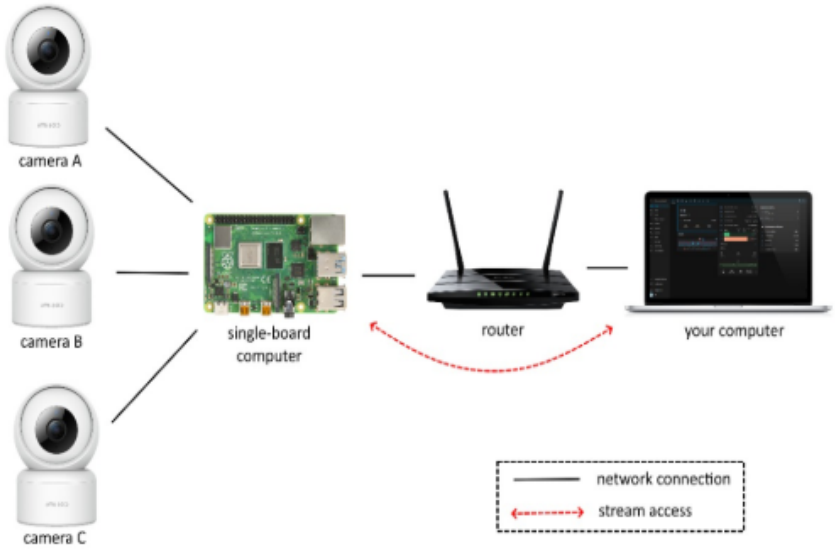


Figure 2.5: La solution proposée

L'envoi de notifications dans le système proposé se fait via Home Assistant en configurant des automatisations qui déclenchent des alertes lorsque du mouvement est détecté par les caméras ou les capteurs de mouvement connectés. Ces notifications sont envoyées aux utilisateurs via des méthodes de communication configurées, telles que les emails.

2.7.4 An novel approach object detection of video surveillance system using OpenCV (2024)

V.Kakulapati et al. [36] ont proposé une solution de surveillance vidéo intelligente et efficace pour améliorer la sécurité dans des zones à présence humaine irrégulière.

Le système utilise un classificateur Haar Casacader pour détecter les mouvements et les visages humains. Les images capturées sont ensuite converties en niveaux de gris, ce qui simplifie le traitement ultérieur. Le système commence l'enregistrement vidéo uniquement lorsqu'il détecte une présence humaine, réduisant ainsi la consommation d'énergie et les besoins de stockage.

Les auteurs ont également utilisé des algorithmes tels que LBPH (Local Binary Patterns Histograms) qui est utilisé pour la reconnaissance faciale ainsi que des méthodes telles que PCA (Principal Component Analysis) et SVM (Support Vector Machine) pour améliorer la reconnaissance et la classification des visages, avec PCA réduisant la dimensionalité des données et SVM classifiant les données en "visages" et "non-visages".

Les auteurs ont montré que le système proposé offre une alternative plus sécurisée et pratique aux mots de passe et autres méthodes d'authentification traditionnelles.

2.7.5 IOT Powered Smart Doorbell System For Enhanced Home Security And Communication (2024)

P.Donepudi et al. [42] ont visé à renforcer la sécurité grâce à la technologie de reconnaissance faciale. Ils ont proposé un système qui compare les images faciales capturées avec les images stockées, déverrouille une porte pour les utilisateurs autorisés et alerte les personnes désignées pour une décision sur l'octroi de l'accès lorsqu'un intrus est détecté, tout cela afin d'obtenir une reconnaissance rapide et précise.

Les auteurs, dans leur projet, exploitent une carte micro-contrôleur Raspberry Pi, un module de caméra Pi pour la reconnaissance faciale et un moteur pas à pas programmable pour améliorer la sécurité à domicile. La caméra Raspberry Pi est utilisée en conjonction avec une caméra USB afin de garantir une configuration de sécurité complète.

la figure 2.6 suivante nous montre l'architecture proposée par les auteurs :

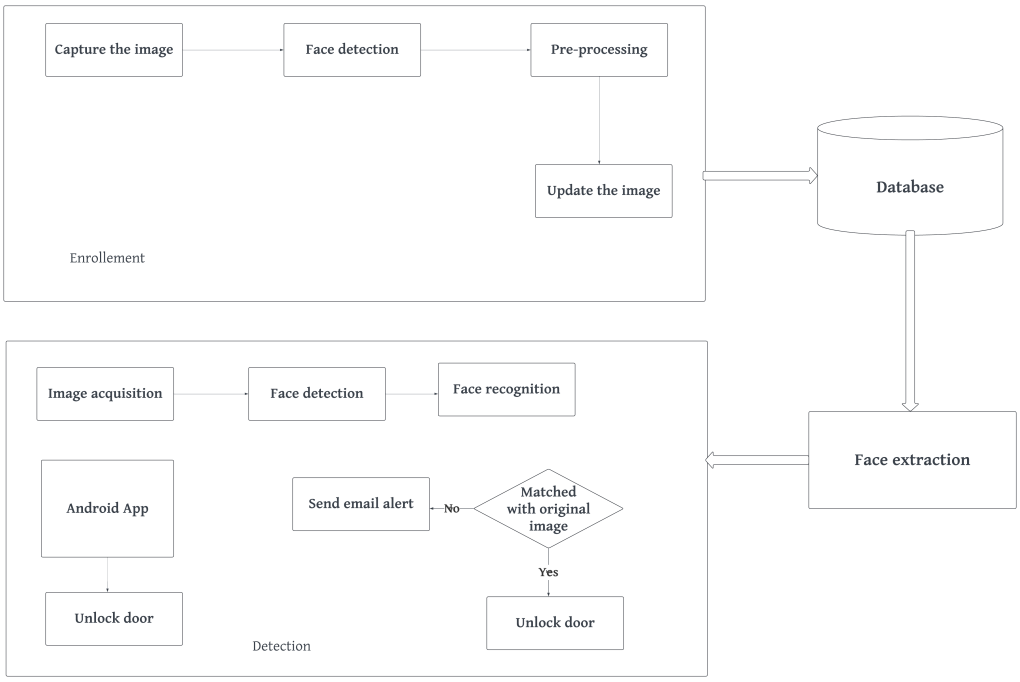


Figure 2.6: La solution proposée

Tout d'abord, une image est capturée à partir de la caméra puis pré-traitée. Le système utilise après cela un algorithme de Deep Learning pour détecter un visage dans l'image. Une fois le visage est détecté, l'algorithme PCA est appliqué afin d'extraire les caractéristiques principales.

Ensuite, le même algorithme de Deep Learning est utilisé à nouveau afin d'identifier à présent le visage détecté. Si l'individu est reconnu, la caméra Raspberry Pi signale d'ouvrir la porte sinon, une alerte est envoyée.

Les auteurs ont montré que leur système assure la sécurité, une gestion efficace de l'alimentation et une fonctionnalité transparente.

2.8 Synthèse des articles de recherche

Critères \ Articles	Realtime Intrusion Detection System using OpenCV (2023)	Smart surveillance monitoring system using Machine Learning and Raspberry Pi (2022)	Integration of a Video Surveillance System Into a Smart Home Using the Home Assistant Platform (2022)	A novel approach object detection of video surveillance system using OpenCV (2024)	IOT Powered Smart Doorbell System For Enhanced Home Security And Communication (2024)
Objectif	Développer et implémenter un système de surveillance intelligent	Améliorer l'environnement de surveillance vidéo	Le contrôle d'accès et la vidéosurveillance	Améliorer la sécurité dans des zones à présence humaine irrégulière	Renforcer la sécurité
Les données collectées	Un dataset	Capture vidéo	Capture vidéo	Capture vidéo	Capture vidéo
Moyens de détection	Un système d'alarme sonore en temps réel	Via la caméra Raspberry Pi 4	Home Assistant et MotionEyeOs	Classificateur Haar	Via la caméra Raspberry Pi 4
Algorithmes utilisés	LBP	Haar	/	Haar Cascades, LBPH, PCA, SVM	Algorithme de Deep Learning

Table 2.1: Tableau récapitulatif des articles de recherche

2.9 Conclusion

En conclusion, ce deuxième chapitre nous a permis de comprendre le fonctionnement d'un système de vidéosurveillance pour la détection d'intrusion dans une maison intelligente, permettant ainsi une surveillance en temps réel des tentatives d'intrusion. Cette technologie contribue à dissuader les intrus et à alerter les propriétaires de manière rapide et efficace. Nous avons fait un état de l'art des différentes méthodes utilisées dans ce contexte, ainsi, nous avons expliqué différentes techniques d'authentification qui permettent de détecter une quelconque activité suspecte.

Dans le prochain chapitre, nous aborderons le Machine Learning ainsi que les différents algorithmes existants pour résoudre le problème de la reconnaissance faciale dans le cadre d'un système de détection d'intrusion dans une maison intelligente par vidéosurveillance. Dans ce contexte, nous proposerons une approche qui combinera plusieurs algorithmes de Machine Learning et améliorera le traitement de détection d'intrusion. Nous discuterons également des résultats obtenus et nous ferons une comparaison entre les algorithmes choisis.

Chapitre 3

Implémentation d'algorithmes de Machine Learning pour la détection d'intrusion dans une maison intelligente

3.1 Introduction

De plus en plus utilisé et largement répandu, le Machine Learning connaît un essor extraordinaire. Il est donc tout à fait natruel qu'on souhaite l'adopter afin de sécuriser notre environnement et dans notre cas, une maison intelligente.

Dans ce chapitre, nous commençons par poser les bases du Machine Mearning, ses différentes techniques et algorithmes puis nous explorons son application dans le domaine de la reconnaissance faciale en utilisant deux techniques de classification supervisée : le Support Vector Machine (SVM) et l'Analyse Discriminante Linéaire (LDA). La reconnaissance faciale est une technologie clé dans de nombreux domaines tels que la sécurité, la biométrie, et l'interaction homme-machine, nécessitant une grande précision et fiabilité pour être efficace.

Le but de ce chapitre est de comparer les performances des modèles SVM et LDA après une réduction de la dimensionnalité par l'Analyse en Composantes Principales (PCA). Cette réduction est cruciale pour traiter efficacement les données d'image de haute dimension et améliorer la performance des algorithmes de classification.

Nous avons également évalué les performances des modèles en redimensionnant les images à quatre tailles différentes pour analyser l'impact de la taille des images sur la précision de la classification.

3.2 Techniques d'apprentissage automatique ou Machine Learning

Le Machine Learning ou apprentissage automatique en français, est un domaine de l'intelligence artificielle (IA) qui permet aux systèmes d'apprendre et de s'améliorer automatiquement à partir de l'expérience sans être explicitement programmés [37].

Il existe plusieurs techniques utilisées dans ce domaine pour résoudre divers problèmes, on

cite quatre techniques : la classification, la régression, le clustering et l'optimisation. Dans les sous-sections suivantes, nous allons expliquer chaque principe cité ci-dessus.

3.2.1 La classification

La classification est utilisée afin de grouper des données en catégories connues, c'est à dire catégoriser des données dans des classes prédéfinies. La classification utilise un algorithme pour attribuer avec précision des données de test à des catégories particulières. L'algorithme reconnaît des entités spécifiques dans le jeu de données et tente de tirer des conclusions sur la façon dont ces entités doivent être étiquetées ou définies [12]. Il existe plusieurs algorithmes de Machine Learning basés sur cette techniques tels que SVM, les arbres de décision, k-Means...etc.

3.2.2 La régression

La régression permet de comprendre la relation entre les variables dépendantes et indépendantes. Elle est couramment utilisée pour établir des projections, telles que le chiffre d'affaires d'une entreprise donnée [12].

Il existe plusieurs types de régression, chacune adaptée à des types spécifiques de données et de problèmes, comme la régression linéaire (simple ou multiple), la régression logistique pour la classification binaire, ou encore des formes plus complexes comme la régression polynomiale ou non linéaire pour capturer des relations plus complexes entre les variables.

- **La régression linéaire** : La régression linéaire est utilisée pour identifier la relation entre une variable dépendante et une ou plusieurs variables indépendantes et est généralement utilisée pour faire des prédictions sur les résultats futurs. Lorsqu'il n'y a qu'une seule variable indépendante et une seule variable dépendante, on parle de régression linéaire simple. Lorsque le nombre de variables indépendantes augmente, on parle de régression linéaire multiple. Pour chaque type de régression linéaire, on cherche à tracer une ligne de meilleur ajustement, qui est calculée par la méthode des moindres carrés. Cependant, contrairement aux autres modèles de régression, cette ligne est droite lorsqu'elle est tracée sur un graphique [12].

  On peut l'utiliser par exemple pour prévoir les ventes futures d'un produit en fonction de facteurs comme la publicité et le prix ou évaluer l'effet de l'expérience et de l'éducation sur le salaire d'une personne.

  La figure 3.1 est un exemple pour illustrer la régression linéaire :

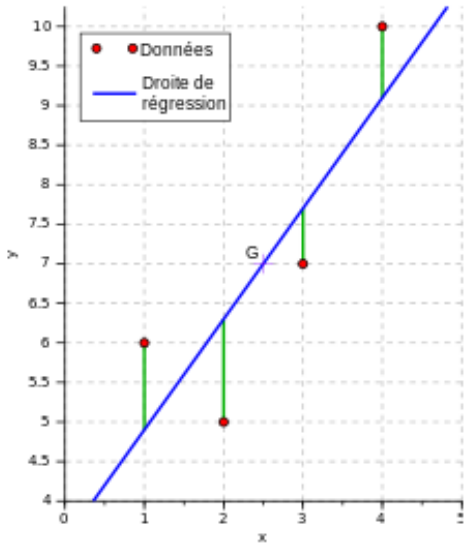


Figure 3.1: La régression linéaire

- **La régression logistique** : La régression logistique est choisie lorsque la variable dépendante est catégorique, c'est-à-dire qu'elle a des résultats binaires, tels que « true » et « false » ou « yes » et « no ». Bien que les deux modèles de régression cherchent à comprendre les relations entre les entrées de données, la régression logistique est principalement utilisée pour résoudre les problèmes de classification binaire, par exemple l'identification du spam [12]. La figure 3.2 nous explique en génarel ce qu'est la régression logistique :

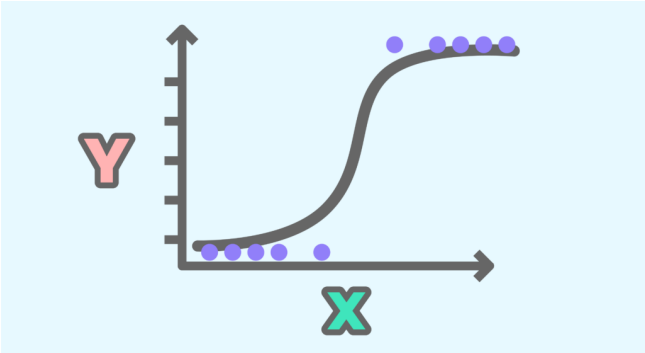


Figure 3.2: La régression ogistique

3.2.3 Le clustering

Le clustering permet de séparer les données en des groupes homogènes ayant des caractéristiques identiques (les clusters). Les algorithmes de clustering placent les points de données dans différents clusters sans connaître la nature des points de données. L'objectif est de partitionner les données de manière à ce que les observations au sein d'un même cluster soient plus similaires entre elles qu'avec celles des autres clusters [25].

3.2.4 L'optimisation

En apprentissage automatique, l'optimisation fait référence au processus de recherche des meilleurs paramètres d'un modèle ou de la meilleure valeur d'une fonction objectif, généralement sous des contraintes spécifiques. Il existe deux types d'optimisation : l'optimisation sans contrainte et l'optimisation contrainte [33]. Nous pouvons donner quelques exemples de techniques d'optimisation telles que l'optimisation de descente de gradient, l'optimisation stochastique, l'optimisation de recherche stochastique, etc. L'optimisation joue un rôle central dans l'amélioration des performances des modèles et des systèmes en ajustant les paramètres de manière à maximiser les résultats souhaités, ce qui en fait un outil important pour de nombreuses applications de science des données et d'ingénierie [33].

3.3 Les types d'apprentissage automatique et les algorithmes utilisés

L'apprentissage automatique est un ensemble de méthodes et d'algorithmes qui permettent à un système informatique d'améliorer ses performances sur une tâche spécifique grâce à l'analyse de données. On trouve plusieurs types d'apprentissage automatique parmi lesquelles on peut citer notamment : l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage semi-supervisé et l'apprentissage par renforcement.

Dans les sous-sections suivantes, nous allons expliquer chacun des types cités ci-dessus ainsi que les algorithmes utilisés dans chaque catégorie.

3.3.1 L'apprentissage automatique supervisé

Il est une sous-catégorie de l'apprentissage automatique. Il consiste à utiliser des jeux de données annotées pour entraîner les algorithmes à classer les données ou à prédire les résultats avec précision [12]. C'est à dire que L'algorithme est entraîné sur un ensemble de données étiquetées, ce qui signifie que chaque exemple d'entraînement est associé à une réponse correcte. Le modèle apprend à prédire la sortie à partir des entrées.

Par exemple, étant donné un large nombre de courriels étiquetés comme étant du spam, on peut entraîner un classifieur de spams qui essaie de prédire si un nouveau message entrant est un spam ou non.

Avec l'apprentissage supervisé, on peut résoudre deux types de problèmes : les problèmes de classification et les problèmes de régression.

3.3.1.1 Les algorithmes utilisés dans l'apprentissage supervisé

L'apprentissage supervisé utilise plusieurs algorithmes afin d'entraîner les modèles à faire des prédictions, nous allons citer les plus utilisés.

- **Machines à vecteurs de support (SVM) :** Une machine à vecteurs de support est un modèle d'apprentissage supervisé populaire développé par Vladimir Vapnik, utilisé à la fois pour la classification et la régression des données. Cela dit, il est généralement utilisé pour les problèmes de classification, en construisant un hyperplan où la distance entre deux classes de points de données est maximale. Cet hyperplan est connu sous le nom de limite de décision, séparant les classes de points de données (par exemple, oranges et pommes) de chaque côté du plan [12].

  La figure 3.3 suivante récapitule le fonctionnement de SVM :

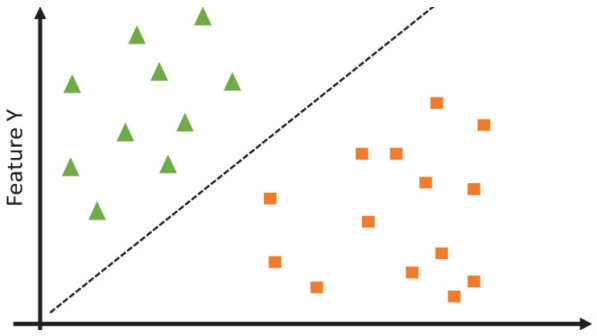


Figure 3.3: SVM

Particulièrement appréciées pour leur capacité à créer des frontières de décision optimales entre les classes, on utilise les SVM pour par exemple détecter des fraudes dans des transactions financières ou identifier des défaillances dans des systèmes industriels, analyser les avis des clients sur des produits ou des services...etc

- **K plus proche voisin (kNN) :** Le k plus proche voisin, également connu sous le nom d'algorithme kNN, est un algorithme non paramétrique qui classe les points de données en fonction de leur proximité et de leur association avec d'autres données disponibles. Cet algorithme suppose que des points de données similaires peuvent être trouvés à proximité les uns des autres. En conséquence, il cherche à calculer la distance entre les points de données, généralement au moyen de la distance euclidienne, puis attribue une catégorie en fonction de la moyenne ou de la catégorie la plus fréquente.

  Ceci est montré dans la figure 3.4 suivante :

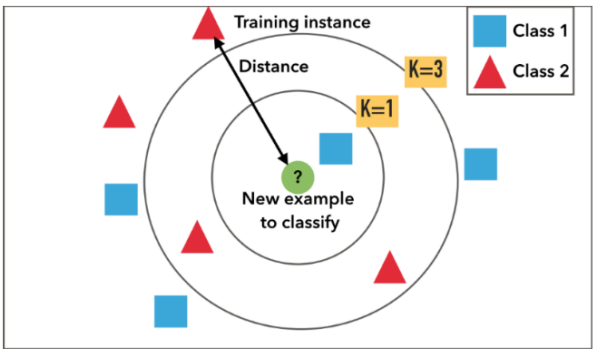


Figure 3.4: kNN

Le kNN est apprécié pour sa simplicité et son efficacité, en particulier pour les ensembles de données de petite à moyenne taille. On l'utilise par exemple pour classer des e-mails en spam ou non-spam, ou identifier le genre musical d'une chanson, estimer le prix d'une maison en fonction des prix des maisons similaires dans la même zone géographique...etc

- **Les cascades de Haar :** Les cascades de Haar font référence à une méthode de détection d'objets dans des images, popularisée par Viola et Jones. Cette méthode utilise des caractéristiques de Haar, qui sont des filtres locaux permettant de mesurer les variations de luminosité dans une image.

  Les classificateurs Cascade de Haar sont un moyen efficace pour la détection d'objets. Cette méthode a été proposée par Paul Viola et Michael Jones en 2001 dans leur article Rapid Object Detection using a Boosted Cascade of Simple Features. Haar Cascade est une approche basée sur l'apprentissage automatique où de nombreuses images positives et négatives sont utilisées pour former le classificateur [19].

  La figure 3.5 nous montre les cascades de Haar :

  - **Images positives** Ces images contiennent les images que l'on veut que le classificateur identifie [19].
  - **Images négatives** Images de tout le reste, qui ne contiennent pas l'objet que l'on veut détecter [19].

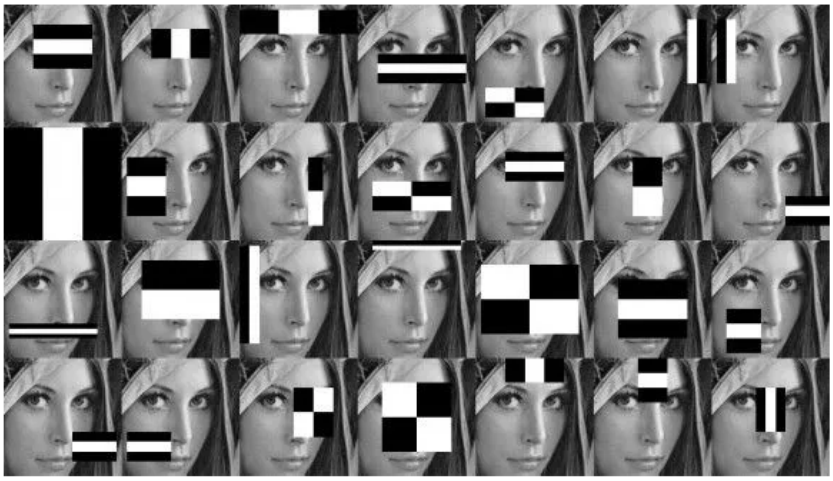


Figure 3.5: Cascades de Haar

- **Analyse discriminante linéaire LDA :** L'algorithme LDA est né des travaux de Belhumeur et al. En 1997 à l'université de Yale. Egalement connu sous le nom de "Fisherfaces" lorsqu'il est utilisé pour la reconnaissance de visage, cet algorithme effectue une séparation de classes, c'est à dire qu'il cherche des directions dans l'espace des caractéristiques qui maximisent la distance entre les moyennes des classes tout en minimisant la

dispersion au sein de chaque classe. Cette capacité à séparer véritablement les classes rend LDA particulièrement utile pour les tâches de classification [39].

L'objectif principal de LDA est de projeter les données dans un espace de dimension inférieure où les classes sont bien séparées. Pour ce faire, LDA calcule les moyennes et les covariances des classes, puis trouve les directions (vecteurs propres) qui maximisent la variance entre les classes tout en minimisant la variance au sein des classes. Ces directions sont ensuite utilisées pour projeter les données, facilitant ainsi la classification en maximisant la séparation entre les différentes classes dans le nouvel espace [39].

Afin de mieux illustrer notre propos, supposons que nous ayons des données avec deux classes A et B et que les données des classes se chevauchent dans l'espace des caractéristiques d'origine. En utilisant LDA, il est possible de trouver une direction de projection qui maximise la séparation entre les moyennes des classes A et B tout en minimisant la variance au sein de chaque classe. Ainsi, les données projetées sur cette nouvelle direction montrent une séparation claire entre les classes A et B.

3.3.2 L'apprentissage automatique non supervisé

Il est une sous-catégorie de l'apprentissage automatique. Il utilise des algorithmes pour analyser et regrouper des jeux de données non étiquetés. Ces algorithmes découvrent des modèles cachés ou des groupements de données sans nécessiter d'intervention humaine [13]. Dans de cas, l'apprentissage par la machine se fait de façon totalement autonome. Des données sont communiquées à la machine sans lui fournir les exemples de résultats attendus en sortie. Elle propose ainsi des réponses à partir d'analyses et de groupement de données.

L'apprentissage non supervisé utilise plusieurs techniques telles que le clustering et la réduction de dimensionnalité. Pour bien comprendre les algorithmes utilisés dans l'apprentissage non supervisé, nous devons d'abord expliquer certaines notions telles la réduction de dimensionnalité.

- La réduction de la dimensionnalité est une technique utilisée lorsque le nombre d'entités, ou dimensions, dans un jeu de données défini est trop élevé. Il réduit le nombre d'entrées de données à une taille gérable tout en préservant autant que possible l'intégrité du jeu de données. Il est couramment utilisé dans l'étape de pré-traitement des données [13].

3.3.2.1 Les algorithmes utilisés dans l'apprentissage non supervisé

Les algorithmes d'apprentissage non supervisé sont des outils puissants utilisés pour explorer et analyser des données sans étiquettes, nous allons citer les algorithmes les plus utilisés.

- **k-Means :** k-Means est un algorithme basé sur la distance. Il essaie de regrouper les points les plus proches en plusieurs groupes [40]. K représente le nombre de groupes selon la distance par rapport au centroïde de chaque groupe. Les points de données les plus proches d'un centroïde donné seront regroupés dans la même catégorie [13].

La figure 3.6 illustre le fonctionnement de k-Means :

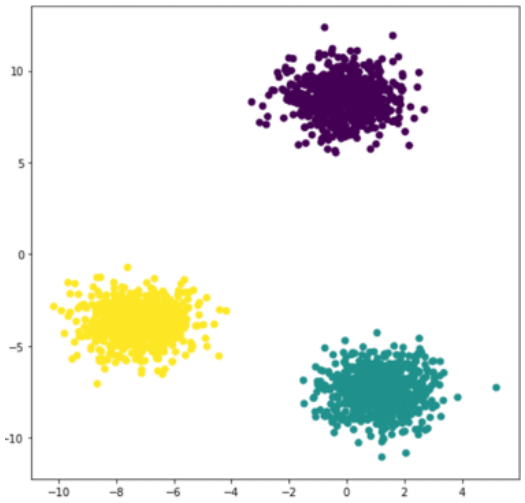


Figure 3.6: k-Means

On utilise cet algorithme par exemple dans la segmentation de marché, l'analyse d'image, le regroupement de documents, la détection de fraudes, la segmentation de réseaux sociaux, et l'analyse des gènes en bioinformatique.

- **L'analyse en composantes principales (PCA) :** PCA est un type d'algorithme de réduction de dimensionnalité utilisé pour réduire les redondances et compresser les jeux de données via l'extraction des caractéristiques. Cette méthode utilise une transformation linéaire pour créer une nouvelle représentation des données et produire un ensemble de « composantes principales ». La première composante principale est la direction qui maximise la variance du jeu de données. Bien que la deuxième composante principale trouve également la variance maximale dans les données, elle est totalement décorrélée de la première composante principale, ce qui donne une direction perpendiculaire, ou orthogonale, à la première composante. Ce processus se répète en fonction du nombre de dimensions, où la composante principale suivante est la direction orthogonale aux composantes précédentes avec le plus de variance [13].

  La figure 3.7 suivante nous résume ce que nous venons de dire :

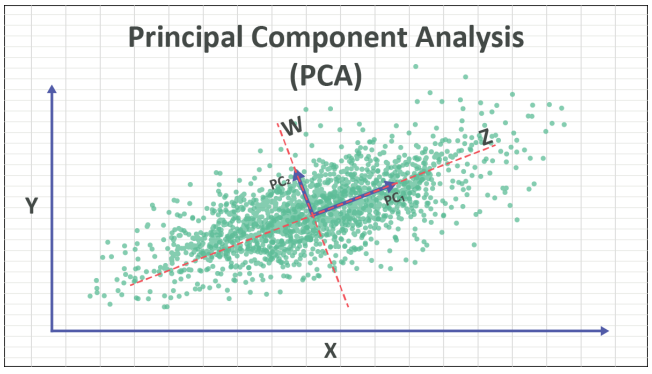


Figure 3.7: PCA

3.3.3 L'apprentissage automatique semi-supervisé

L'apprentissage semi-supervisé est une technique d'apprentissage automatique hybride qui utilise une combinaison de données étiquetées et non étiquetées [15].

Si l'apprentissage semi-supervisé est généralement utilisé sur les mêmes cas d'utilisation que les méthodes d'apprentissage supervisé, il se distingue car il intègre, en plus des données étiquetées requises à l'apprentissage supervisé, des données non étiquetées dans l'entraînement de modèles [14].

Les méthodes d'apprentissage semi-supervisées sont particulièrement utiles dans les situations où il est difficile ou coûteux d'obtenir un volume suffisant de données étiquetées, mais où les données non étiquetées sont relativement faciles à acquérir en revanche. Dans de tels cas, ni les méthodes d'apprentissage entièrement supervisées ni non supervisées ne fourniront des solutions adéquates [14].

3.3.4 Apprentissage automatique par renforcement

L'apprentissage par renforcement est un procédé d'apprentissage automatique consistant, pour un système autonome, à apprendre les actions à réaliser, à partir d'expériences, de façon à optimiser une récompense quantitative au cours du temps [5].

Le système est plongé au sein d'un environnement, et prend ses décisions en fonction de son état courant. En retour, l'environnement procure une récompense, qui peut être positive ou négative [5].

3.4 Les étapes d'exécution d'un algorithme de Machine Learning

3.4.1 La définition du problème

Avant de commencer toute opéaration, il est primordial d'identifier clairement les besoins et les objectifs du modèle de Machine Learning que l'on souhaite construire afin de déterminer si le problème est supervisé ou non.

Il est également important de comprendre la nature des données à disposition afin de choisir la technique optimale afin de résoudre le problème.

3.4.2 La collecte des données

Etant une phase cruciale, la collecte des données en machine learning implique d'identifier et d'extraire des données pertinentes à partir de diverses sources telles que bases de données, API, fichiers CSV, capteurs, ou encore web scraping. Cette étape nécessite de veiller à la qualité et à la pertinence des données collectées, en s'assurant qu'elles sont représentatives du problème

à résoudre, suffisamment volumineuses et diversifiées, et qu'elles couvrent toutes les variations possibles des entrées.

3.4.3 Le pré-traitement des données

Une fois les données collectées, il faut à présent les préparer à l'utilisation et cette préparation se fait à travers une étape cruciale et indispensable qui est le pré-traitement des données.

Cette étape est un processus qui consiste à préparer les données brutes collectées afin qu'elles soient dans un format adéquat et de haute qualité pour être utilisées dans un modèle de Machine Learning. Plusieurs techniques sont utilisées pour cela. Dans les sous-sous-sections qui suivent, nous allons résumer ces techniques.

3.4.3.1 Le nettoyage des données

Des erreurs telles que des données manquantes ou du bruit non contrôlé peuvent être présentes dans la base de données d'apprentissage. L'étape de nettoyage des données est généralement nécessaire pour filtrer ou corriger les données erronées afin de garantir des données pretes à l'exploitation.

3.4.3.2 Tranfsormation des données

Cette technique implique la conversion des données brutes en un format plus adapté à l'entraînement de modèles. Par exemple, la normalisation ajuste les valeurs numériques pour qu'elles se situent dans une plage spécifique, généralement entre 0 et 1, ce qui peut améliorer la convergence des algorithmes d'apprentissage. La standardisation, quant à elle, modifie les données pour qu'elles aient une moyenne de 0 et un écart-type de 1, ce qui est souvent nécessaire pour les modèles sensibles à l'échelle des caractéristiques, comme les SVM ou les réseaux de neurones. L'encodage des variables catégorielles, comme le One-Hot Encoding, transforme les données catégorielles en format numérique, rendant ces informations exploitables par les algorithmes de machine learning. Ces transformations sont essentielles pour garantir que les données sont homogènes et optimisées pour les étapes d'apprentissage et d'évaluation des modèles.

3.4.3.3 Identification et suppression les doublons

Cette étape consiste à identifier et éliminer les enregistrements répétés dans un jeu de données. Les doublons peuvent introduire un biais et affecter la performance du modèle en faussant les statistiques descriptives et les relations entre les variables. En supprimant ces entrées redondantes, on s'assure que chaque observation est unique, ce qui améliore la qualité et la fiabilité des analyses. Cette étape permet de travailler avec un jeu de données plus précis et représentatif, essentiel pour construire des modèles de machine learning robustes et efficaces.

3.4.3.4 La réduction de la dimensionnalité

Les techniques de réduction de dimensionnalité, comme l'analyse en composantes principales, réduisent les variables considérées, simplifiant ainsi le modèle sans perdre d'informations significatives. Cette méthode peut améliorer les performances du modèle et réduire la complexité des calculs [2].

3.4.4 Le choix du modèle

Cette étape consiste à sélectionner le modèle le plus adapté au problème à résoudre, qu'il s'agisse de classification, de régression, de clustering ou autre. Le modèle est choisi après avoir effectué une comparaison entre les différents algorithmes tels que SVM, kNN...etc en fonction des caractéristiques des données et des objectifs du projet. Le choix final repose sur des critères tels que la performance prédictive, la compléxit et la vitesse d'entrainement.

3.4.5 Entraînement du modèle

L'entraînement du modèle en machine learning est le processus par lequel le modèle apprend à partir des données d'entraînement. Durant cette étape, les algorithmes ajustent les paramètres internes du modèle pour minimiser l'erreur entre les prédictions du modèle et les valeurs réelles des données d'entraînement. Ce processus utilise des techniques d'optimisation, telles que la descente de gradient, pour améliorer la précision du modèle. L'entraînement peut être itératif, impliquant plusieurs passages sur les données (époques), et peut inclure l'utilisation de la validation croisée pour s'assurer que le modèle ne surapprend pas (overfitting) aux données d'entraînement. Une fois le modèle bien entraîné, il peut être évalué sur des données de test pour vérifier sa capacité de généralisation.

3.4.6 Évaluation du modèle

L'évaluation du modèle consiste à mesurer sa performance et sa capacité à généraliser sur des données non vues. Cela se fait en appliquant le modèle à un ensemble de données de test distinct de l'ensemble d'entraînement. Les performances sont ensuite quantifiées à l'aide de métriques spécifiques telles que la précision, le rappel, le F1-score pour la classification, ou la racine carrée de l'erreur quadratique moyenne RMSE (Root Mean Square Error) pour la régression. Ces métriques permettent d'évaluer l'exactitude, la robustesse et l'efficacité du modèle.

L'objectif de l'évaluation est de s'assurer que le modèle fonctionne bien non seulement sur les données d'entraînement, mais aussi sur des données nouvelles, garantissant ainsi une bonne capacité de généralisation. Dans les sous-sous-sections suivantes, nous allons expliquer brièvement quelques unes de ces métriques de performance, mais avant cela il est important de comprendre quelques notions de base telles que le vrai positif (TP), le faux positif (FP), le vrai négatif (TN) et le faux négatif (FN).

- **Le Vrai Positif (TP) :** : les cas dans lesquels le modèle prédit « + » et la sortie réelle était également « + ».
- **Le Faux Positif (FP) :** : les cas dans lesquels le modèle prédit « + » et la sortie réelle était« - ».
- **Le Vrai Négatif (TN) :** les cas dans lesquels le modèle prédit « - » et la sortie réelle était « - ».
- **Le Faux Négatif (FN) :** les cas dans lesquels le modèle prédit « - » et la sortie réelle était « + ».

3.4.6.1 Accuracy (La précision globale)

L'accuracy est le ratio du nombre de prédictions correctes par rapport au nombre total de prédictions. Elle est définie comme donnée dans l'equation 3.1 suivante:

$$\text{Accuracy} = \frac{\text{Nombre total de prédictions correctes}}{\text{Nombre total de prédictions}} \tag{3.1}$$

3.4.6.2 Precision (La précision)

La précision mesure la proportion des exemples pertinents parmi les exemples récupérés, c'est le ratio du nombre de vrais positifs par rapport au nombre total de prédictions positives. Elle est définie dans l'équation 3.2 suivante:

$$\text{Precision} = \frac{\text{Vrai Positifs (TP)}}{\text{Vrai Positifs (TP)} + \text{Faux Positifs (FP)}} \tag{3.2}$$

3.4.6.3 Recall (La sensibilité)

Le Recall est défini comme le taux de vrais positifs ce qui correspond à la proportion de points de données positifs qui sont correctement considérés comme positifs, par rapport à tous les points de données positifs, c'est est le ratio du nombre de vrais positifs par rapport au nombre total d'objets réels. Recall est calculé dans l'équation 3.3 suivante:

$$\text{Recall} = \frac{\text{Vrai Positifs (TP)}}{\text{Vrai Positifs (TP)} + \text{Faux Négatifs (FN)}} \tag{3.3}$$

3.4.6.4 Specificity (La spécificité)

La spécificité est définie comme le taux de vrais négatifs correspondant à la proportion de points de données négatifs qui sont correctement considérés comme négatifs, par rapport à tous les points de données négatifs. Elle est calculée dans l'équation 3.4 suivante:

$$\text{Specificity} = \frac{\text{Vrai Négatif (TN)}}{\text{Vrai Négatif (TN)} + \text{Faux Positif (FP)}} \tag{3.4}$$

3.4.6.5 F1 Score

Le F1 score est la moyenne harmonique de la precision et du recall. Il fournit un équilibre entre les deux métriques et est définie dans l'équation 3.5 suivante:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.5}$$

3.4.6.6 La matrice de confusion

La matrice de confusion est en quelque sorte un résumé des résultats de prédiction pour un problème particulier de classification. Elle compare les données réelles pour une variable cible à celles prédites par un modèle. Les prédictions justes et fausses sont révélées et réparties par classe, ce qui permet de les comparer avec des valeurs définies [17].

3.5 Outils Python pour la programmation d'un algorithme de Machine Learning

Les outils populaires utilisés pour implémenter les algorithmes d'apprentissage automatique sont R, Python, Spark et Mahout. Dans notre travail, nous préférons opter pour l'outil Python car il offre un nombre considérable d'avantages pour le développement et la mise en oeuvre de modèles d'apprentissage automatique tels qu'un large écosystème de bibliothèques spécialement dédiées au machine learning qui fournissement des outils robustes pour construire et entrainer des modèles.

Python est également lisible et possède une syntaxe claire et concise, ce qui facilite grandement l'écriture et la lecture du code, il peut aussi s'intégrer à d'autres langages. En résumé, c'est le choix parfait pour notre implémentation.

3.5.1 Le langage Python

Le langage Python est un langage de programmation open source multi-plateformes et orienté objet. Grâce à des bibliothèques spécialisées, Python s'utilise pour de nombreuses situations comme le développement logiciel, l'analyse de données, ou la gestion d'infrastructures [11].

Grâce à ses nombreuses bibliothèques telles Panda, Numpy, Scipy, Scrapy, Matpotlib, Scikit-Learn ou encore TensorFlow, Python est très largement utilisé dans l'apprentissage automatique, c'est d'ailleurs dans ce contexte là que nous allons utiliser ce langage de programmation.

Dans les sous-sections qui suivent, nous allons détailler les bibliothèques utilisées dans le machine learning.

3.5.1.1 NumPy

Son nom signifie Numerical Python. C'est une bibliothèque incontournable en Python. Elle permet de faire du calcul numérique et permet la gestion des tableaux de données et matrices. Elle possède la plupart des fonctions usuelles telles que l'exponentielle, le logarithme ou encore arctan. De plus, elle est optimisée pour les calculs et va permettre de paralléliser les opérations, c'est-à-dire utiliser tous les processeurs de l'ordinateur pour aller plus vite dans les calculs [1].

Dans l'apprentissage automatique, NumPy manipule et traite efficacement les données. Elle fournit des structures de données puissantes (ndarrays) et des fonctions optimisées pour effectuer des calculs vectorisés, des opérations d'algèbre linéaire et du prétraitement des données, comme la normalisation et la mise à l'échelle. NumPy permet également de générer des données synthétiques pour tester des algorithmes.

La figure 3.8 suivante résume tout ce que peut faire NumPy :

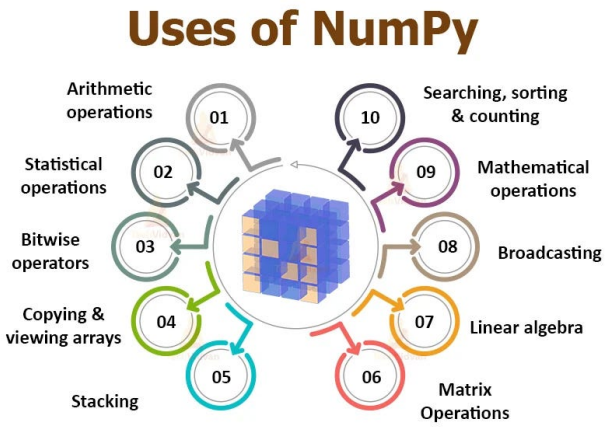


Figure 3.8: NumPy

3.5.1.2 SciPy

De son nom Scientific Python est une bibliothèque utilisée pour le calcul scientifique et technique. Elle s'utilise en synergie avec Numpy puisqu'elle travaille sur des données du format Numpy array. SciPy offre un catalogue d'opérations scientifiques : algèbre linéaire, algorithmes de régression, fonctions statistiques... [7].

En particulier, SciPy permet de travailler sur des projets d'optimisation numérique qui consistent à chercher à obtenir le plus petit chiffre (ou le plus grand) possible en modifiant certaines variables [7].

3.5.1.3 Matplotlib

Matplotlib est une bibliothèque de visualisation de données en Python, largement utilisée en machine learning pour créer des graphiques et des visualisations de haute qualité. c'est en effet cette bibliothèqe qui permet, en une ligne de code, de créer des graphiques qui modélisent les données sur lesquelles on travaille. Matplotlib s'utilise en synergie avec Numpy ou Pandas [7].

Matplotlib offre également une large variété de types de graphes qui s'adaptent à tous les besoins : histogrammes, boîtes à moustache, courbes, scatter plots... [7].

La figure 3.9 suivante nous montre quelques exemples d'utilisation de Matplotlib :

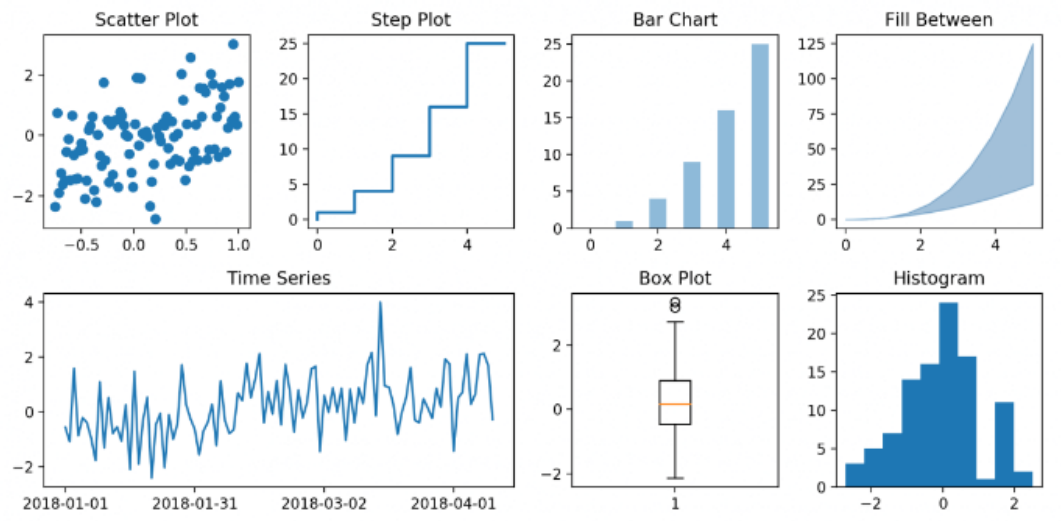


Figure 3.9: Matplotlib

3.5.1.4 Pandas

Pandas est une bibliothèque complète en ce qui concerne la manipulation de données.Elle est spécifiquement conçue pour la manipulation et l'analyse de données en langage Python. Elle est à la fois performante, flexible et simple d'utilisation [8].

Grâce à Pandas, le langage Python permet enfin de charger, d'aligner, de manipuler ou encore de fusionner des données [8].

Elle est basée sur 2 types d'objets : les séries, très similaires aux listes en termes de fonctionnement et les data frames qui sont donc des tableaux à plusieurs colonnes. On peut aussi inclure un type d'objet appelé Panels qui permet de manipuler des objets en 3 ou 4 dimensions [1].

De plus, elle facilite la lecture de données provenant de différentes sources : CSV, SQL ou encore texte. Bref, c'est l'outil incontournable pour manipuler des données sur Python. Cette librairie permet aussi d'avoir une meilleure vue d'ensemble sur les données [1].

On peut illustrer le fonctionnement de la bibliothèque Pandas dans cette figure 3.10 :

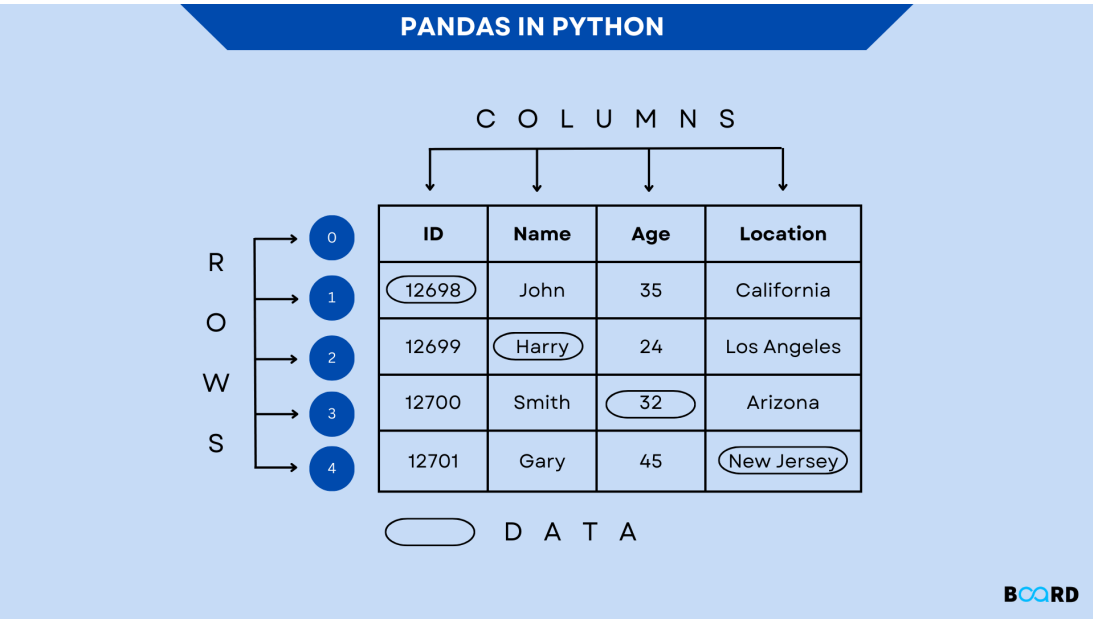


Figure 3.10: Pandas

Dans l'apprentissage automatique, pandas est utilisée dans le prétraitement des données, l'exploration des données, l'ingénierie des caractéristiques et la préparation des données pour l'apprentissage.

3.5.1.5 Scikit-learn

Scikit-learn est une bibliothèque Python spécialement conçue pour l'apprentissage automatique, elle offre un large choix d'algorithmes d'apprentissage supervisé et non supervisé pour la classification, la régression, le clustering, la réduction de dimensionnalité et bien d'autres tâches liées à l'analyse de données.

De plus, cette bibliothèque possède de nombreuses métriques qui permettront d'évaluer la qualité des modèles. Elle permet de gérer de bout en bout la création d'un algorithme d'apprentissage automatique [7].

3.5.1.6 OpenCV

OpenCV signifie Open Source Computer Vision. Elle a été créée pour les applications de Computer Vision et accélérer leur déploiement. Elle prend en entrée de nombreuses entrées visuelles telles que des images et des vidéos. Elle permet par exemple de faire ce qu'on appelle de l'OCR (Optical character recognition) permettant de résoudre les problèmes de reconnaissance de caractère sur une image ou par exemple un PDF. Les tableaux reconnus par OpenCV peuvent ensuite être directement transformés en objets numpy pour les retraiter facilement en Python. De plus, cette librairie s'intègre bien avec Matplotlib [1].

Dans le domaine de l'apprentissage automatique, OpenCV est principalement utilisée pour le prétraitement des données d'images, la création de caractéristiques à partir d'images et l'intégration de la vision par ordinateur dans les pipelines de l'apprentissage automatique.

3.5.1.7 Tkinter

Le paquet tkinter (« interface Tk ») est l'interface Python standard de la boîte à outils d'interface utilisateur graphique GUI (Graphical User Interface) avec Tcl/Tk [22]. Elle permet de développer des applications graphiques de manière simple et efficace.

Tkinter est une bibliothèque puissante et accessible pour la création d'applications GUI en Python. Elle est bien adaptée pour des applications simples et moyennement complexes, offrant un bon équilibre entre fonctionnalité et facilité d'utilisation.

3.5.1.8 Pillow

Anciennement PIL, Pillow est une bibliothèque open source spécifiquement conçue pour le traitement d'images via Python [9].

Contrairement à d'autres librairies ne proposant accessoirement que quelques méthodes de traitement d'image, Pillow est presque exclusivement centrée sur ce domaine. Elle est donc bien plus exhaustive et optimisée pour la manipulation d'images [9].

Pillow est une bibliothèque puissante et flexible pour travailler avec les images en Python, offrant un large éventail de fonctionnalités pour répondre à divers besoins en manipulation d'images.

3.5.1.9 Pywhatkit

Pywhatkit est une bibliothèque Python qui permet d'envoyer des messages WhatsApp à une heure précise via des scripts Pyhton, et qui possède également plusieurs autres fonctionnalités telles que la recherche sur Google, la lecture de vidéos YouTube...etc.

3.6 Reconnaissance de visage à l'aide d'algorithmes de Machine Learning

Dans ce mémoire, nous nous intéressons à la reconnaissance du visage, c'est à dire que nous voulons implémenter un algorithme qui soit capable de reconnaître une personne autorisée (un habitant par exemple) grâce aux traits de son visage et qu'il soit capable de détecter une personne non autorisée (un intrus).

3.6.1 Processus de reconnaissance du visage

La reconnaissance du visage, un domaine clé de la vision par ordinateur et de l'intelligence artificielle, implique un processus complexe et multi-étapes permettant d'identifier ou de vérifier l'identité d'une personne à partir d'une image ou d'une vidéo. Ce processus se déroule en plusieurs étapes clés. Dans les sous-sections suivantes, nous allons détailler les différentes étapes qui constituent ce processus.

3.6.1.1 La détection du visage

La caméra détecte et localise l'image d'un visage, seul ou dans une foule. L'image peut montrer la personne de face ou de profil [18].

3.6.1.2 L'analyse du visage

Ensuite, une image du visage est capturée et analysée. La plupart des technologies de reconnaissance faciale utilisent la 2D plutôt que la 3D, car il est plus pratique de comparer une image en 2D à des photos ou aux images d'une base de données. Le logiciel analyse la géométrie du visage. Les facteurs clés incluent la distance entre les yeux, la profondeur des orbites, la distance entre le front et le menton, la forme des pommettes, ainsi que le contour des lèvres, des oreilles et du menton. Le but est d'identifier les spécificités du visage détecté [18].

3.6.1.3 La conversion de l'image en données

Le processus de capture du visage transforme les informations analogues (un visage) en un ensemble d'informations numériques (les données) selon les caractéristiques du visage de la personne. En fait, l'analyse du visage est transformée en formule mathématique. Le code numérique est appelé une empreinte faciale. De la même manière que les empreintes digitales sont uniques, tout le monde a sa propre empreinte faciale [18].

3.6.1.4 La recherche de la correspondance

L'empreinte faciale est ensuite comparée à une base de données avec d'autres visages connus. Par exemple, le FBI a accès à près de 650 millions de photos sur de nombreuses bases de données d'État. Sur Facebook, les photos avec des personnes identifiées rejoignent leur base de données qui peut aussi être utilisée pour la reconnaissance faciale. Si l'empreinte faciale correspond à une image de la base de données utilisée par la reconnaissance faciale, une correspondance est effectuée [18].

On peut résumer toutes ces étapes dans le schéma de la figure 3.11 suivante :

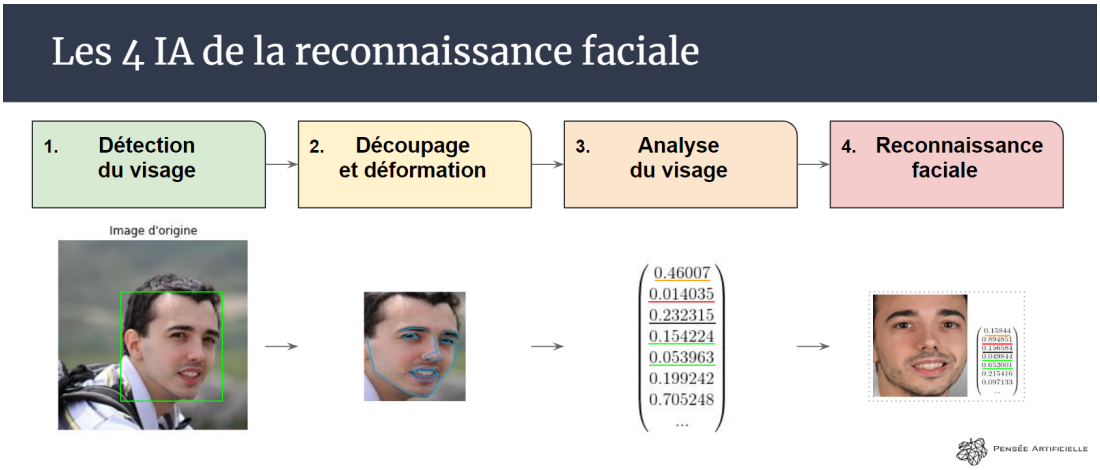


Figure 3.11: Le processus de reconnaissance de visage

3.7 Proposition et évaluation de performances

Ce que nous proposons dans ce projet est l'implémentation d'un système de détection d'instrusion utilisant les algorithmes de Machine Learning PCA, SVM et LDA par reconnaissance du visage à travers des vidéos de surveillance mises en place dans une maison intelligente. Ce système intervient lorsqu'un habitant (une personne autorisée) est absente, elle recevra donc un message d'alerte via WhatsApp qui lui notifiera qu'une intrusion a été détectée.

En fait, en examinant les travaux de recherche de la littérature et les systèmes proposés, les chercheurs se concentrent principalement sur la reconnaissance de la personne physiquement comme principale méthode d'authentification pour la détection d'intrusion mais ils ne prévoient pas d'alertes ou d'actions immédiates lorsque des intrus non autorisés sont détectées, ce qui représente un autre avantage pour notre approche proposée.

Nous pouvons résumer notre proposition dans le schéma de la figure 3.12 suivant :

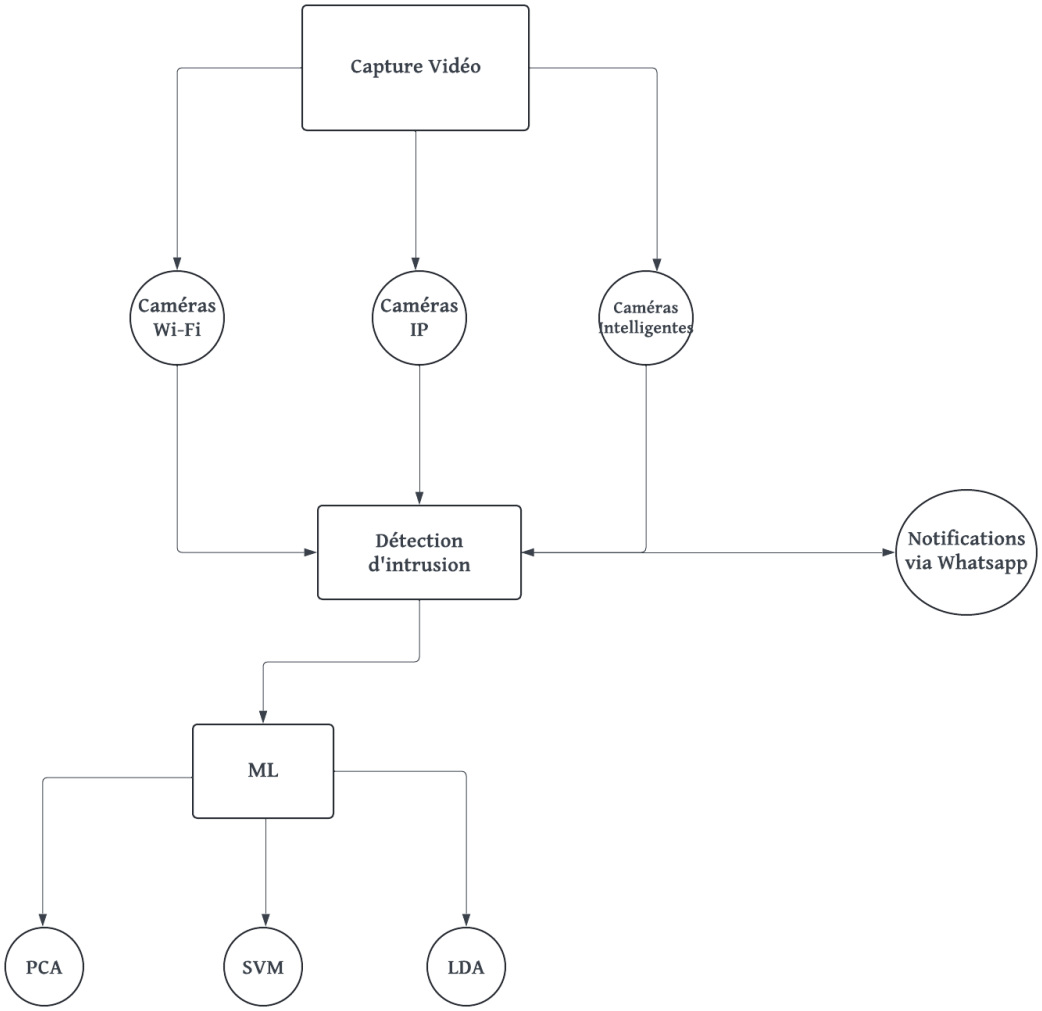


Figure 3.12: Le schéma de la proposition

En suivant les différentes étapes, nous allons expliquer dans les sous-sections suivantes comment nous avons implémenté ces différents algorithmes et les résultats obtenus.

3.7.1 Le choix du Dataset

Dans le cadre de notre projet, nous avons utilisé le dataset Security System Annotation Dataset [23] qui provient du site Roboflow, il a plusieurs cas d'utilisation mais celui qui nous intéresse est lorsqu'il est utilisé pour créer des systèmes de sécurité domestique intelligents capables de distinguer automatiquement les propriétaires des intrus, garantissant ainsi que les alarmes ne soient déclenchées que lorsque cela est nécessaire. Cet ensemble de données comprend trois dossiers : train qui contient les images d'entraînement, valid qui contient les données de validation et test qui contient les données de test, chaque dossier contient à son tour des images représentant le visage de trois filles différentes : Resa, Ruby et Jocelyn, sous différents angles et différentes poses.

Chaque dossier contient un fichier .CSV qui inclut les annotations de chaque image, entre autre le nom de l'image, la taille de l'image et la classe de cette image. Le dossier train englobe en tout 2958 images dont 887 images pour Resa, 1235 images pour Jocelyn et 836 pour Ruby. Le dossier valid englobe quant à lui 366 images dont 109 pour Resa, 153 pour Jocelyn et 104 pour Ruby et pour finir le dossier test inclut 377 images dont pour 114 Resa, 157 pour Jocelyn et 106 pour Ruby.

On remarque qu'il y une claire distinction entre les données de test et de validation, en effet, les données de validation sont utilisées pour ajuster les hyperparamètres du modèle pendant l'entraînement, afin d'optimiser les performances sur des données non vues auparavant. Quant aux données de test, elles sont utilisées pour évaluer les performances finales du modèle une fois que celui-ci a été entraîné et optimisé sur l'ensemble de validation.

En résumé, l'ensemble de validation est utilisé pour optimiser le modèle pendant l'entraînement, tandis que l'ensemble de test est utilisé pour évaluer la performance finale du modèle après l'entraînement. C'est une pratique standard en apprentissage machine pour assurer une évaluation impartiale et fiable des modèles.

Les images regroupées dans les figures 3.13, 3.14, 3.15 suivantes nous montrent les trois filles sous différentes poses :

(a) Pose 1 (b) Pose 2 (c) Pose 3 (d) Pose 4

Figure 3.13: Resa

(a) Pose 1 (b) Pose 2 (c) Pose 3 (d) Pose 4

Figure 3.14: Jocelyn

(a) Pose 1 (b) Pose 2 (c) Pose 3 (d) Pose 4

Figure 3.15: Ruby

Pour utiliser ces images, il faut tout d'abord charger le dataset via Python. Pour ce faire, il a fallu faire appel à plusieurs bibliothèques comme illustré dans la figure 3.16 suivante :

```
import os
import pandas as pd
import cv2
```

Figure 3.16: Les bibliothèques utilisées pour charger le dataset

- **os :** cette bibliothèque est utilisée pour la manipulation des chemins des fichiers.
- **pandas :** cette bibliothèque est utilisée pour la gestion des fichiers CSV.
- **cv2 :** cette bibliothèque est utilisée pour le chargement et le traitement des images.

Une fois les bibliothèques nécessaires au chargement du dataset appelées, la prochaine étape est de charger les images. Nous illustrons cela dans la figure 3.17 :

```python
# Chemin du répertoire de base
base_directory = r'C:\Users\ikhle\Downloads\Security System Annotation.v8i.tensorflow'

# Chemins des fichiers CSV pour train, valid et test
train_csv_path = os.path.join(base_directory, 'train', '_annotations.csv')
valid_csv_path = os.path.join(base_directory, 'valid', '_annotations.csv')
test_csv_path = os.path.join(base_directory, 'test', '_annotations.csv')

# Fonction pour charger les images à partir des chemins spécifiés dans le DataFrame
def load_images(df, base_path):
    images = []
    labels = []
    for index, row in df.iterrows():
        image_path = os.path.join(base_path, row['filename'])  # Chemin de l'image
        image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
        if image is not None:
            images.append(cv2.resize(image, (200, 200)))  # Redimensionner à 200x200
            labels.append(row['class'])  # Récupérer la classe de l'image
    return images, labels

# Charger les images et les étiquettes d'entraînement par classe
def load_images_by_class(csv_path, base_directory):
    df = pd.read_csv(csv_path)
    images_by_class = {}
    for girl in ['Resa', 'Jocelyn', 'Ruby']:
        images_by_class[girl] = df[df['class'] == girl]
        images_by_class[girl], _ = load_images(images_by_class[girl], base_directory)
    return images_by_class

# Charger les images par classe
train_images = load_images_by_class(train_csv_path, os.path.join(base_directory, 'train'))
valid_images = load_images_by_class(valid_csv_path, os.path.join(base_directory, 'valid'))
test_images = load_images_by_class(test_csv_path, os.path.join(base_directory, 'test'))
```

Figure 3.17: Chargement du dataset

- **Chargement du répertoire de base :** `base_directory` stocke le chemin de base où se trouvent tous les fichiers du dataset.
- **Chemins des fichiers CSV pour train, valid et test :** Les lignes du code précedent construisent les chemins complets vers les fichiers CSV contenant les annotations des images pour les ensembles d'entraînement (`train`), de validation (`valid`) et de test (`test`). `os.path.join` est utilisé pour assembler les différents segments de chemin en un chemin compatible avec le système d'exploitation.
- **Fonction pour charger les images à partir des chemins spécifiés dans le Dataframe :** La fonction `load_images` est définie pour charger les images à partir des chemins spécifiés dans un DataFrame pandas, en utilisant OpenCV pour lire chaque image et la redimensionner en niveaux de gris à une taille de 200x200 pixels.
- **Charger les images et les étiquettes d'entrainement par classe :** La fonction `load_images_by_class` est définie pour charger les images par classe à partir d'un fichier CSV spécifié. Elle utilise `pd.read_csv` pour lire le fichier CSV, filtre les données par classe ('Resa', 'Jocelyn', 'Ruby'), puis appelle la fonction `load_images` pour charger les images correspondantes.

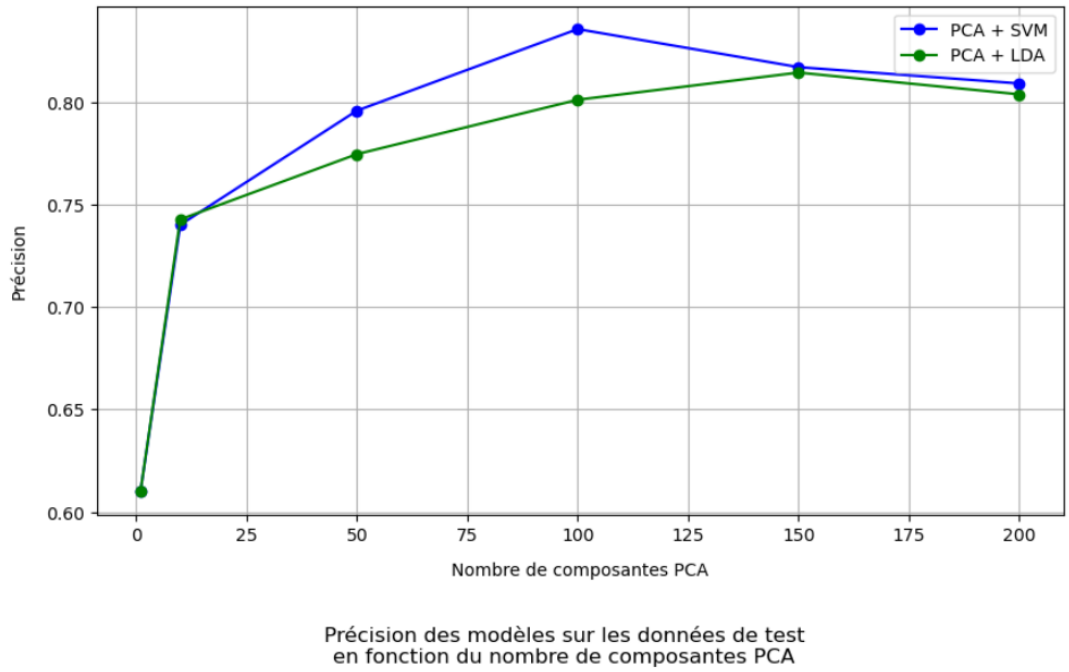
3.7.2 Le pré-traitement des données

Une fois les images chargées, il est obligatoire à présent de les pré-traiter afin qu'elles puissent etre utilisées dans le modèle. Pour cela il faut :

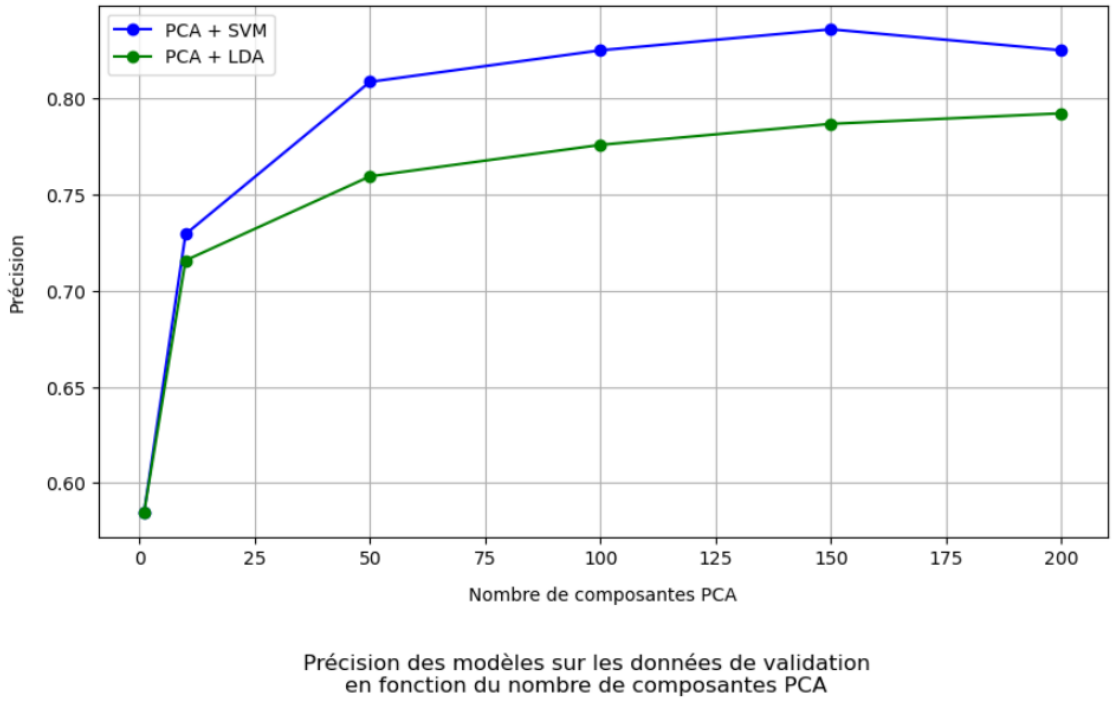
- **Normaliser les données :** La normalisation des données est réalisée en utilisant le StandardScaler de la bibliothèque scikit-learn. Le StandardScaler ajuste les données pour qu'elles aient une moyenne de 0 et un écart type de 1. Cette étape est cruciale pour assurer que les données sont comparables et que les algorithmes de machine learning fonctionnent efficacement.

- **Appliquer PCA sur les données d'entrainement:** Le but de l'utilisation de PCA dans notre projet est l'extraction de composantes principales à partir des images d'origine et lorsque cet algorithme est appliqué aux images, chaque composante principale peut être vue comme une "eigenface" ou un motif de base extrait des images de visages. Ces eigenfaces sont des images aux allures fantomatiques construites à partir des vecteurs propres et représentent des caractéristiques saillantes comme des contours, des ombres, etc.

  Pour ce qui est du nombre de caractéristiques principales à sélectionner, nous avons tout d'abord sélectionné deux algorithmes de Machine Learning : SVM et LDA à utiliser afin de faire de la reconnaissance faciale et nous avons évalué la précision des deux modèles par rapport aux données de test et de validation afin de choisir le nombre optimal de caractéristiques à extraire. Les courbes suivantes nous montrent la précision sur les données de test et de validation en fonction des nombres de caractéristiques extraites. Ce qui est résumé dans la figure 3.18 suivante :



(a) Précision des modèles sur les données de test en fonction du nombre de composantes PCA



(b) Précision des modèles sur les données de validation en fonction du nombre de composantes PCA

Figure 3.18: Précision des modèles en fonction du nombre de composantes PCA

Nous décidons d'opter pour 100 composantes principales car sur les données de test, nous obtenons une précision de plus de 80%.

- **Afficher les Eigenfaces:** Après avoir à présent extrait les Eigenfaces, nous allons les afficher afin de bien illustrer nos propos. Ces eigenfaces seront ensuite utilisées par les

modèles SVM et LDA afin de les entrainer et faire des prédictions. Les figures 3.19, 3.20 et 3.21 suivantes montrent 10 Eigenfaces pour chaque fille.

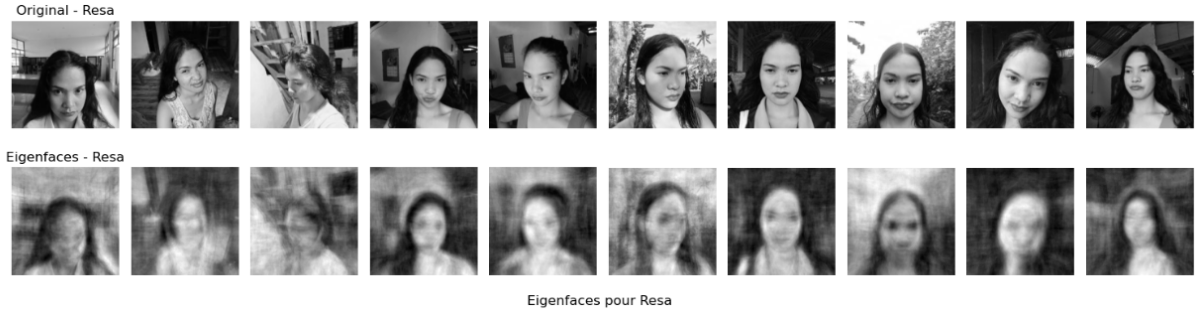


Figure 3.19: Eigenfaces pour Resa

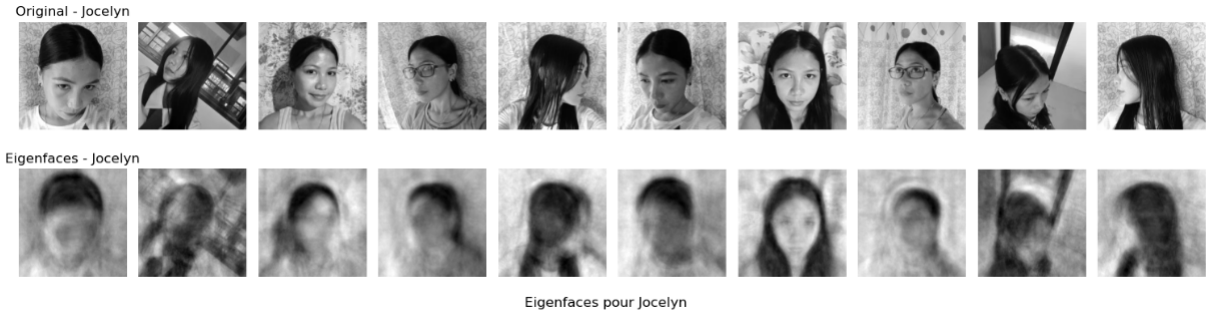


Figure 3.20: Eigenfaces pour Jocelyn

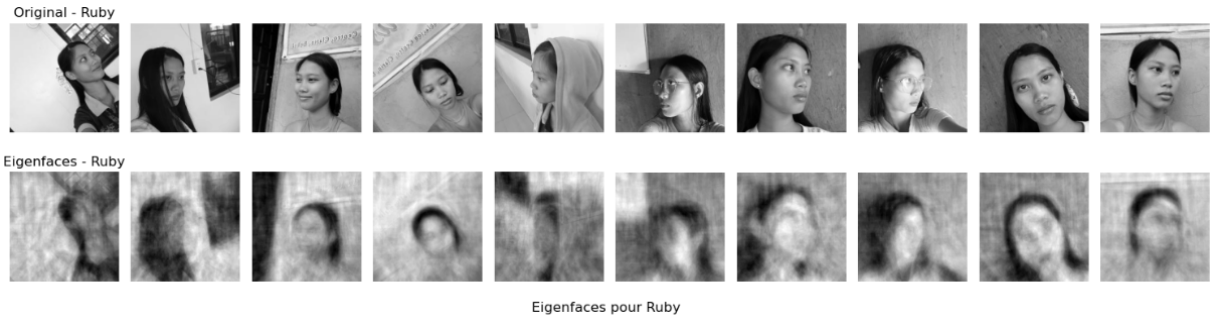


Figure 3.21: Eigenfaces pour Ruby

3.7.2.1 Les personnes autorisées

Dans notre projet, nous avons décidé de subdivisé notre dataset composé de trois jeunes filles en deux sous-datasets bien distincts. Le premier sous-dataset est constitué de Resa que nous avons décidé de considérer comme la seule personne autorisée (l'habitante légtitime par exemple) tandis que Jocelyn et Ruby sont toutes deux considérées comme intruses et cela pour avoir une idée précise de comment appliquer notre système de reconnaissance faciale.

3.7.3 Entrainement du modèle SVM

Cette phase d'entraînement du modèle joue un rôle crucial pour assurer des performances précises et fiables. L'objectif est de former un modèle capable de séparer efficacement les différentes

classes de visages basé sur ces caractéristiques. Voici comment cette phase d'entraînement est mise en œuvre et évaluée dans le contexte de notre projet de reconnaissance faciale.

3.7.3.1 Initialisation du SVM

La première étape est d'initialiser le SVM en spécifiant un noyau linéaire à l'aide de SVC(kernel='linear'). Cela configure le SVM pour séparer linéairement les classes dans l'espace de caractéristiques résultant de l'analyse en composantes principales (PCA).

Nous avons choisi un noyau linéaire pour sa simplicité, sa compatibilité avec PCA ainsi que sa rapidité d'exécution.

3.7.3.2 Entraînement

Le modèle est ensuite entraîné en utilisant les données d'entraînement transformées par PCA (les eigenfaces) en utilisant `train_pca_images` et `train_labels_all` avec la fonction `svm.fit`. Le modèle apprend à séparer les différents visages représentés par les images en fonction de leurs caractéristiques extraites par PCA, comme illustré dans le code présent dans la figure 3.22 qui suit :

```
# Entraîner SVM
svm = SVC(kernel='linear')
svm.fit(train_pca_images, train_labels_all)
```

Figure 3.22: Initialisation et entrainement du modèle SVM

3.7.3.3 Sauvegarde du modèle entrainé

Une fois le modèle SVM entrainé, il est sauvegardé ainsi que le modèle PCA et le scaler utilisés pour la transformation des données. Cela permet de conserver ces modèles pour une utilisation ultérieure.

3.7.3.4 Évaluation sur les données de validation et de test

Pour évaluer le modèle SVM dans le cadre de la reconnaissance faciale, deux étapes clés sont nécessaires : l'évaluation sur les données de validation et sur les données de test. Sur les données de validation, le SVM prédit les étiquettes des images transformées par PCA et compare ces prédictions avec les étiquettes réelles pour calculer sa précision. De manière similaire, sur les données de test, le SVM est évalué en prédisant les étiquettes des images transformées par PCA et en mesurant la précision par rapport aux étiquettes réelles. Ces évaluations permettent de vérifier la capacité du modèle à généraliser sur de nouvelles données après l'entraînement initial.

En gros, l'évaluation sur les données de validation teste la capacité du modèle à généraliser sur des données similaires mais non utilisées pour l'entraînement, tandis que l'évaluation sur les données de test mesure sa performance sur un ensemble indépendant de données inédites.

3.7.4 Entraînement du modèle LDA

3.7.4.1 Initialisation du modèle LDA

La première étape dans ce contexte est de créer une instance de la classe `LinearDiscriminantAnalysis` à l'aide de la bibliothèque scikit-learn déja importée, comme fait dans la figure 3.23 qui suit :

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA

# Initialisation du modèle LDA
lda = LDA()
```

Figure 3.23: Initialisation du modèle LDA

Une fois instanciée, elle est prête à l'utilisation.

3.7.4.2 Entraînement

Les données d'entraînement préalablement transformées par PCA `train_pca_images` sont utilisées pour entraîner le modèle LDA en l'associant avec leurs étiquettes correspondantes `train_labels_all`. L'objectif de LDA est de maximiser la séparation entre les classes tout en minimisant la variance intra-classe. L'entraînement du modèle est fait comme expliqué dans la figure 3.24 suivante.

```
# Entraîner LDA
lda = LDA()
lda.fit(train_pca_images, train_labels_all)
```

Figure 3.24: Entraînement du modèle LDA

3.7.4.3 Sauvegarde du modèle entraîné

Comme pour le modèle SVM, le modèle LDA entraîné est sauvegardé dans un répértoire spécifié à l'aide d'une bibliothèque `pickle` importée au préalable.

3.7.4.4 Évaluation sur les données de validation et de test

L'évaluation sur les données de validation et de test pour le LDA (Linear Discriminant Analysis) suit un processus similaire à celui du SVM. Après avoir entraîné le modèle LDA sur les données d'entraînement transformées par PCA, les prédictions sont générées pour les ensembles de validation et de test. La précision du modèle est calculée en comparant ces prédictions avec les étiquettes réelles respectives. Cela permet d'évaluer la capacité du LDA à discriminer efficacement entre les classes d'images de visages, tant pour les données de validation utilisées pendant l'entraînement que pour les données de test non utilisées précédemment, assurant ainsi une évaluation complète de sa performance dans des contextes réels de reconnaissance faciale.

3.7.5 Les résultats obtenus

Après voir entraîné les modèles et fait des prédictions, nous avons décidé d'utiliser une interface graphique utilisateur GUI avec python afin de pouvoir visualiser les résultats obtenus avec nos deux modèles et ainsi avoir une idée plus précise de comment fonctionne la reconnaissance faciale. L'intégration de cette interface enrichit significativement l'expérience utilisateur en offrant une plateforme intuitive pour interagir avec le système. Cette GUI permet non seulement de sélectionner et de visualiser des images mais aussi d'appliquer notre modèle de reconnaissance faciale en temps réel. Cela simplifie l'accès à des fonctionnalités avancées telles que la prédiction des visages et l'affichage des résultats directement à l'écran, rendant ainsi notre système plus accessible et convivial pour diverses applications.

3.7.5.1 L'interface GUI

En lançant l'exécution du programme, l'interface graphique permettant la détection d'intrusion apparaît sur la figure 3.25 suivante.

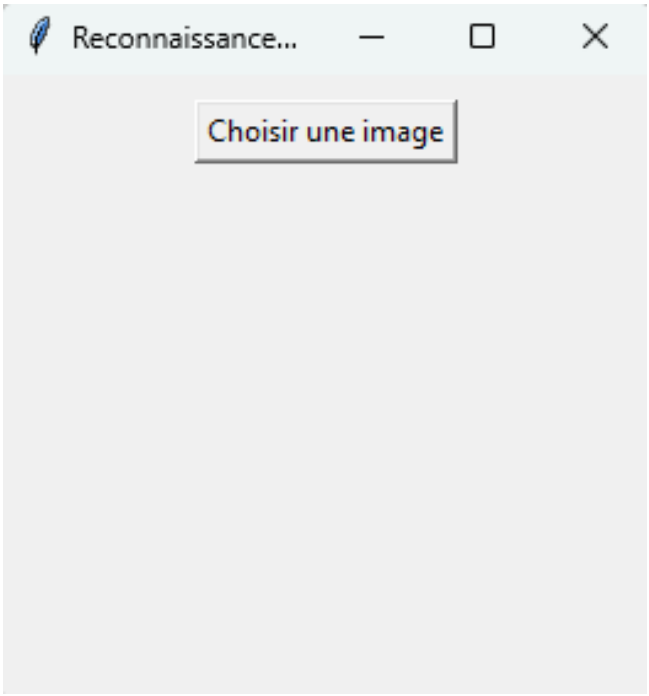
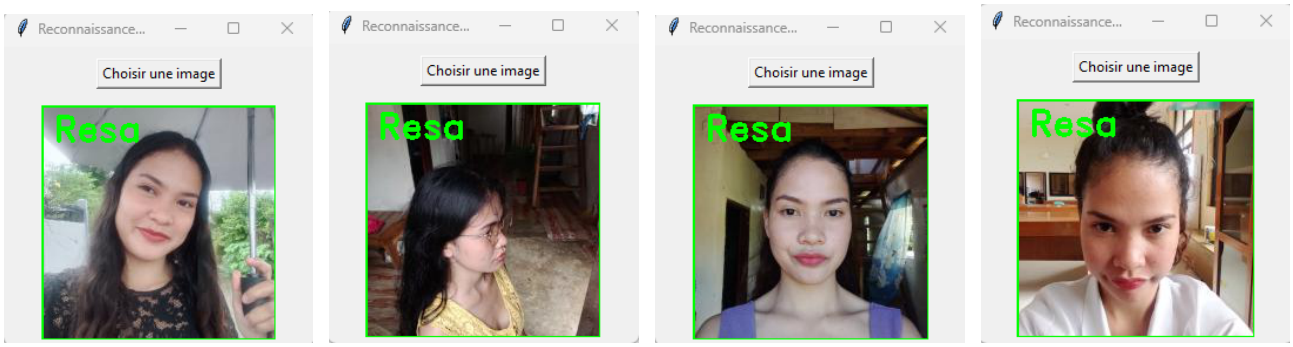


Figure 3.25: L'interface GUI

Elle se constitue d'un bouton 'choisir une image', lorsqu'on clique dessus, on choisit une image à partir du dossier test. Nous aurons dans ce cas deux options comme on peut le voir dans les figures qui suivent :

- **Resa est reconnue :**

  Dans ce cas, le système affiche l'image de Resa ainsi que son prénom en vert pour signaler que c'est bien une personne autorisée, ceci est résume dans la figure 3.26 suivant :

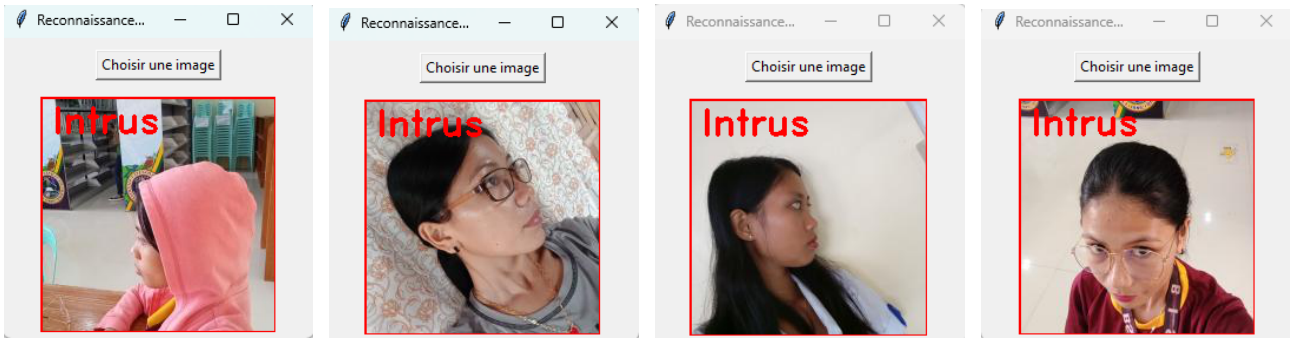


(a) Resa est détectée (b) Resa est détectée (c) Resa est détectée (d) Resa est détectée

Figure 3.26: Le système reconnaît Resa

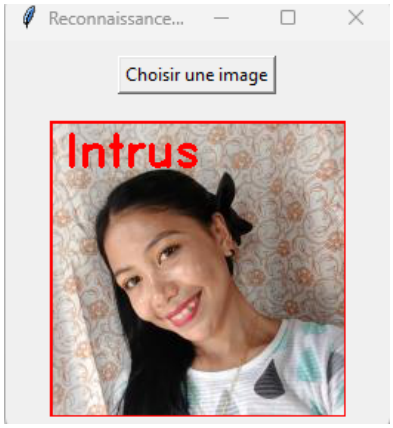
- **Un intrus est détecté :**

  Lorsqu'un intrus est détecté, l'image affiche intrus en rouge et un message Whatsapp est envoyé pour signaler qu'une personne non autorisée a été détectée comme on peut le voir dans la figure 3.27. Le message est une alerte, on peut le voir dans la figure 3.28, où nous avons reçu une notification d'alerte sur Whatsapp.

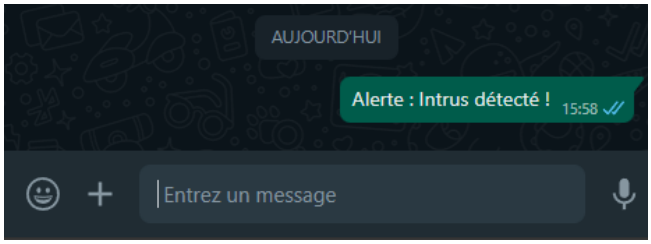


(a) Un intrus est détecté (b) Un intrus est détecté (c) Un intrus est détecté (d) Un intrus est détecté

Figure 3.27: Le système détecte un intrus



(a) Un intrus est détecté



(b) Le message d'alerte est envoyé au propriétaire de la maison sur Whatsapp

Figure 3.28: Le système envoie une alerte

3.7.6 Évaluation de performances

Nous allons évaluer les deux approches PCA + SVM et PCA + LDA.

3.7.6.1 Pour SVM

Nous avons calculé quelques métriques de performance et avons obtenu les résultats suivants affichés sur la console, on les a résumés dans la figure 3.29 :

```
Les métriques de performance du système en utilisant SVM:
Accuracy: 0.92
Precision: 0.86
Recall: 0.86
F1 Score: 0.86
Confusion Matrix:
[[247  16]
 [ 16  98]]
```

Figure 3.29: Les métriques de performance du système en utilisant SVM

- **Accuracy :** Dans ce cas, l'accuracy est de 0.92, ce qui signifie que notre modèle prédit correctement 92% des échantillons.
- **Precision :** C'est la proportion d'échantillons positifs prédits correctement parmi tous les échantillons prédits comme positifs. Une précision de 0.86 signifie que lorsque le modèle prédit que l'échantillon est positif (dans ce cas, "Resa"), il est correct 86% du temps.
- **Recall :** C'est la proportion d'échantillons positifs prédits correctement parmi tous les échantillons réellement positifs. Un rappel de 0.86 indique que le modèle identifie correctement 86% de tous les échantillons réellement positifs.
- **F1 Score :** C'est une mesure combinée de la précision et du rappel, calculée comme une moyenne pondérée des deux. Il est particulièrement utile lorsque les classes sont déséquilibrées en taille. Un F1 score de 0.86 indique une bonne harmonie entre la précision et le rappel de notre modèle.
- **La matrice de confusion :**
  - Les valeurs sur la diagonale principale (de haut à gauche à bas à droite) représentent les prédictions correctes, dans ce cas 247 prédiction.
  - Hors de la diagonale, en haut à droite, on retrouve les faux positifs (des cas où le modèle a prédit "Resa" mais c'était incorrect), on en a 16.
  - En bas à gauche, on retrouve les faux négatifs (des cas où le modèle a prédit "Intrus" mais c'était incorrect), on en a 16 également.

- En bas à droite, ce sont les vrais négatifs (des cas où le modèle a prédit "Intrus" et c'était correct), on en a 98.

En conclusion, notre modèle SVM semble performant pour la tâche de reconnaissance de la classe "Resa" dans un contexte où les autres classes sont regroupées sous "Intrus". Il a une bonne capacité à identifier correctement les échantillons de la classe cible ("Resa"), tout en maintenant un faible taux de faux positifs.

3.7.6.2 Pour LDA

Nous avons calculé quelques métriques de performance par rapport au modèle LDA et avons obtenu les résultats suivants affichés sur la console, comme on peut le voir dans la figure 3.30 suivante :

```
Métriques de performance du système en utilisant LDA
Accuracy : 0.90
Precision : 0.88
Recall : 0.77
F-score : 0.82
Confusion Matrix :
[[251  12]
 [ 26  88]]
```

Figure 3.30: Les métriques de performance du système en utilisant LDA

- **Accuracy :** Dans ce cas, l'accuracy est de 0.90, ce qui signifie que notre modèle LDA prédit correctement 90% des échantillons.
- **Precision :** Une précision de 0.88 signifie que lorsque le modèle prédit "Resa", il a raison 88% du temps.
- **Recall :** Un rappel de 0.77 signifie que le modèle a correctement identifié 77% des instances de "Resa".
- **F1 Score :** Un F1 Score de 0.82 montre un bon équilibre entre la précision et le rappel.
- **La matrice de confusion :**
  - Les valeurs sur la diagonale principale (de haut à gauche à bas à droite) représentent les prédictions correctes, dans ce cas 251 prédiction.
  - Hors de la diagonale, en haut à droite, on retrouve les faux positifs (des cas où le modèle a prédit "Resa" mais c'était incorrect), on en a 12.
  - En bas à gauche, on retrouve les faux négatifs (des cas où le modèle a prédit "Intrus" mais c'était incorrect), on en a 26 également.

– En bas à droite, ce sont les vrais négatifs (des cas où le modèle a prédit "Intrus" et c'était correct), on en a 88.

En Conclusion, ces résultats montrent que le modèle LDA est assez précis et a un bon équilibre entre précision et rappel pour la tâche de reconnaissance de "Resa" par rapport aux intrus. Cependant, il y a encore des cas où "Resa" est confondue avec des intrus et vice-versa, comme l'indiquent les faux positifs et faux négatifs dans la matrice de confusion car le système n'est pas sûr à 100%.

3.7.7 La précision globale (Accuracy) des modèles par rapport aux dimensions de l'image

Les résultats que nous avons obtenus ci-dessus sont obtenus avec une taille d'images de 200x200 pixels, telle qu'elle a été définie par défaut dans le dataset. Ce nous voulons voir à présent, c'est la précision de nos modèles par rapport à la taille de l'image, est-ce que les modèles sont plus précis si l'image est plus petite ou plus grande.

Pour faire cette comparaison, nous avons refait les mêmes étapes que précédemment sauf que cette fois-ci, nous avons changé la taille des images utilisées et nous avons obtenu ces résultats illustrés dans la figure 3.31 suivante :

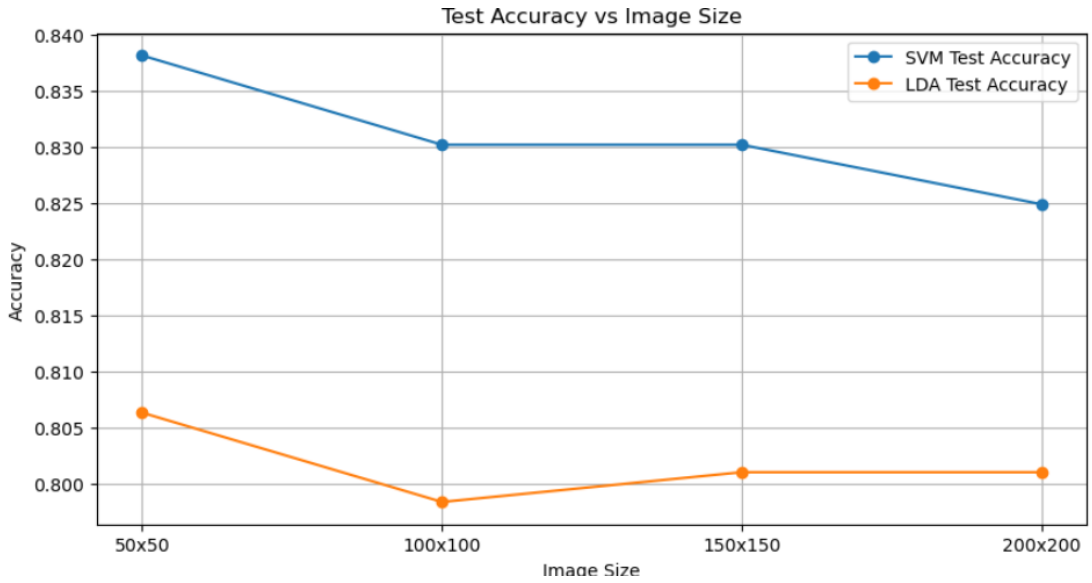


Figure 3.31: La précision globale des modèles par rapport aux dimensions des images

On remarque que pour toutes les tailles d'images, le modèle SVM obtient toujours une précision légèrement supérieure à celle du modèle LDA. Cela suggère que SVM est meilleur que LDA pour cette tâche de classification d'images dans ce contexte particulier. On peut également conclure que la précision des deux modèles ne varie pas de manière significative avec la taille des images, restant relativement stable autour de 82-84% pour SVM et 80-81% pour LDA.

Dans notre cas précis, il semble que le choix de la taille de l'image n'ait pas un impact majeur sur les performances des modèles, et que SVM soit une meilleure option que LDA pour cette tâche spécifique de classification d'images.

3.7.8 Résumé de toutes les étapes

Nous pouvons résumer tout ce que nous avons fait dans le schéma de la figure 3.32 suivante :

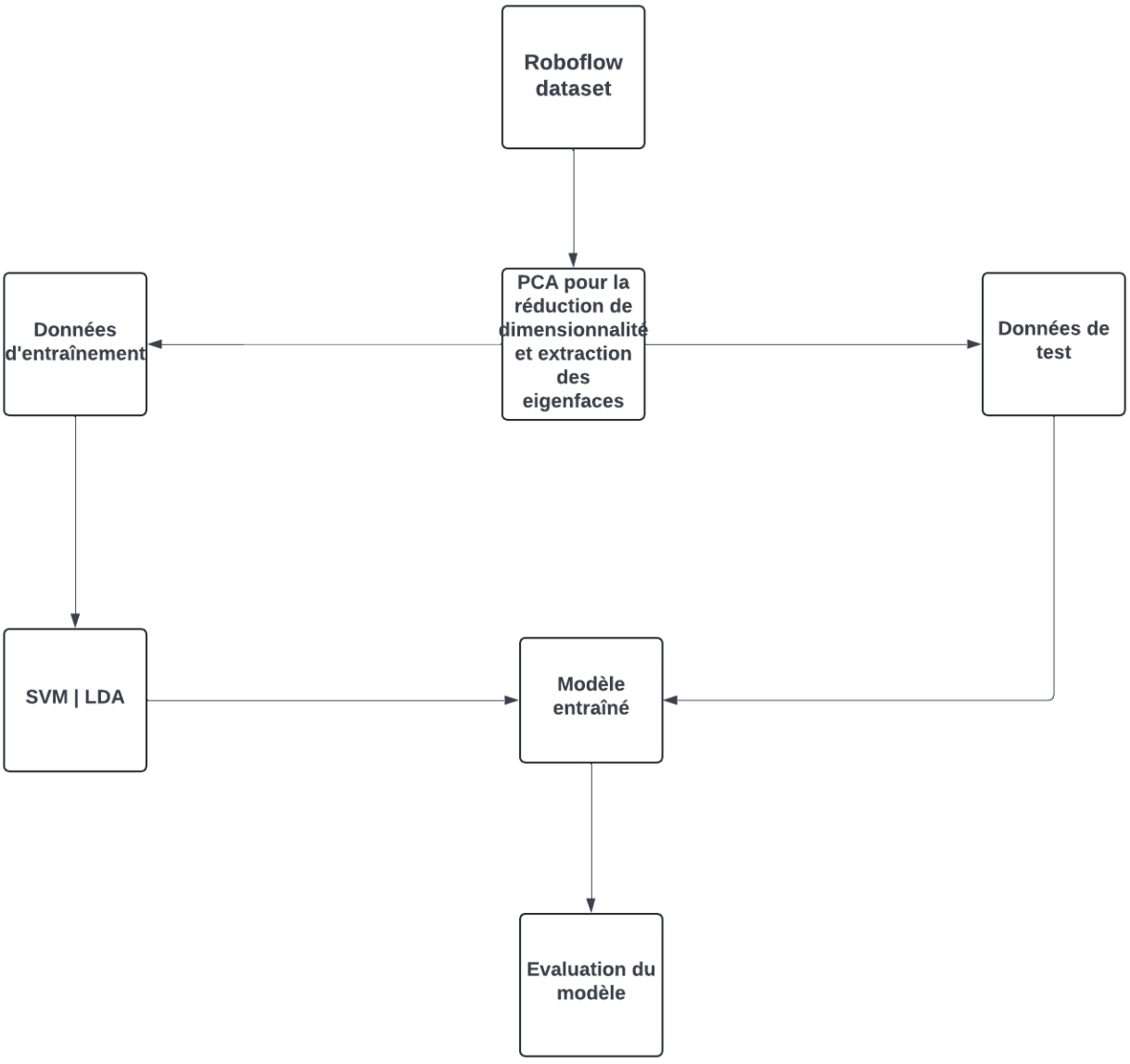


Figure 3.32: Schéma récapitulatif

3.8 Conclusion

Dans ce chapitre, nous avons étudié l'efficacité de deux méthodes de classification supervisée, le Support Vector Machine (SVM) et l'Analyse Discriminante Linéaire (LDA), appliquées à la reconnaissance faciale après une réduction de dimensionnalité par l'Analyse en Composantes Principales (PCA). Les expérimentations ont été menées sur un dataset qui contient des images de visages de trois filles distinctes : Resa, Jocelyn, et Ruby. Nous avons tout d'abord considéré que "Resa" est la seule habitante autorisée tandis que les deux autres sont considérées comme "intrus". Puis, nous avons mis en place un modèle de reconnaissance faciale utilisé pour détecter des intrusions dans une maison en fournissant l'ooprotunité d'envoyer des alertes en temps réel via des messages WhatsApp. Enfin, nous avons évalué l'impact du nombre de caractéristiques des images et la dimension de différentes tailles d'images sur les performances des deux modèles PCA + SVM ainsi que PCA + LDA.

Les résultats obtenus montrent que le SVM et le LDA offrent des niveaux de précision comparables, bien que le SVM ait surpassé le LDA de quelques points de pourcentage.

Cette analyse souligne l'importance du pré-traitement des données, en particulier de la réduction de la dimensionnalité avec PCA, pour améliorer l'efficacité des algorithmes de Machine Learning dans des applications comme la reconnaissance faciale. En outre, le choix du modèle de classification a aussi un impact significatif sur les performances finales, avec le SVM légèrement supérieur dans ce cas.

Conclusion générale et perspectives

Cette partie clôture notre mémoire en résumant notre contribution tout en donnant quelques directions concernant nos perspectives. Nous constatons aujourd'hui une expansion fulgurante de l'IoT dans notre vie quotidienne, nos objets deviennent de plus en plus connectés et nous devons donc nous adapter à cette évolution, adapter notre quotidien et essayer de tirer un avantage de cet essor. En effet, dans le contexte d'une maison intelligente, il nous est apparu logique d'utiliser cette interconnexion d'objets afin de garantir notre propre sécurité tel que nous l'avons fait en intégrant le Machine Learning dans un système de reconnaissance faciale afin de détecter un intrus dans une maison intelligente et de ce fait améliorer notre qualité de vie.

Le but de notre projet de fin d'études est de faire une comparaison des méthodes de classification supervisée à savoir l'algorithme SVM et l'algorithme LDA pour la reconnaissance faciale après avoir appliqué l'algorithme PCA pour la réduction de la dimensionnalité ainsi que le pré-traitement des données, dans notre cas, des images tirées d'un dataset.

Dans le premier chapitre, nous avons parlé des l'IoT en général ainsi que ses domaines d'application, nous avons également souligé l'importance de la sécurité au niveau de la couche application ainsi que la sécurité physique, c'est à dire la sécurité de la personne. Dans le second chapitre, nous avons abordé la vidéosurveillance en choisissant une maison intelligente comme contexte d'étude et avons fait un état de l'art des différentes techniques de détection d'intrusion puis dans le troisième chapitre, nous avons démontré l'efficacité des algorithmes de Machine Learning pour faire de la reconnaissance faciale. Les résultats sont plutôt satisfaisants pour les deux algorithmes de classification, bien qu'il y ait une légère supériorité pour SVM.

En résumé, l'utilisation de PCA combinée avec des modèles de classification tels que SVM et LDA peut fournir des solutions efficaces pour la reconnaissance faciale, avec une performance optimale obtenue pour des images qu'on a redimensionné afin qu'elles aient une plus petite dimension. Les résultats de ce chapitre fournissent des indications pour le développement de systèmes de reconnaissance faciale, en soulignant les avantages spécifiques des algorithmes SVM et LDA dans différents contextes de pré-traitement des données.

Notre mémoire met en lumière l'impact positif des techniques avancées d'analyse de données et de Machine Learning dans le domaine de la sécurité IoT, en particulier pour la reconnaissance faciale et la surveillance intelligente. Les résultats obtenus fournissent des bases solides pour le développement de systèmes robustes et efficaces, capables de répondre aux défis croissants de sécurité dans les environnements connectés.

Dans la continuité du travail que nous avons proposé, nous pourrions envisager les perspectives suivantes :

- **Amélioration des modèles proposés :** Poursuivre la recherche sur les techniques de classification et d'authentification pour optimiser davantage la précision et la fiabilité des systèmes de reconnaissance faciale dans des conditions variées.
- **Entraîner de nouveaux modèles :** Utiliser d'autres méthodes de Machine Learning afin de faire de la reconnaissance faciale.

Bibliography

[1] Albertschool. https://www.albertschool.com/blog/quelles-sont-les-bibliotheques-python-pour-le-machine-learning.

[2] Astera. https://www.astera.com/fr/type/blog/data-preprocessing/: :text=Le

[3] Cisco. https://www.cisco.com/c/fr_ca/products/wireless/what-is-wifi.html.

[4] Cnil. https://www.cnil.fr/fr/definition/reconnaissance-faciale#:~:text=La%20reconnaissance%20faciale%20est%20une,'un%20controled'accÃÍs).

[5] Cnil. https://www.cnil.fr/fr/definition/apprentissage-par-renforcement: :text=L'apprentissage

[6] Connectwave. https://www.connectwave.fr/techno-appli-iot/iot/reseaux-et-infrastructures-iot/.

[7] Data-bird. https://www.data-bird.co/blog/bibliotheque-python.

[8] Datascientest. https://datascientest.com/pandas-python-data-science.

[9] Datascientist. https://datascientest.com/pillow-tout-savoir.

[10] Futura. https://www.futura-sciences.com/tech/definitions/electronique-bluetooth-16527/.

[11] Futura-sciences. https://www.futura-sciences.com/tech/definitions/informatique-python-19349/.

[12] Ibm. https://www.ibm.com/fr-fr/topics/supervised-learning.

[13] Ibm. https://www.ibm.com/fr-fr/topics/unsupervised-learning.

[14] Ibm. https://www.ibm.com/fr-fr/topics/semi-supervised-learning.

[15] Intuit mailchimp. https://mailchimp.com/fr/resources/semi-supervised-learning/: :text=L'apprentissageConsulté le 27 mai 2024.

[16] Ip-systèmes. https://www.ip-systemes.com/quest-ce-que-lora-lorawan.html.

[17] Jedha bootcamp. https://www.jedha.co/formation-ia/matrice-confusion.

[18] Kaspersky. https://www.kaspersky.fr/resource-center/definitions/what-is-facial-recognition.

[19] Python | haar cascades for object detection. https://www.geeksforgeeks.org/python-haar-cascades-for-object-detection/.

[20] Safetyculture. `https://safetyculture.com/fr/themes/smart-farming/`.

[21] Sbedirect. `https://sbedirect.com/fr/blog/article/comprendre-la-rfid-en-10-points.html`.

[22] tkinter — python interface to tcl/tk. https://docs.python.org/fr/3/library/tkinter.html.

[23] Home Automation and Intruder Detection. Security system annotation dataset. `https://universe.roboflow.com/home-automation-and-intruder-detection/security-system-annotation`, apr 2023.

[24] Mario Ballano Barcena and Candid Wueest. Insecurity in the internet of things, 2015.

[25] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[26] Ali Bouchareb. Développement d'une architecture flexible pour la gestion et la sécurité des transactions et flux de données pour les objets connectés dans le contexte de la maison intelligen. Master's thesis, L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES, 2023.

[27] Dr. Savitha Choudhary, Ajith D R, H Likith Sai Varma, Lakshman Kumar S, and Lekhith R. Smart surveillance monitoring system using machine learning and raspberry pi. *International Research Journal of Modernization in Engineering Technology and Science*, 2022.

[28] Félix-Bazin Polla de Ndjampa. *Reconnaissance d'actions à l'aide d'un imageur infrarouge basse résolution*. PhD thesis, Université d'Orléans, France, 2020.

[29] Bruno Dorsemaine, Jean-Philippe Gaulier, Jean-Philippe Wary, Nizar Kheir, and Pascal Urien. Internet of things : A definition taxonomy. In *In Next Generation Mobile Applications, Services and Technologies*, pages 72–77, 2015.

[30] Maheen Zulfiqar et al. Deep face recognition for biometric authentication. In *1st International Conference on Electrical, Communication and Computer Engineering (ICECCE)*, Swat, Pakistan, 2019.

[31] Sébastien Faux. La sécurité à l'ère des objets connectés : Comment s'y prendre ? *Workshop "Sécurité Des Objets Connectés",*, 2017.

[32] Tanguy Godquin. *Sécurisation adaptative des objets de l'IoT par méthodes logicielles (White box) et combinées (hardware et software)*. PhD thesis, Normandie Université, 2020.

[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[34] Jérôme Gorin. . objets connect´es : Typologie des risques sur la vie privée. *Workshop "Sécurité Des Objets Connectés*, 2017.

[35] Ms Jyoti, J Jadhav, and J Jyoti. Moving object detection and tracking for video surveliance. *International journal of engineering research and general science*, 2014.

[36] Vijayalakshmi Kakulapati, Sridhar Potla, Mayukha Sista, and Ashish Kumar Chelukali. An novel approach object detection of video surveillance system using opencv. *ResearchGate*, 2024.

[37] G. R. Karpagam, A. Mukeshpandi, K. S. Sanjay Krishnan, and S. Vanjinathan. Ai powered partially parallelized scheme for multifactor authentication.

[38] Filipe Moura and João de Abreu e Silva. Smart cities: Definitions,evolution of the concept and examples of initiatives. *Springer Nature Switzerland AG*, 2019.

[39] MERAMRIA Nabila. Reconnaissance de visages par analyse discriminante linéaire(lda). Master's thesis, UNIVERSITE BADJI MOKHTAR ANNABA, 2016.

[40] Thi Quynh Nguyen. *Apprentissage automatique non supervisé pour la détection de trafics illégitimes*. PhD thesis, Université Paul Sabatier - Toulouse III, France, 2023.

[41] Zhanserik Nurlan, Batyrzhan K. Akhmetzhanov, Omar Aslan Gazizuly, and Nurkhat Zhakiyev. Integration of a video surveillance system into a smart home using the home assistant platform. In *2022 International Conference on Smart Information Systems and Technologies (SIST)*, Turquie, 2022.

[42] Mrs.Donepudi Priyanka, Peddagamalla Kavitha, Naragam Vineela, Muthireddy Naga Sai Deepika, and Molleti Aanoor Venkata Mahesh. Iot powered smart doorbell system for enhanced home security and communication. *Journal of Management Entrepreneurship*, 2024.

[43] Balaji S R. A survey on moving object tracking using image processing. In *2017 11 th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, Tamilnadu, India, 2017.

[44] MAHAMAT CHARFADINE SALIM. *Gestion dynamique et évolutive de règles de sécurité pour l'Internet des Objets*. PhD thesis, UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE, Reims, France, 2019.

[45] Bruce Schneier. Lessons from the dyn ddos attack, 2016.

[46] Patrick Sebastian, Yap Vooi Voon, , and Richard Comley. Colour space effect on tracking in video surveillance. *International Journal on Electrical Engineering and Informatics*, 2010.

[47] Tidiane Sylla. *Sécurité et vie privée centrées sur l'utilisateur dans l'IoT*. PhD thesis, Université des sciences, des techniques et des technologies.

[48] Tidiane Sylla, Mohamed-Aymen Chaloufi, and Francine Krief. *Adaptation du niveau de sécurité des applications IoT*. PhD thesis, Université des sciences, des techniques et des technologies.

[49] Akula Surya Teja, Ginni Chandra Mohini, Dannana Dhanunjay, and Dr P M Manohar. Realtime intrusion detection system using open cv. *Journal of Survey in Fisheries Sciences*, 2023.

[50] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking : A survey. *Acm computing surveys (CSUR)*, 2006.

[51] Zhi-Kai Zhang, Michael Cheng Yi Cho, Chia-Wei Wang, Chia-Wei Hsu, ChongKuan Chen, and Shiuhpyng Shieh. Iot security : Ongoing challenges and research opportunities. In *International Conference on ServiceOriented Computing and Applications*, page 230–234, 2014.

Résumé

Dans ce mémoire de fin d'étude, nous avons proposé un modéle de détection d'intrusion dans le cas d'une maison intelligente. Le modéle proposé utilise un système de reconnaissance faciale au moyen de vidéo de surveillance tout en se basant sur des techniques d'Intelligence Artificielle et des algorithmes de Machine Learning, à savoir : PCA, SVM et LDA. Nous avons ensuite implémenté et évalué la proposition en utilisant le langage de programmation Python.
L'évaluation de performance du modéle proposé : PCA pour le pré-traitement des données, SVM pour la classification ainsi que LDA, a été réalisée en calculant plusieurs métriques et en variant le nombre de caractéristiques PCA ainsi que la dimension des images, tout en sachant que les images du Dataset ont été capturées depuis des vidéos. L'analyse des résultats montrent que l'approche PCA combinée SVM (PCA+SVM) est légèrement meilleure que PCA combinée LDA (PCA+LDA) bien que les deux soient satisfaisantes.

Mots clés : IoT, Maison Intelligente, Détection d'Intrusion, Vidéo Surveillance, Reconnaissance Faciale, Apprentissage Automatique.

Abstract

In this thesis, we proposed an intrusion detection model in the case of a smart home. The proposed model uses a facial recognition system with video surveillance based on Artificial Intelligence techniques and Machine Learning algorithms, namely: PCA, SVM and LDA. We then implemented and evaluated the proposal using the Python programming language.
The performance evaluation of the proposed model: PCA for data pre-processing, SVM for classification and LDA, was performed by calculating several metrics and varying the number of PCA characteristics as well as the image dimension, knowing that the Dataset images have been captured from the videos. The analysis of the results shows that the combined PCA SVM (PCA+SVM) approach is slightly better than the combined PCA LDA (PCA+LDA) although both are satisfactory.

Keywords : IoT, Smart Home, Intrusion Detection, Video Surveillance, Facial Recognition, Machine Learning.