

**UNIVERSITE INTERNATIONALE DE TUNIS
ÉCOLE SUPÉRIEURE INTERNATIONALE PRIVÉE DE TUNIS**

Projet de fin d'études pour l'obtention de la
Licence en Génie Logiciel & Développement des
Systèmes d'information

Présenté et soutenu publiquement par

Adam LASSOUED

**DÉTECTION EN TEMPS RÉEL DANS LES
VIDÉOS EN STREAMING UTILISANT UNE
APPROCHE DE DEEP LEARNING**

Juin 2024

Encadrant académique : Mme Leila BEN AYED

Encadrant professionnel : Mr Sofiane CHAABA

Attestation de stage

Appréciations, signature et cachet des encadrants

Je suis ravi de vous annoncer que M. Adam Lassoued a terminé avec succès son stage au sein de l'entreprise Opalia Pharma. Pendant ces 4 mois, il a travaillé sur le projet de détection en temps réel dans les vidéos en streaming grâce à une approche de deep learning. M. Lassoued s'est rapidement intégré à l'équipe et a accompli ses missions avec brio, démontrant une constante volonté de se surpasser. Adam a fait preuve d'une grande capacité d'adaptation, de rigueur, d'autonomie et d'efficacité dans toutes ses tâches. Je reste à votre disposition pour toute information complémentaire et vous prie d'agréer, Madame, Monsieur, l'expression de mes salutations distinguées.

Fait à Tunis le 20/06/2024

Sofien CHAABA
IT Manager
OPALIA PHARMA SA.
Chaaba.s@opaliarecordati.com
Tel. 00216 58381917

J'ai suivi Monsieur Adam Lassoued dans son projet de fin d'études portant sur la détection en temps réel dans les vidéos en streaming utilisant une approche de deep learning. Le travail élaboré est consistant sur les plans méthodologique et technologique. Il a adopté l'approche SCRUM dans la gestion des composants qu'il a mis en place. Il fait preuve de créativité avec un esprit collaboratif qui ne fait qu'augmenter la qualité du produit élaboré et des modules implémentés. Je le félicite pour tout l'effort fourni, pour la qualité de son travail et la rigueur qu'il prouve. Je lui souhaite un excellent avenir et je suis confiante de sa réussite.



Fait à Tunis le 20/06/2024

Leila BEN AYED
Professeur
Ecole Nationale des sciences de
l'Informatique -TUNISIE
Laboratoire LIPSIC FST
Leila.benayed2023@gmail.com
Tel. 00216 52313247

Dédicaces

En guise de reconnaissance et de gratitude, je souhaite dédier ce travail à toutes les personnes qui m'ont soutenu et encouragé tout au long du parcours.

À mes **chers parents**, Aucune dédicace et aucun mot ne pourrait exprimer mon respect, mon estime et l'amour éternel pour tous vos sacrifices et vos efforts. J'espère que j'ai été à la hauteur de vos attentes et que vous trouverez ce travail le fruit de votre dévouement et l'expression de mon profond amour.

À ma **sœur Sabine**, Je te remercie pour ton soutien et de tes précieux conseils et j'espère qu'on sera toujours unis.

À tous **mes proches et mes amis**, Je vous remercie pour vos encouragements et pour l'espoir que vous avez fondé en moi.

Remerciements

Au terme de mon projet de fin d'étude, je souhaite rendre hommage à tous ceux qui ont contribué à la réalisation de ce travail.

J'adresse mes remerciements à **Madame Leila Ben Ayed**, mon encadrante académique qui n'a jamais cessé de m'aider, pour l'effort remarquable et son encadrement rigoureux.

Je tiens aussi à exprimer ma gratitude à **Monsieur Sofiane Chaaba**, mon encadrant professionnel, pour sa confiance et pour ses conseils avisés dans le domaine professionnel.

Je tiens également à remercier le jury pour l'honneur qu'il m'a fait en acceptant d'évaluer mon travail.

Finalement, je tiens à remercier tous mes enseignants à qui je dois ma formation et qui n'ont jamais cessé de nous assister.

Sommaire

Introduction générale	1
Chapitre 1	3
Cadre général du projet	3
1 Introduction	3
2 Présentation de l'organisme d'accueil.....	3
3 Présentation du projet.....	8
4 Conclusion.....	13
Chapitre 2	15
Analyse et spécification des besoins	15
1 Introduction	15
2 Identification des acteurs.....	15
3 Spécifications des besoins	15
4 Pilotage du projet avec SCRUM	18
5 Conclusion.....	21
Chapitre 3	22
Conception	22
1 Introduction	22
2 Architecture	22
3 Conception détaillée	23
4 Conclusion.....	29
Chapitre 4	30
Étude des modèles de l'intelligence artificielle adoptés	30
1 Introduction	30
2 Modèle de détection de visages.....	30
3 Modèle de reconnaissance faciale	32
4 Conclusion.....	36
Chapitre 5	37
Réalisation.....	37
1 Introduction	37
2 Environnement de travail	37
3 Mise en œuvre	40
5 Conclusion.....	51

Conclusion générale	52
Références	54

Liste des figures

Figure 1 : Logo Opalia Pharma	3
Figure 2 : L'urbanisme de l'entreprise.....	5
Figure 3 : Exemples de caméras Dahua	6
Figure 4 : Systèmes NVR et DVR	6
Figure 5 : Organigramme de l'entreprise Opalia Pharma	7
Figure 6 : Schéma du fonctionnement global de la solution proposée.....	9
Figure 7 : logo do Litum	10
Figure 8 : Suivi des déplacements des employés via le RTLS de Litum	10
Figure 9 : Interface de visualisation de Smart PSS	13
Figure 10 : Diagramme de cas d'utilisation global	17
Figure 11 : Cycle d'un projet piloté avec SCRUM.....	19
Figure 12 : Architecture de l'application	22
Figure 13 : Diagramme de séquence du sprint 1	24
Figure 14 : Diagramme de séquence du sprint 2.....	25
Figure 15 : Diagramme de séquence du sprint 3.....	26
Figure 16 : Diagramme d'activités du système.....	27
Figure 17 : Diagramme de séquence du sprint 4.....	29
Figure 18 : Détection de visages utilisant Haar Cascade	31
Figure 19 : Calcul de similarité dans un réseau de neurones siamois	33
Figure 20 : Architecture convolutive sélectionnée pour la tâche de vérification	34
Figure 21 : logo Visual Studio Code.....	37
Figure 22 : logo draw.io	38
Figure 23 : logo Python.....	38
Figure 24 : logo OpenCV.....	38
Figure 25 : logo TensorFlow.....	39
Figure 26 : logo Keras.....	39
Figure 27 : logo Tkinter	39
Figure 28 : logo MySQL Workbench	40
Figure 29 : Détection et extractions de visages à partir d'images vidéo	41
Figure 30 : Extrait d'images faciales de la base de données LFW.....	42
Figure 31 : Résultat de la fonction de prétraitement des images.....	42
Figure 32 : Interprétation de la couche « embedding »	43
Figure 33 : Interprétation du réseau siamois	44
Figure 34 : Processus d'entraînement du modèle siamois	45
Figure 35 : Exécution de la fonction de vérification	46
Figure 36 : Aperçu des tables de la base de données	47
Figure 37 : Opérations d'archivage des données.....	48
Figure 38 : Fenêtre d'authentification.....	49
Figure 39 : Vue globale de l'interface de visualisation des données	49
Figure 40 : Exemple 1 d'un employé sélectionné.....	50
Figure 41 : Exemple 2 d'un employé sélectionné.....	50
Figure 42 : Fenêtre pop-up d'un historique personnel	51

Liste des tableaux

Tableau 1 : Identification des acteurs..... 15

Tableau 2 : Répartition des rôles dans SCRUM..... 18

Tableau 3 : Programmation des sprints 19

Tableau 4 : Product Backlog 20

Tableau 5 : Backlog du sprint 1 24

Tableau 6 : Backlog du sprint 2 24

Tableau 7 : Backlog du sprint 3 25

Tableau 8 : Backlog du sprint 4 28

Tableau 9 : Avantages et limites du modèle Haar Cascade..... 32

Liste des abréviations

ZAC : Zone d'Aménagement Concerté

IP : Internet Protocol

CAT5e : Category 5 enhanced

CAT6 : Category 6

PoE : Power over Ethernet

DVR : Digital Video Recorder

NVR : Network Video Recorder

RTLS : Real-Time Location System

RFID : Radio Frequency Identification

CNN : Convolutional Neural Network

GPU : Graphics Processing Unit

GUI : Graphical User Interface

SQL : Structured Query Language

LFW : Labeled Faces in the Wild

Introduction générale

Ce rapport présente le travail réalisé dans le cadre d'un projet visant à améliorer le système de sécurité de l'entreprise Opalia Pharma et à répondre à son besoin croissant en matière de surveillance intelligente en temps réel.

Ce projet est entrepris dans le but d'obtenir le Diplôme de Licence Fondamentale en Génie Logiciel et Développement des Systèmes d'Information à l'Université Internationale de Tunis.

La surveillance des flux vidéo provenant des différentes caméras installées dans les secteurs administratifs et opérationnels est assurée par une application de surveillance qui est en service depuis plusieurs années. Toutefois, il est à ce jour essentiel de surveiller en temps réel les images vidéo ou d'analyser les enregistrements ultérieurs pour détecter les intrusions d'employés dans des zones non autorisées à eux.

L'objectif principal de cette étude est de concevoir et mettre en œuvre un système autonome capable de détecter toute infraction de manière proactive. Plus spécifiquement, les objectifs de ce projet sont les suivants :

- Acquisition en temps réel des flux vidéo et détection des visages, suivie de leur reconnaissance.
- Vérification de l'autorisation de présence de l'employé identifié sur la zone de la caméra source.
- Enregistrement en temps réel d'images identifiées, horodatées et localisées de toute infraction détectée.
- Compilation de données statistiques individuelles, illustrant le nombre d'apparitions dans des zones non autorisées pour chaque employé.

Pour une conduite efficace de ce projet, nous avons opté pour une démarche fondée sur la méthodologie SCRUM. Cette méthodologie a facilité la gestion grâce à sa structure itérative et collaborative, nous permettant de planifier et de prioriser les tâches de manière claire. SCRUM s'est également avéré adaptable aux changements imprévus, permettant des ajustements rapides.

Nous avons structuré le travail en utilisant des sprints, en suivant les principes de cette méthodologie. Des réunions périodiques, telles que les daily stand-ups et les rétrospectives, étaient tenues pour évaluer la progression du projet, définir les priorités pour les prochains sprints et résoudre les éventuels obstacles rencontrés.

Ce rapport expose le travail accompli de la manière suivante :

- Dans le premier chapitre, nous introduisons le cadre global du projet en présentant l'organisme d'accueil, tout en mettant en lumière l'état de l'art, les objectifs visés et la solution proposée.
- Le deuxième chapitre se focalise sur l'analyse et la spécification des besoins, en détaillant les fonctionnalités attendues.
- Le troisième chapitre présente l'architecture du projet, ainsi que la conception détaillée de chaque sprint.
- Le quatrième chapitre s'intéresse à l'étude des deux modèles de l'intelligence artificielle adoptés pour réaliser les tâches de détection de visages et de reconnaissance faciale au sein du système.
- Le cinquième chapitre décrit en détail les méthodes et les ressources utilisées pour réaliser le projet. Nous y expliquons les différentes étapes du processus de développement, les outils et les technologies employées pour produire les différentes composantes de la solution.

En guise de conclusion, ce rapport récapitule la réalisation du projet et met en évidence l'importance du système de contrôle intelligent qui a été élaboré. Il souligne également les implications et les avantages potentiels de ce système dans le contexte de la surveillance et de la sécurité en temps réel.

Chapitre 1

Cadre général du projet

1 Introduction

L'objectif de ce chapitre est de déterminer le contexte du projet de fin d'études. Nous débuterons en présentant l'organisme d'accueil et les prestations qu'il propose. Nous allons ensuite présenter le projet en étudiant sa problématique et ses objectifs. Subséquemment, nous exposerons le travail demandé et la solution que nous proposons. Enfin, nous allons révéler une solution similaire et critiquer l'opération classique au sein de l'entreprise.

2 Présentation de l'organisme d'accueil

Au cours de cette partie, nous présentons l'entreprise au sein de laquelle notre projet a été réalisé, à savoir Opalia Pharma. [1]



Figure 1 : Logo Opalia Pharma

2.1 Présentation générale

Opalia Pharma a été créée en 1988 par le Dr Alya El HEDDA avec une vision claire ayant pour objectif de fournir des solutions pharmaceutiques de qualité à un coût abordable. Cette entreprise développe, fabrique et commercialise une large gamme de médicaments génériques, compléments alimentaires, dispositifs médicaux et produits pharmaceutiques à usage humain sous différentes formes pharmaceutique.

Opalia Pharma a obtenu la certification de son système qualité, démontrant son engagement envers l'excellence et la conformité aux normes internationales.

2.2 Secteurs d'activité

Opalia Pharma dispose de trois laboratoires distincts qui revêtent une importance cruciale dans les domaines de la recherche, du développement, de la fabrication et du contrôle qualité des médicaments :

- Le laboratoire de recherche et développement : ce laboratoire est chargé du développement de nouveaux médicaments. Il se concentre sur l'identification de nouvelles molécules, la conception de formulations pharmaceutiques et l'établissement de procédés de fabrication.
- Le laboratoire de contrôle qualité physicochimique : ce laboratoire se spécialise dans l'évaluation de la qualité des médicaments sur le plan physicochimique. Il effectue des tests pour évaluer la pureté, l'identité, la stabilité et d'autres caractéristiques physicochimiques des produits pharmaceutiques. Son objectif principal est de garantir que les médicaments respectent les normes de qualité et de conformité réglementaires.
- Le laboratoire de microbiologie : ce laboratoire est responsable du contrôle qualité microbiologique des médicaments fabriqués par Opalia Pharma. Il réalise des tests pour détecter la présence de contaminants microbiens tels que les bactéries, les champignons ou les virus, afin de garantir la sécurité des médicaments et d'éviter tout risque d'infection pour les patients.

2.3 Infrastructure de l'entreprise et de l'installation industrielle

Dans cette partie, nous présenterons l'infrastructure physique de l'entreprise, y compris les zones administratives et les zones opérationnelles de l'usine. De plus, comme le projet est axé sur la sécurité, nous décrirons également l'infrastructure de la sécurité mise en place.

2.3.1 Infrastructure physique

La structure occupe une surface de 20000 m² subdivisée comme suit :

- 4000 m² de zone à atmosphère contrôlée (ZAC)
- 4000 m² d'étage technique (traitement d'eau et l'air)
- 5000 m² de magasins (capacité de stockage de 3800 palettes)
- 4000 m² d'administration

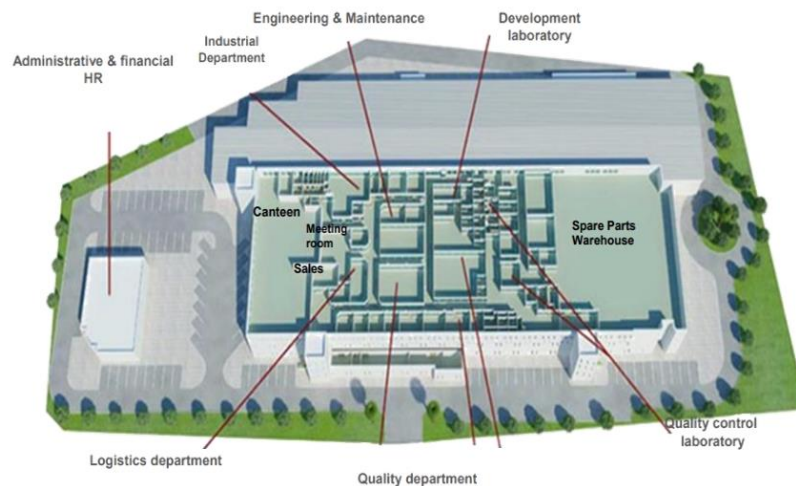


Figure 2 : L'urbanisme de l'entreprise

2.3.2 Infrastructure de la sécurité

L'infrastructure de la sécurité mise en place repose sur un système de vidéosurveillance intégrant un total de 136 caméras de la marque Dahua. Elles varient en résolution, allant de 2 mégapixels à 12 mégapixels, assurant ainsi une qualité d'image adaptée à différents besoins de surveillance.

Ces caméras se divisent en deux catégories principales :

- **Les caméras IP (Internet Protocol) :** Elles sont connectées via un réseau Ethernet, offrent une qualité d'image supérieure et une grande flexibilité en termes de déploiement grâce à leur capacité de transmission de données sur de longues distances sans perte de qualité. Elles utilisent des câbles réseau (CAT5e ou CAT6) pour l'alimentation électrique et la transmission de données, souvent via la technologie Power over Ethernet (PoE).
- **Les caméras coaxiales :** Plus traditionnelles, elles se connectent à l'aide de câbles coaxiaux et sont généralement alimentées séparément. Ces caméras sont souvent utilisées dans des installations existantes où le câblage est contraignant ou coûteux. Elles sont robustes et fiables pour des distances plus courtes sans nécessiter de compression de signal, ce qui les rend idéales pour certains contextes spécifiques.

Par ailleurs, des caméras 360 degrés ont été installées, notamment au niveau du magasin, offrant une couverture panoramique sans angles morts, ce qui est crucial pour surveiller une zone aussi large et ouverte. Ces caméras 360 degrés complètent le système en fournissant une vue d'ensemble intégrale, renforçant ainsi la sécurité globale de l'installation.



Figure 3 : Exemples de caméras Dahua

Pour la gestion et l'enregistrement des flux vidéo, l'entreprise dispose de trois NVR (Network Video Recorder) et trois DVR (Digital Video Recorder) [2] :

- **Les NVR** : dédiés aux caméras IP et fonctionnent en réseau, permettant une centralisation des données avec un accès à distance et des fonctionnalités avancées de traitement et d'analyse vidéo
- **Les DVR** : utilisés avec les caméras coaxiales, enregistrant les flux vidéo sur des disques durs locaux via une connexion directe



Figure 4 : Systèmes NVR et DVR

Cette distinction entre MVR et DVR garantit une flexibilité maximale et une optimisation de la sécurité en fonction des spécificités des différents types de caméras déployées sur le site.

2.4 Organigramme

Pour couronner la présentation de l'entreprise, il est crucial de présenter l'organigramme car il offre une vue claire de la structure organisationnelle, facilitant ainsi la compréhension des rôles et des responsabilités de chacun.

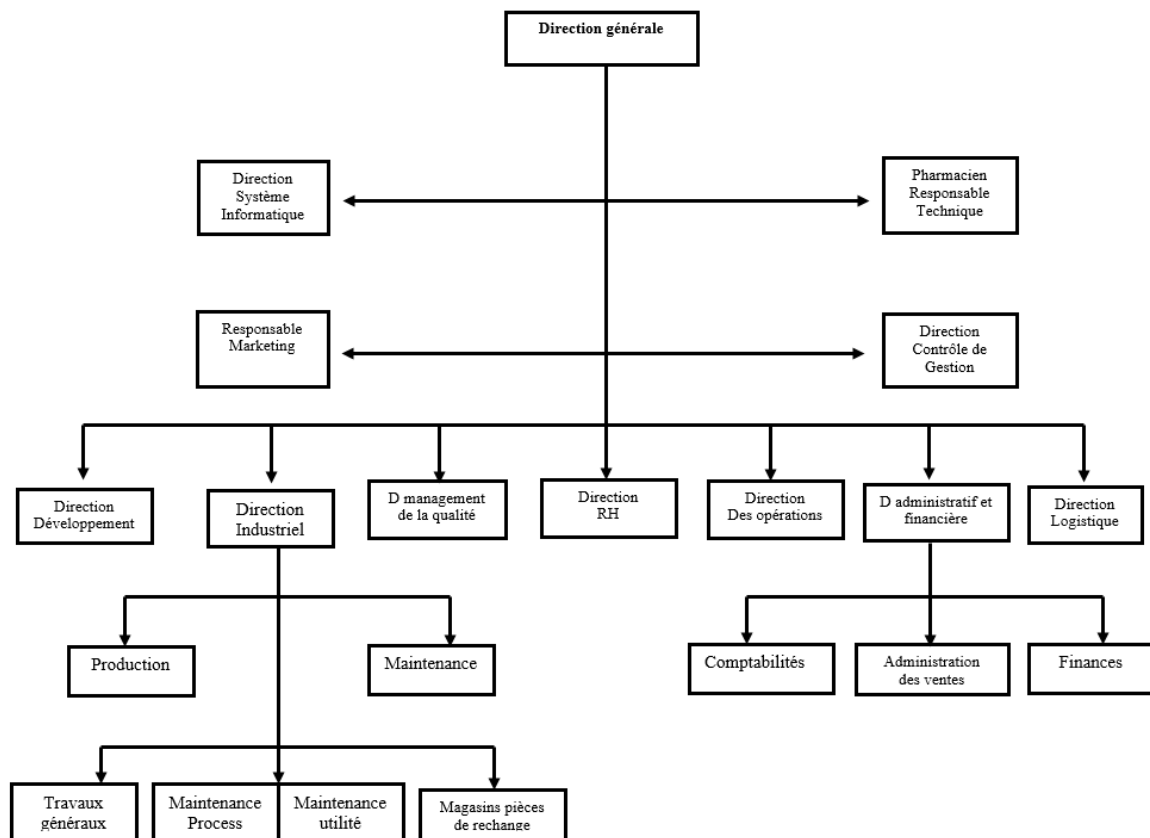


Figure 5 : Organigramme de l'entreprise Opalia Pharma

Cet organigramme permet d'identifier les lignes de communication et de hiérarchie, ce qui aide à améliorer l'efficacité et la coordination entre les équipes. De plus, il sert de guide pour les nouveaux employés, leur permettant de se familiariser rapidement avec la structure interne de l'entreprise.

3 Présentation du projet

Ce projet a pour ambition de concevoir un système de surveillance intelligent destiné à superviser de manière efficace la présence et les déplacements des employés au sein de l'entreprise, répondant ainsi aux exigences critiques de l'industrie pharmaceutique, tout en exploitant le réseau de caméras déployées à travers les différents secteurs.

3.1 Problématique et objectifs du projet

L'entreprise est confrontée à un obstacle majeur en raison de l'absence d'un système de suivi et de contrôle des déplacements des employés au sein de sa structure. Cette carence affecte la sécurité, la productivité et la gestion des ressources humaines, rendant difficile l'optimisation des opérations et la prévention des risques.

L'objectif est donc de développer une solution efficace pour détecter les individus présents dans les flux de vidéos en direct capturés par les caméras de surveillance, et de les identifier tout en vérifiant leur autorisation à accéder à des zones spécifiques. Cette solution permettra au responsable de la sécurité d'être informé, via un historique clair, lorsqu'un employé non autorisé dans une zone déterminée est détecté, lui permettant ainsi de prendre les mesures nécessaires.

3.2 Travail demandé

Afin d'atteindre les objectifs décrits précédemment, il m'a été demandé de me concentrer sur les aspects suivants :

- Effectuer des recherches approfondies pour identifier la technologie de détection d'objets la plus appropriée afin de détecter les visages en prenant en compte les angles des caméras installées et les différentes conditions de lumières.
- Conduire des investigations approfondies pour sélectionner le modèle de reconnaissance faciale optimal et spécifier la nature des données nécessaires à obtenir de l'entreprise pour entraîner ce modèle.
- Mettre en place un système de gestion des autorisations d'accès des employés, en fonction des différentes zones surveillées chacune par un certain nombre de caméras.
- Développer une application simple et intuitive permettant de visualiser de manière intégrée les données relatives à toutes les infractions enregistrées.

Ces points clés sont essentiels pour assurer la mise en œuvre réussie du projet, en garantissant particulièrement la qualité des technologies utilisées et globalement un fonctionnement optimal de la solution proposée.

3.3 Solution proposée

Une fois que les objectifs du projet et le travail demandé ont été clairement définis, il est maintenant approprié de contribuer en proposant une solution répondant à la problématique identifiée.

Dans ce contexte, nous avons conçu une solution composée de deux principaux éléments interconnectés via une base de données.

D'une part, la composante système fonctionne en temps réel et intègre les modèles de détection de visages et de reconnaissance faciale. Elle assure la vérification des accès pour chaque personne identifiée sur les images vidéo, archivant les infractions détectées dans une base de données. Les données sont enregistrées sous forme de chiffres dans des tables MySQL, d'images étiquetées, ainsi que de détails textuels dans un fichier journal incluant l'identité, l'horodatage et la localisation des infractions.

D'autre part, l'application de bureau analyse ces données et offre une vue d'ensemble des infractions détectées, tout en permettant un accès détaillé à l'historique personnel de chaque employé.

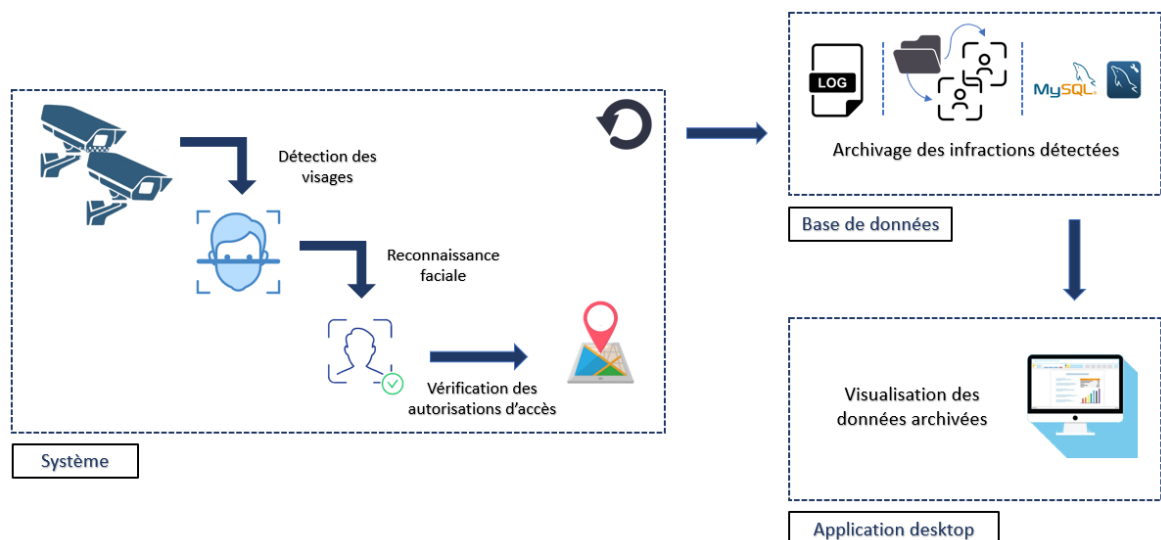


Figure 6 : Schéma du fonctionnement global de la solution proposée

3.4 Solution similaire



Figure 7 : logo do Litum

Un système qui vise à atteindre des objectifs similaires à ceux que nous poursuivons est le système Litum RTLS (Real-Time Location System) qui représente une solution innovante. Grâce à sa technologie avancée de localisation en temps réel, Litum RTLS permet de suivre et de gérer avec précision les actifs, le personnel et les processus au sein d'un environnement donné. Ce système améliore non seulement l'efficacité opérationnelle, mais aussi la sécurité et la gestion des ressources, offrant ainsi une réponse complète et efficace aux défis modernes de suivi et de gestion. [4]



Figure 8 : Suivi des déplacements des employés via le RTLS de Litum

Parmi les fonctionnalités clé de RTLS [5], nous pouvons citer :

- **Suivi en temps réel** : Litum utilise des étiquettes RFID actives qui émettent des signaux en continu vers des récepteurs installés dans l'installation. Ces récepteurs collectent les données de localisation en temps réel, permettant de savoir exactement où se trouvent les employés à tout moment.
- **Géofencing** : La technologie permet de définir des zones géographiques virtuelles (géofences) au sein de l'installation. Lorsqu'un employé entre ou sort de ces zones, des alertes peuvent être déclenchées. Ceci est particulièrement utile pour sécuriser les zones sensibles et restreindre l'accès non autorisé.
- **Alertes et notifications** : Le système peut être configuré pour envoyer des notifications instantanées en cas de violation des règles de sécurité, comme l'entrée dans une zone restreinte sans autorisation. Cela permet une réaction rapide pour éviter les incidents.
- **Analyse des données** : Litum RTLS enregistre toutes les données de mouvement, permettant une analyse approfondie pour améliorer les processus de sécurité et d'efficacité. En cas d'incident, les employeurs peuvent retracer les mouvements des employés pour comprendre ce qui s'est passé et éviter des problèmes similaires à l'avenir.
- **Sécurité et secours** : En situation d'urgence, le système peut localiser rapidement les employés, facilitant ainsi les opérations de secours. Les badges RFID peuvent être équipés de boutons de panique pour permettre aux employés de demander de l'aide immédiatement en cas de détresse.
- **Intégration avec les systèmes de sécurité** : Litum RTLS peut être intégré avec d'autres systèmes de sécurité, comme les systèmes de contrôle d'accès et les systèmes de gestion des alarmes, pour fournir une solution de sécurité complète et cohérente.

Quelques exemples d'utilisation de la solution de Litum incluent les cas suivants :

- **Industries manufacturières** : Suivi des opérateurs de machines pour s'assurer qu'ils ont les compétences requises et prévenir les accidents.
- **Hôpitaux** : Surveillance des mouvements des patients et du personnel pour optimiser les soins et les réponses aux urgences.
- **Mines et chantiers de construction** : Localisation des travailleurs pour les évacuations rapides et la prévention des accidents.

Le déploiement de la solution RTLS offre plusieurs avantages, parmi lesquels nous pouvons citer :

- **Réduction des coûts** : En réduisant le besoin de sécurité physique et en améliorant l'efficacité des opérations.
- **Amélioration de la sécurité** : En fournissant une visibilité en temps réel et des alertes instantanées, le système aide à prévenir les incidents et à réagir rapidement en cas de problème.
- **Conformité réglementaire** : Enregistrement précis des mouvements et des heures de travail pour se conformer aux régulations de travail et de sécurité (Litum RTLS).

En résumé, Litum RTLS est une solution robuste et adaptable pour le suivi en temps réel des employés et des actifs, offrant des fonctionnalités avancées pour améliorer la sécurité, l'efficacité et la conformité réglementaire dans divers environnements industriels.

3.5 Etude et critique de l'opération classique

Au sein d'Opalia Pharma, la solution actuelle est déployée principalement pour la visualisation des flux vidéo provenant des caméras de surveillance.

Elle se présente sous la forme de l'application Smart PSS [3], développée par Dahua, la même entreprise produisant les caméras déployées localement. Elle permet la navigation entre les différentes zones, chacune étant associée à un ensemble spécifique de vues de caméras.

Smart PSS est décrite par ses développeurs comme « une application de vidéosurveillance complète et tout-en-un, idéale pour les petites et moyennes entreprises qui ont besoin de surveiller les personnes, les locaux et les biens, intégrant toutes les caméras réseau et tous les périphériques de stockage Dahua dans une interface intuitive et facile à utiliser ».

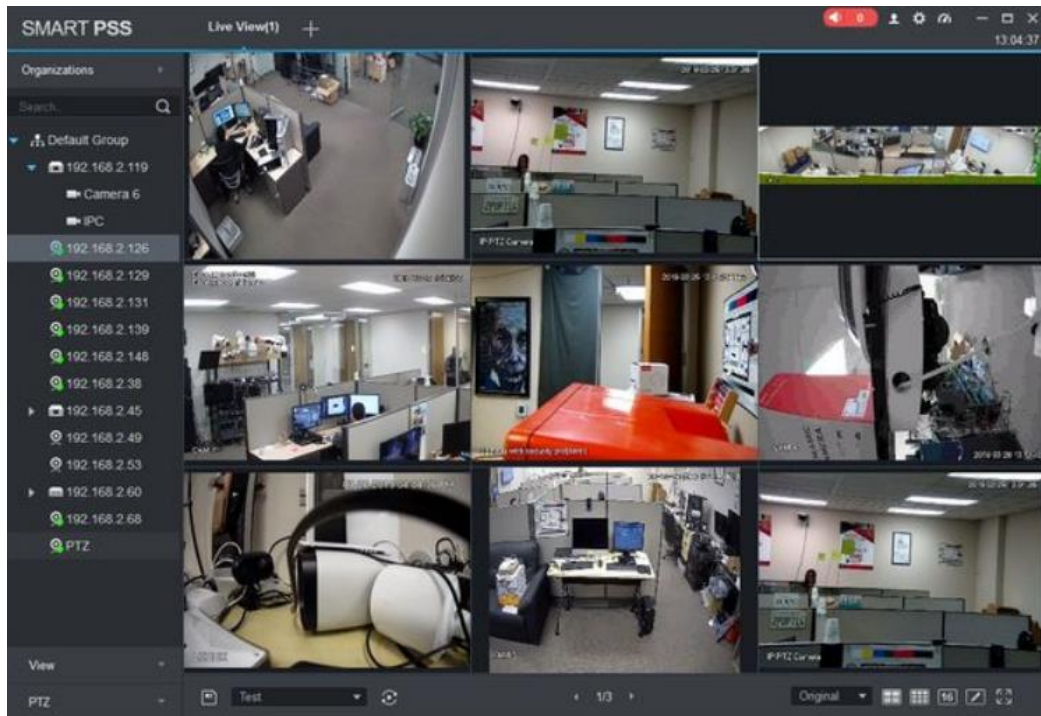


Figure 9 : Interface de visualisation de Smart PSS

Toutefois, l'utilisation de cette application implique une surveillance continue en temps réel des flux vidéo ou une analyse ultérieure des enregistrements pour détecter toute infraction commise. Cela engendre des dépenses et des efforts humains supplémentaires pour la détection d'éventuelles anomalies se rapportant à la présence d'un employé dans une zone où il n'est pas autorisé.

Même avec des ressources administratives et humaines élargies, il n'est pas toujours possible de détecter toutes les infractions dans un environnement où le nombre d'employés atteint quotidiennement 400 personnes, sans compter les visiteurs externes. La gestion d'un tel flux constant de personnes rend le suivi exhaustif des comportements et des activités particulièrement complexe, dépassant souvent la capacité des dispositifs de surveillance et de contrôle en place.

4 Conclusion

En conclusion, ce chapitre a permis de présenter l'entreprise d'accueil et de contextualiser le projet. Nous avons examiné le besoin croissant de l'entreprise de réduire le nombre d'incidents potentiellement non détectés par les méthodes traditionnelles de surveillance.

C'est dans cette optique que l'entreprise nous a proposé de développer un système autonome et plus robuste, capable de répondre efficacement à ces défis de sécurité et de conformité.

Enfin, il est essentiel à ce stade de bien comprendre les besoins et les fonctionnalités souhaitées, et de mettre en place une méthodologie adéquate pour le projet.

Chapitre 2

Analyse et spécification des besoins

1 Introduction

Ce chapitre se focalise sur la spécification et l'analyse des besoins, divisés en trois parties distinctes. La première partie aborde la présentation des acteurs impliqués dans la solution, la deuxième partie se concentre sur la spécification des besoins fonctionnels et non fonctionnel, tandis que la troisième partie traite de la gestion du projet avec SCRUM.

2 Identification des acteurs

Acteurs	Description
Système	Acteur apte à détecter les visages sur les images vidéo reçues, effectuer la reconnaissance faciale et enregistrer en images et en données chiffrées toute infraction commise.
Responsable sécurité	Acteurs aptes à s'authentifier à l'interface de visualisation afin de consulter en images et en données chiffrées toute infraction enregistrée par le système.

Tableau 1 : Identification des acteurs

3 Spécifications des besoins

Après avoir identifié les acteurs, la prochaine étape consiste à spécifier les besoins fonctionnels et non fonctionnels auxquels la solution proposée doit répondre.

3.1 Besoin fonctionnels

Les besoins fonctionnels spécifiques à chaque acteur sont les suivants :

Système

- Lire les images vidéo en continu
- Détecter les visages et enregistrer des captures des images entières et des images cadrant uniquement les visages
- Utiliser le modèle siamois de reconnaissance faciale pour identifier la personne détectée

- Vérifier dans la base de données la permission de la personne reconnue à apparaître sur la caméra source
- Sauvegarder une image étiquetée du cadre entier en cas d'infraction
- Enregistrer dans un fichier journal les détails de chaque infraction
- Comptabiliser le nombre d'apparitions dans les zones non autorisées de chaque personne détectée et reconnue

Responsable sécurité

- S'authentifier à l'interface utilisateur
- Consulter des images horodatées, identifiées et localisées des dernières infractions
- Consulter la liste des noms des employés à infractions récentes
- Consulter le nombre d'infractions journalières
- Consulter la mini-carte des zones autorisées et les chiffres sur le nombre d'apparition dans chaque zone non autorisée
- Sélectionner un employé et consulter l'historique d'infractions personnel en images horodatées et localisées

3.2 Besoin non-fonctionnels

Les besoins non-fonctionnels comprennent :

- **Performance** : Traitement fluide des flux des images vidéo et détection et reconnaissance des individus en temps réel
- **Précision** : Pourcentage de reconnaissance des individus élevé et minimalisation des fausses détections
- **Fiabilité** : Fonctionnement stable et sans interruption du système et capacité à gérer des conditions de réseau difficiles
- **Sécurité** : Protection des données sensibles se rapportant aux employés à identifier par le système
- **Facilité d'utilisation** : Interface de visualisation intuitive et facile à utiliser
- **Evolutivité** : Capacité à gérer un nombre croissant de caméras et de flux vidéo
- **Normalisation et aspects juridiques** : Respect de normes et réglementations relatives

3.3 Diagramme de cas d'utilisation global

Pour donner un aperçu des fonctionnalités offertes par notre système à chaque acteur, nous avons réalisé le diagramme de cas d'utilisation global ci-dessous.

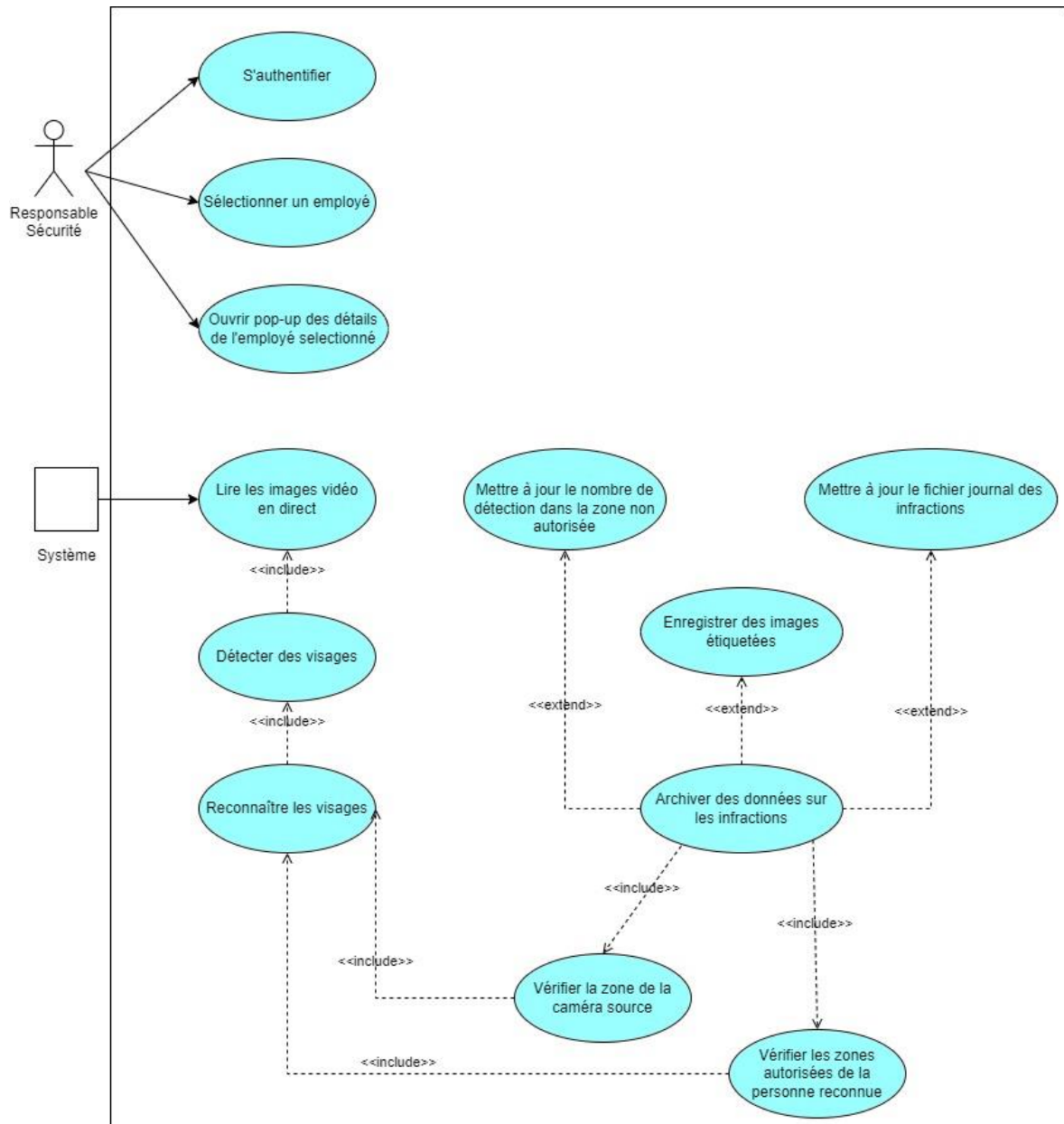


Figure 10 : Diagramme de cas d'utilisation global

4 Pilotage du projet avec SCRUM

SCRUM se présente comme une méthode de gestion de projet agile, conçue pour stimuler la productivité des équipes et optimiser le produit final grâce à des feedbacks réguliers des utilisateurs finaux. Ce cadre de travail agile fondamental permet également de déployer progressivement l'agilité à l'échelle de l'entreprise.

Dans cette section, nous allons mettre en lumière les différents éléments clés qui ont contribué à la mise en œuvre réussie du cadre SCRUM pour la gestion de notre projet.

4.1 Répartition des rôles

Dans le cadre de la méthodologie SCRUM, une répartition claire des rôles est essentielle pour assurer une collaboration efficace et une gestion optimale du projet.

SCRUM se distingue par sa structure organisée où chaque rôle, du Product Owner au SCRUM Master, joue une fonction cruciale pour garantir la réussite et la fluidité des processus de développement.

Rôle	Description	Affectation
Product Owner	Responsable de la maximisation de la valeur du produit en gérant et en priorisant le backlog du produit	Bassem Ben Hassen
Scrum Master	Facilite l'application des pratiques SCRUM, aide à éliminer les obstacles et favorise une communication efficace	Sofiane Chaaba
Development Team	Une équipe auto-organisée et multifonctionnelle qui délivre des incréments de produit potentiellement livrables à chaque sprint.	Adam Lassoued

Tableau 2 : Répartition des rôles dans SCRUM

4.2 Planification et plan des sprints

Dans le cadre de notre projet, nous avons prévu d'organiser un total de quatre sprints. Leurs planifications est un processus structuré en trois étapes essentielles [6] :

- 1- **Sprint Planning Meeting** (Réunion de planification de sprint) : L'équipe SCRUM définit les objectifs du sprint et sélectionne les tâches à réaliser.
- 2- **Daily SCRUM** (Réunion quotidienne) : Chaque jour, l'équipe discute de ce qui a été accompli, des prochaines étapes et des éventuels obstacles.
- 3- **Sprint Review** (Revue de sprint) : À la fin du sprint, l'équipe présente le travail réalisé et obtient des feedbacks pour ajuster les futures itérations.

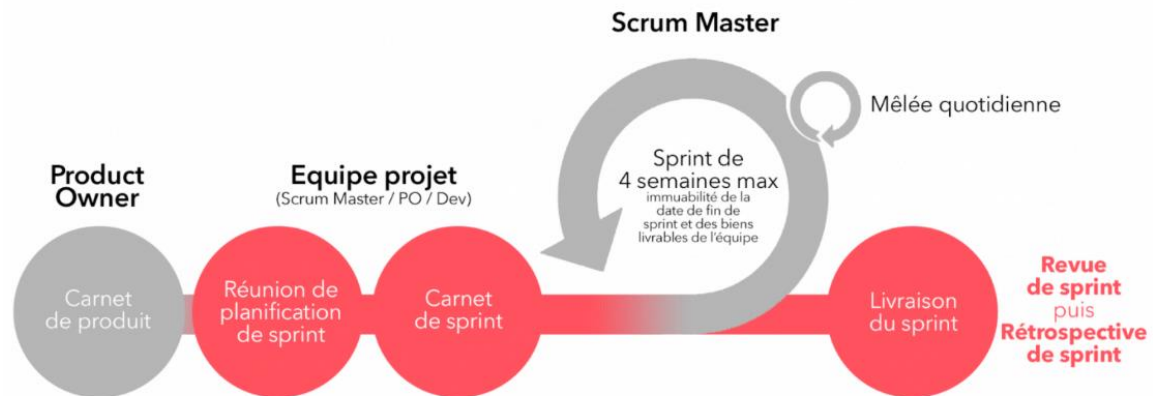


Figure 11 : Cycle d'un projet piloté avec SCRUM

Au total, nous avons réparti la réalisation du projet en 4 sprints. Chacun de ces sprints a été programmé selon la période décrite dans le tableau ci-dessous :

ID	Sprint	Date de début	Date de fin
1	Sprint 1 : Intégration de la détection des visages dans le système	01/03/2024	21/03/2024
2	Sprint 2 : Intégration de la reconnaissance faciale dans le système	22/03/2024	25/04/2024
3	Sprint 3 : Implémentation de la partie contrôle dans le système de contrôle	26/04/2024	16/05/2024
4	Sprint 4 : Développement de l'interface graphique de visualisation	17/05/2024	20/06/2024

Tableau 3 : Programmation des sprints

4.3 Product Backlog

Le Product Backlog est un élément crucial du framework SCRUM, servant de liste ordonnée et évolutive des éléments nécessaires à l'amélioration d'un produit. Il représente la source unique de travail pour l'équipe SCRUM et constitue un outil indispensable pour la planification et la gestion du développement du produit.

En tant que colonne vertébrale du processus SCRUM, le Product Backlog est maintenu par le Product Owner, qui assume la responsabilité de sa priorisation et de sa clarification continue. Il s'agit d'une liste dynamique qui évolue au fil du projet, s'enrichissant de nouvelles fonctionnalités, de corrections de bugs et d'autres éléments identifiés comme nécessaires pour répondre aux objectifs du produit.

ID	Fonctionnalité	ID	User story
1	Détection de visages	1.1	En tant que système, je souhaite avoir accès à des images vidéo en continu.
		1.2	En tant que système, je souhaite détecter des visages sur les images vidéo reçue.
		1.3	En tant que système, je souhaite stocker temporairement des images du cadre entier et des images du cadre d'un visage détecté.
2	Reconnaissance Faciale	2.1	En tant que système, je souhaite recevoir des images du cadre d'un visage détecté.
		2.2	En tant que système, je souhaite vérifier l'image du visage reçue contre le modèle de reconnaissance faciale.
		2.3	En tant que système, je souhaite retenir l'identité de la personne détectée.
3	Contrôle	3.1	En tant que système, je souhaite vérifier la zone à laquelle appartient la caméra source.
		3.2	En tant que système, je souhaite vérifier les zones autorisées à l'employé identifié.
		3.3	En tant que système, je souhaite mettre à jour le nombre d'apparitions sur les zones non-autorisées pour l'employé identifié.
		3.4	En tant que système, je souhaite enregistrer l'image du cadre entier de l'infraction enregistrée.
		3.5	En tant que système, je souhaite ajouter dans un fichier journal les détails de chaque infraction.
4	Visualisation des données	4.1	En tant que responsable de la sécurité, je souhaite visualiser des images identifiées et horodatées et localisées des infractions récentes.
		4.2	En tant que responsable de la sécurité, je souhaite consulter le nombre d'infractions journalières enregistrée.
		4.3	En tant que responsable de la sécurité, je souhaite consulter la liste des derniers employés ayant commis des infractions.
		4.4	En tant que responsable de la sécurité, je souhaite visualiser une mini-carte des zones autorisées et des zones non-autorisées pour un employé donné.
		4.5	En tant que responsable de la sécurité, je souhaite consulter le nombre d'infractions dans chaque zone non-autorisée pour un employé donné.
		4.6	En tant que responsable de la sécurité, je souhaite consulter un historique en images horodatées et localisées de toutes les infractions commises par un employé donné.

Tableau 4 : Product Backlog

5 Conclusion

En conclusion, ce chapitre a exploré en détail la spécification et l'analyse des besoins du projet, en mettant l'accent sur trois aspects essentiels : la présentation des acteurs clés, la définition des besoins fonctionnels et non fonctionnels, ainsi que la gestion de projet avec SCRUM. À présent, nous sommes prêts à passer à l'étape suivante : définir l'architecture de notre système et affiner les détails de chaque sprint. Cela nous permettra de concrétiser nos concepts en solutions techniques efficaces et de progresser vers la réalisation de notre projet de manière structurée et méthodique.

Chapitre 3

Conception

1 Introduction

Dans ce chapitre, nous allons détailler la conception de notre application. Nous allons fournir une vue d'ensemble de l'architecture et une vue plus détaillée de chaque composante. Nous allons explorer en premier lieu les différentes composantes et interconnexions qui forment cette architecture. Par la suite, pour une compréhension approfondie du processus de développement, chaque sprint SCRUM sera analysé en détail, mettant en lumière les étapes clés et les livrables associés. Ce chapitre vise à offrir une compréhension claire et exhaustive de la structure et de l'évolution de notre système.

2 Architecture

Afin de mieux comprendre le fonctionnement interne de l'application, nous allons maintenant nous pencher sur son architecture. C'est en l'examinant que nous allons saisir la manière dont ses différents éléments interagissent entre eux.

Nous avons opté pour un modèle en 3 tiers. Ce design repose sur une couche de présentation assurée par une interface utilisateur intuitive, une couche applicative qui contient la logique métier et traite les requêtes, et une couche de données qui gère le stockage et l'accès aux données essentielles. Une couche d'infrastructure supplémentaire relie ces trois couches, assurant la communication, la sécurité et la gestion des ressources réseau. [7]

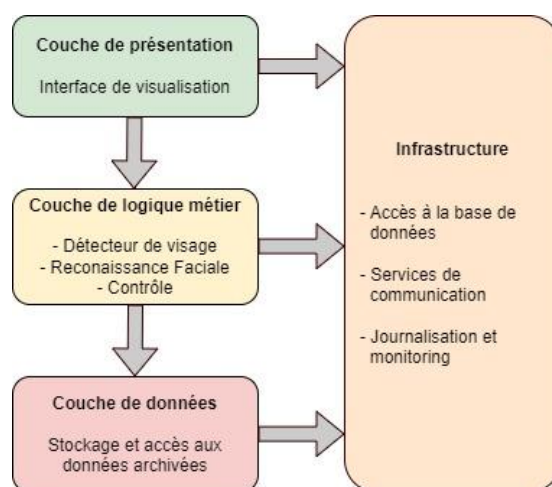


Figure 12 : Architecture de l'application

L'adoption d'une architecture en 3 tiers présente plusieurs avantages significatifs, les plus importants étant :

- Une séparation claire des préoccupations entre la présentation, la logique métier et l'accès aux données.
- Une facilité de maintenance, de test et de déploiement grâce au développement indépendant de chaque couche.
- Un renforcement de la sécurité par un contrôle centralisé des accès aux données sensibles.
- Une amélioration des performances grâce à une répartition efficace de la charge de travail.

3 Conception détaillée

Dans cette section, nous allons explorer en détail chaque sprint du projet. Nous commencerons par définir le Sprint Backlog, qui contient les tâches à accomplir pour chaque itération. Ensuite, nous présenterons des diagrammes de séquence pour illustrer le fonctionnement de chaque composante du système. Ces diagrammes permettront de visualiser les interactions entre les différentes parties de l'application entière et de mieux comprendre leur intégration et leur fonctionnement global.

3.1 Sprint 1 : Intégration de la détection des visages dans le système

Dans cette section, nous allons détailler l'utilisation d'un modèle de détection de visages pré-entraîné appelé Haar Cascade. Nous expliquerons comment ce modèle est intégré dans la composante détectrice de visages. De plus, nous décrirons le fonctionnement global de cette composante au sein du système.

3.1.1 Backlog du Sprint 1

ID	Description
US-01	En tant que système, je souhaite me connecter à la camera source via le nom d'utilisateur et le mot de passe de configuration.
US-02	En tant que système, je souhaite charger de la modèle de détection de visage Haar Cascade.
US-03	En tant que système, je souhaite lire des images vidéo en continu.
US-04	En tant que système, je souhaite détecter la présence d'un visage.
US-05	En tant que système, je souhaite capturer une image uniquement le cadre du visage.

US-06	En tant que système, je souhaite capturer une image entière contenant la personne au visage détecté.
US-07	En tant que système, je souhaite sauvegarder temporairement l'image du visage détecté.
US-08	En tant que système, je souhaite sauvegarder temporairement l'image entière contenant la personne au visage détecté.

Tableau 5 : Backlog du sprint 1

3.1.2 Diagramme de séquence du Sprint 1

Le diagramme ci-dessous illustre le fonctionnement séquentiel de la sous-composante de détection de visages au sein du système.

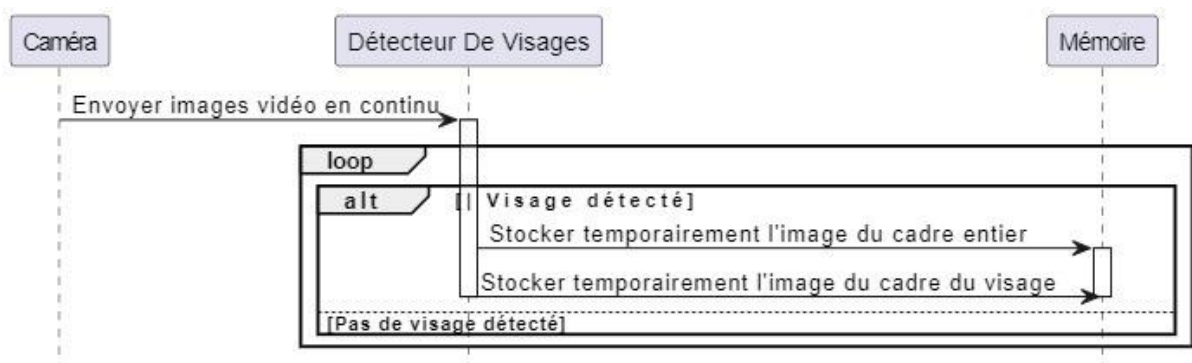


Figure 13 : Diagramme de séquence du sprint 1

3.2 Sprint 2 : Intégration de la reconnaissance faciale dans le système

Dans cette section, nous allons explorer l'intégration du modèle de la reconnaissance faciale dans la composante dédiée de notre système. De plus, nous décrirons le fonctionnement global de cette composante.

3.2.1 Backlog du Sprint 2

ID	Description
US-01	En tant que système, je souhaite recevoir des images des visages capturés.
US-02	En tant que système, je souhaite fournir la valeur des hyper paramètres des seuils de détection et de vérification à la fonction de vérification.
US-03	En tant que système de contrôle, je souhaite accéder au modèle siamois entraîné à reconnaître les employés.
US-04	En tant que système, je souhaite vérifier l'image du visage reçu contre le modèle de reconnaissance faciale pour identifier la personne.
US-05	En tant que système de contrôle, je souhaite stocker temporairement l'identité de la personne identifiée.

Tableau 6 : Backlog du sprint 2

3.2.2 Diagramme de séquence du Sprint 2

Le diagramme ci-dessous montre le fonctionnement séquentiel de la sous-composante de reconnaissance faciale au sein du système.

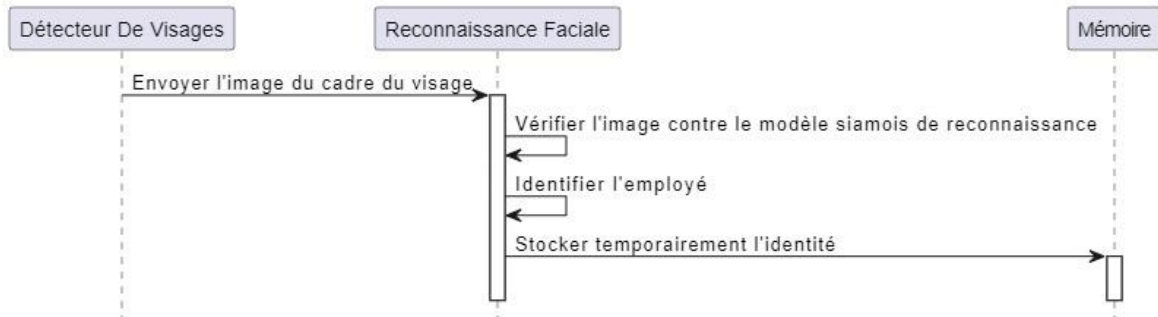


Figure 14 : Diagramme de séquence du sprint 2

3.3 Sprint 3 : Implémentation de la partie contrôle dans le système

Dans cette section, nous allons nous concentrer sur le contrôle au sein du système. Nous expliquerons les mécanismes mis en œuvre, en précisant les sous-composantes impliquées dans cette opération. De plus, nous décrirons le fonctionnement global de cette composante.

3.3.1 Backlog du Sprint 3

ID	Description
US-01	En tant que système, je souhaite me connecter à la base de données.
US-02	En tant que système, je souhaite vérifier la zone à laquelle appartient la caméra source.
US-03	En tant que système, je souhaite vérifier les zones autorisées à la personne identifiée.
US-04	En tant que système, je souhaite mettre à jour le nombre d'apparitions sur les zones non-autorisées pour une personne identifiée.
US-05	En tant que système, je souhaite enregistrer définitivement l'image du cadre entier de l'infraction enregistrée.
US-07	En tant que système, je souhaite enregistrer la date et heure de l'évènement d'infraction.
US-08	En tant que système, je souhaite générer une ligne de texte formatée contenant l'identité, la date et heure et zone de l'infraction commise.
US-09	En tant que système, je souhaite ajouter le texte formaté généré au fichier journal listant les infractions.

Tableau 7 : Backlog du sprint 3

3.3.2 Diagramme de séquence du sprint 3

Le diagramme suivant illustre le fonctionnement de la partie contrôle du système, reliant les sous-composantes de détection et de reconnaissance faciale d'une part, et la base de données d'autre part.

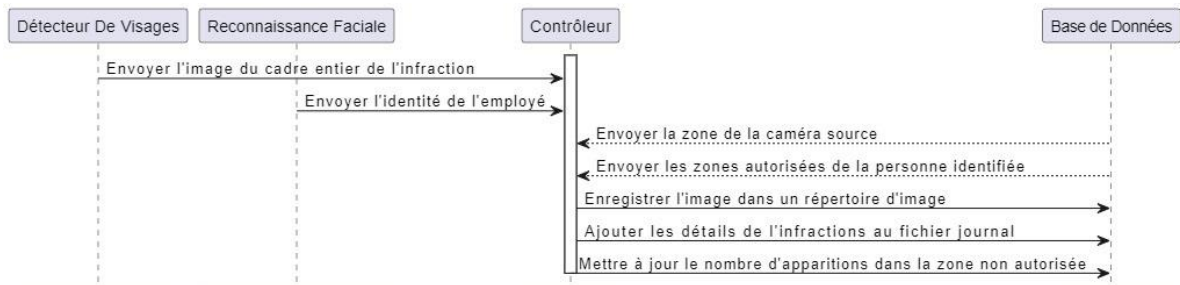


Figure 15 : Diagramme de séquence du sprint 3

3.3.3 Synthèse du fonctionnement global du système : Diagramme d'activités

Après avoir détaillé la conception de chaque sous-composante du système, nous passons maintenant à la modélisation du fonctionnement global du système qui les englobe et définit leurs interactions.

Le diagramme d'activités suivant illustre les différents scénarios d'exécution.

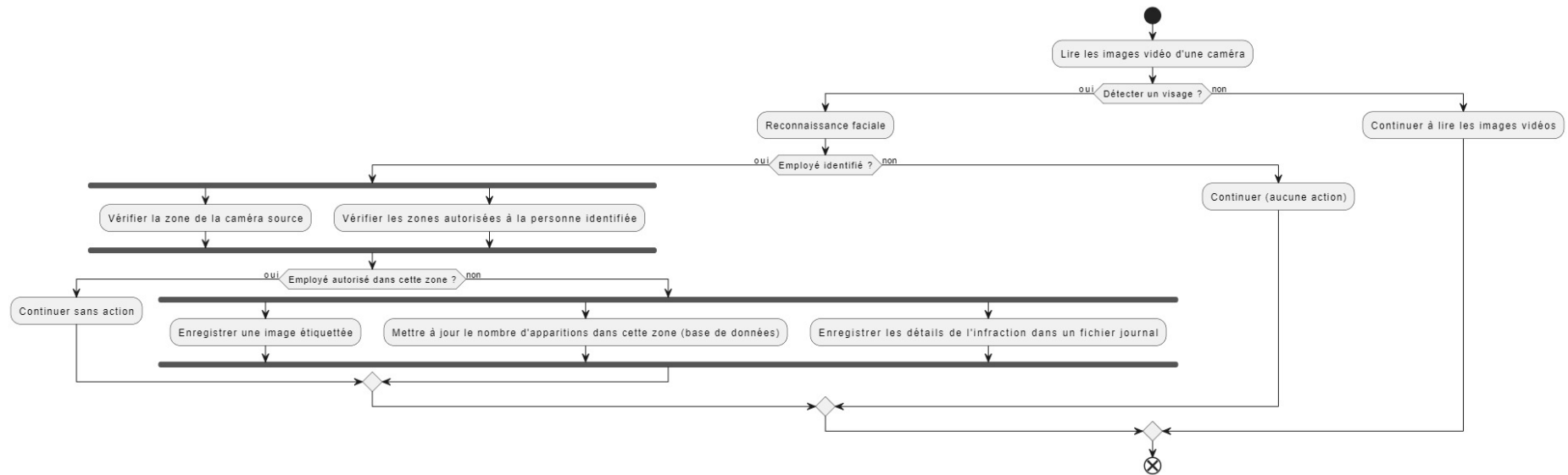


Figure 16 : Diagramme d'activités du système

3.4 Sprint 4 : Développement de l'interface graphique de visualisation

Dans cette section, nous mettons l'accent sur la conception de l'interface de visualisation des données, réalisée pendant le sprint 4. Nous décrivons les fonctions disponibles pour le responsable de la sécurité dans cette interface, ainsi que la nature des données affichées à l'écran.

3.4.1 Backlog du Sprint 4

ID	Description
US-01	En tant que responsable de la sécurité, je souhaite m'authentifier.
US-02	En tant que responsable de la sécurité, je souhaite consulter les images identifiées, horodatées et localisées des infractions récentes.
US-03	En tant que responsable de la sécurité, je souhaite consulter le nombre d'infractions journalières enregistrées.
US-04	En tant que responsable de la sécurité, je souhaite consulter dans un ordre décroissant la liste des derniers employés à commettre des infractions.
US-05	En tant que responsable de la sécurité, je souhaite sélectionner un employé parmi les employés ayant commis des infractions.
US-06	En tant que responsable de la sécurité, je souhaite visualiser une mini-carte présentant les zones autorisées et les zones non-autorisées d'un employé sélectionné.
US-07	En tant que responsable de la sécurité, je souhaite visualiser une table contenant le nombre d'apparitions d'un employé sélectionné dans chaque zone non-autorisée à lui.
US-08	En tant que responsable de la sécurité, je souhaite ouvrir un pop-up contenant plus de détails sur l'employé sélectionné.
US-09	En tant que responsable de la sécurité, je souhaite visualiser le nombre d'infractions totales et journalières dans le pop-up personnel.
US-10	En tant que responsable de la sécurité, je souhaite consulter les images horodatées et localisées de toutes les infractions dans le pop-up personnel.

Tableau 8 : Backlog du sprint 4

3.4.2 Diagramme de séquence du Sprint 4

Ce diagramme présente la séquence d'interactions entre le responsable de la sécurité et l'interface de visualisation, débutant par l'authentification et se poursuivant jusqu'à l'accès aux informations personnelles les plus détaillées.

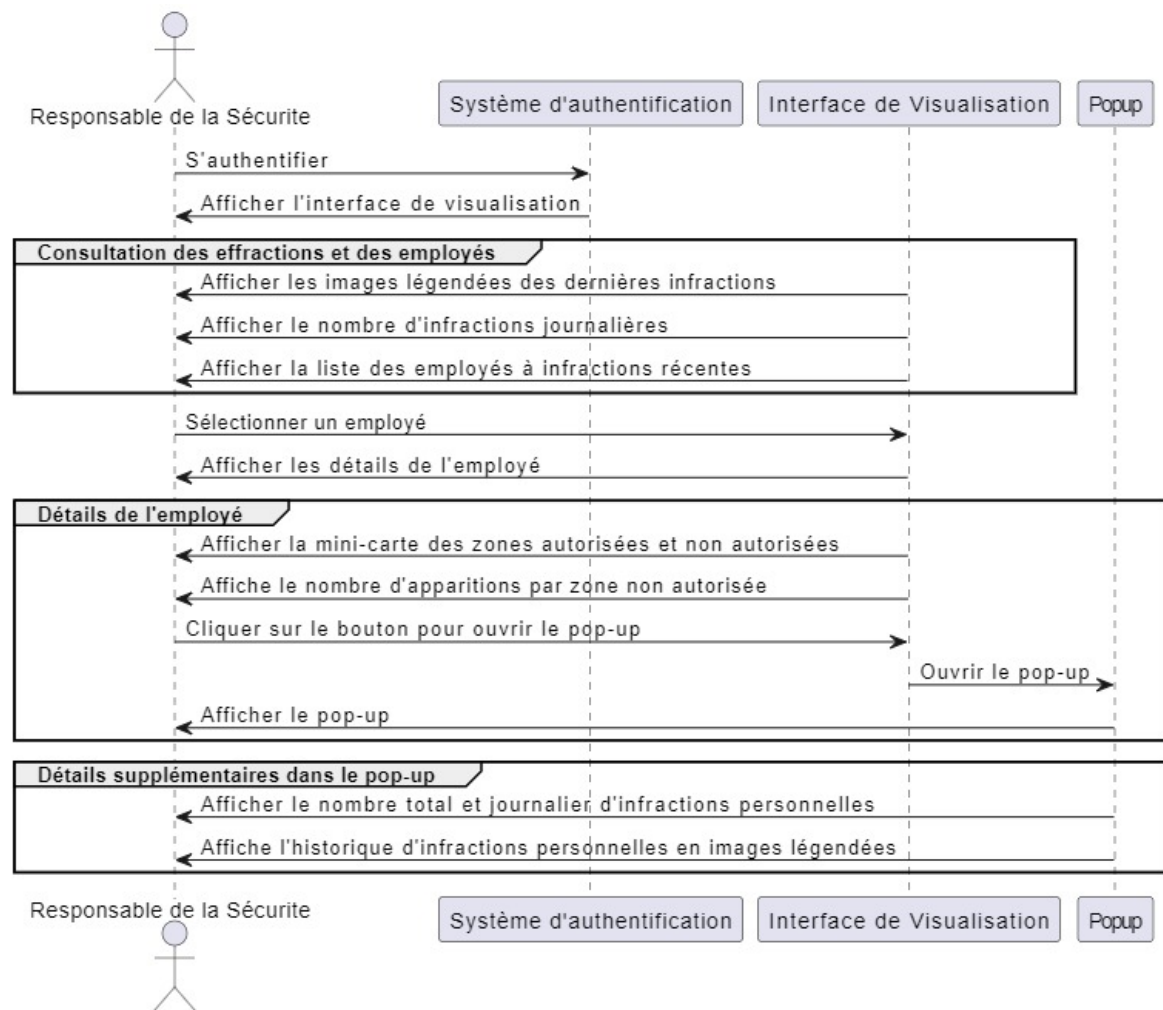


Figure 17 : Diagramme de séquence du sprint 4

4 Conclusion

En conclusion, ce chapitre a permis d'approfondir la conception de notre système en offrant une vue d'ensemble détaillée de son architecture ainsi qu'une exploration approfondie de chaque composante. Dans le prochain chapitre, nous explorerons les modèles de l'intelligence artificielle adoptés, essentiels pour la mise en œuvre efficace de notre solution, afin de répondre aux besoins et aux exigences identifiés dans la phase de conception.

Chapitre 4

Étude des modèles de l'intelligence artificielle adoptés

1 Introduction

Dans ce chapitre, nous allons nous pencher sur les deux modèles de l'intelligence artificielle utilisés dans notre projet. Le premier modèle est un modèle de détection de visage pré-entraîné, choisi pour sa robustesse et sa précision éprouvées dans l'identification des traits faciaux. Le second modèle est un modèle siamois que nous avons développé spécifiquement au cours d'une étape ultérieure du projet pour les besoins de la reconnaissance faciale. Cette analyse comprendra une présentation détaillée de chaque modèle, leurs principes de fonctionnement, ainsi que les résultats obtenus lors de leur implémentation.

2 Modèle de détection de visages

Dans cette section, nous allons présenter la détection d'objets et le modèle de détection de visages déployé.

2.1 Définition de la détection d'objets

La détection d'objets est une technique de traitement d'images qui permet de localiser et d'identifier des objets spécifiques dans une image ou une vidéo. Elle repose sur des algorithmes de vision par ordinateur qui analysent les caractéristiques de l'image pour détecter la présence d'objets d'intérêt, tels que des personnes, des voitures, des animaux, des bâtiments, etc. Cette technique est largement utilisée dans de nombreux domaines, notamment la sécurité, la surveillance, l'industrie, la robotique, la santé et bien plus encore.

2.2 Le modèle déployé : Haar Cascade Frontal Face

Le modèle Haar Cascade est une méthode de détection d'objets largement utilisée, particulièrement Haar Cascade Frontal Face pour la détection de visages. Il s'agit d'un algorithme de classification basé sur l'apprentissage supervisé, qui a été entraîné sur un ensemble de données volumineux d'images contenant des visages et des arrière-plans.



Figure 18 : Détection de visages utilisant Haar Cascade

2.2.1 Le fonctionnement du modèle

Le modèle Haar Cascade repose sur trois principes fondamentaux [10] :

1. **Extraction de caractéristiques :** Le modèle utilise des caractéristiques de Haar, qui sont des rectangles simples de différentes tailles et orientations, pour représenter les visages dans les images. Ces caractéristiques permettent de capturer des traits distinctifs des visages, tels que les yeux, le nez et la bouche.
2. **Apprentissage par boosting :** Le modèle est entraîné à l'aide d'une technique appelée boosting, qui combine plusieurs classificateurs faibles en un seul classificateur plus performant. Chaque classificateur faible se concentre sur une caractéristique de Haar spécifique et détermine si elle est présente ou non dans une région de l'image. Les classificateurs faibles sont ensuite combinés de manière adaptative, en attribuant plus d'importance à ceux qui ont de meilleures performances.
3. **Détection de visages :** Pour détecter un visage dans une image, le modèle Haar Cascade déplace une fenêtre de détection de taille fixe sur l'image. À chaque position de la fenêtre, les caractéristiques de Haar sont calculées et évaluées par les

classificateurs faibles. Si suffisamment de classificateurs indiquent la présence d'un visage, la fenêtre est considérée comme contenant un visage.

2.2.2 Avantages et limites

Ce modèle présente plusieurs avantages et limites. Nous détaillons les plus importants dans le tableau suivant :

Avantages	Limites
Efficacité : Le modèle Haar Cascade est un algorithme relativement simple et rapide à exécuter, ce qui le rend adapté aux applications en temps réel.	Performance : La performance du modèle peut être affectée par la qualité de l'image et la présence d'objets perturbateurs.
Robustesse : Le modèle est robuste aux variations d'éclairage, de pose et d'expression des visages.	Détection de faux positifs : Le modèle peut parfois détecter des objets non-visages comme des visages, ce qui génère des faux positifs.
Facilité d'utilisation : Le modèle est facilement disponible dans des bibliothèques d'apprentissage automatique populaires, telles que OpenCV, ce qui le rend accessible à un large public.	Détection de faux négatifs : Le modèle peut également manquer de détecter certains visages, ce qui génère des faux négatifs.

Tableau 9 : Avantages et limites du modèle Haar Cascade

3 Modèle de reconnaissance faciale

Dans cette section, nous présentons le concept de la reconnaissance faciale et le modèle développé, appelé modèle Siamois.

3.1 Définition de la reconnaissance faciale

La reconnaissance faciale est une technologie de traitement d'images et de vision par ordinateur qui permet d'identifier ou de vérifier l'identité d'une personne à partir d'une image ou d'une vidéo de son visage. Cette technologie repose sur des algorithmes d'apprentissage automatique pour analyser les caractéristiques faciales uniques, telles que la distance entre les yeux, la forme du nez, la structure des pommettes, et d'autres traits distinctifs. La reconnaissance faciale est couramment utilisée dans des applications de sécurité, de surveillance, d'authentification biométrique, ainsi que dans des outils de gestion et de catégorisation de photos.

3.2 Le modèle développé : Le modèle siamois

Dans cette section, nous allons introduire le modèle de reconnaissance que nous avons décidé de développer pour notre solution, à savoir le réseau de neurones siamois.

3.2.1 Introduction aux réseaux de neurones siamois

Les réseaux de neurones siamois se sont imposés comme un outil puissant pour la reconnaissance faciale, permettant de comparer et de discriminer des paires d'images de manière efficace. Cette section présente les étapes techniques clés du développement d'un modèle siamois pour la reconnaissance faciale, en abordant les principes des réseaux de neurones, le type de données requis, les considérations architecturales, la configuration des entrées et les paramètres de test et d'évaluation. [11]

3.2.2 Principes des réseaux de neurones siamois

Un réseau siamois se compose de deux branches identiques qui traitent des entrées indépendantes, mais partagent les mêmes paramètres de réseau. Les sorties de ces branches sont ensuite comparées à l'aide d'une fonction de similarité, telle que la distance euclidienne ou le cosinus de similarité. [8] L'objectif est d'apprendre à générer des représentations similaires pour les entrées appariées (visages de la même personne) et des représentations distinctes pour les entrées non appariées (visages de personnes différentes).

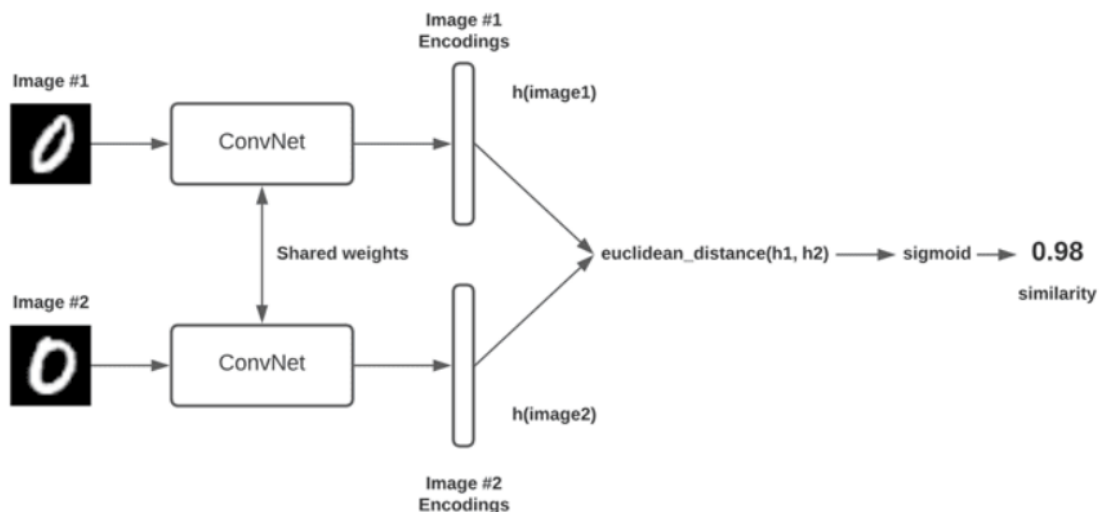


Figure 19 : Calcul de similarité dans un réseau de neurones siamois

3.2.3 Architecture du réseau

L'architecture d'un réseau siamois pour la reconnaissance faciale peut varier en fonction de la complexité du problème et de la quantité de données disponibles. Cependant, une architecture courante implique l'utilisation d'un réseau convolutif profond (CNN) pour extraire des caractéristiques des images d'entrée. Les sorties des CNN sont ensuite aplaties et projetées dans un espace vectoriel de dimension réduite, représentant l'empreinte faciale de chaque image.

Cette architecture implique l'utilisation de plusieurs couches ("layers") empilées les unes sur les autres. Chaque couche joue un rôle spécifique dans le processus d'extraction et de traitement des caractéristiques des images d'entrée. [9]

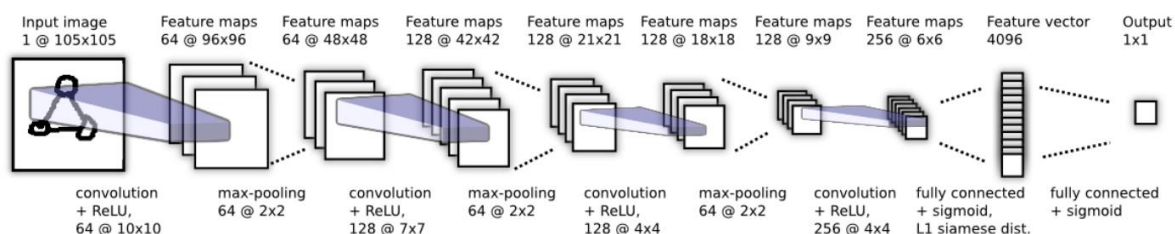


Figure 20 : Architecture convolutive sélectionnée pour la tâche de vérification

On peut distinguer deux types principaux de couches [12] :

- **Couches convolutives:** Ces couches appliquent des filtres convolutifs aux images d'entrée pour extraire des caractéristiques locales, telles que des bords et des textures.
- **Couches de pooling:** Ces couches réduisent la dimensionnalité des sorties des couches convolutives en appliquant des fonctions de pooling, comme la moyenne ou le maximum.

3.2.4 Données d'entraînement

Pour entraîner efficacement un modèle siamois pour la reconnaissance faciale, il est crucial de disposer d'un ensemble de données adéquat composé d'images appariées et non appariées.

- **Images positives :** Paires d'images représentant la même personne sous différentes conditions d'éclairage, d'angle et d'expression.

- **Images négatives** : Paires d'images représentant des personnes différentes, sélectionnées de manière aléatoire dans l'ensemble de données.
- **Images ancrées** : Un sous-ensemble d'images positives utilisé comme référence pour comparer les autres images.

3.2.5 Processus d'entraînement

L'entraînement d'un modèle siamois implique l'itération sur les paires d'images positives et négatives, en calculant la similarité entre les représentations faciales et en ajustant les paramètres du réseau pour minimiser la distance entre les représentations appariées et maximiser la distance entre les représentations non appariées.

Cela est assuré par la fonction de perte triplet (« Triplet Loss Function ») [13] qui permet à notre modèle de rapprocher deux images similaires et d'éloigner une image dissimilaire par rapport à une paire d'images d'échantillon. Elle utilise une image ancre comme référence, une image positive similaire à l'ancre pour renforcer les similitudes, et une image négative différente pour apprendre à distinguer les dissimilarités par rapport à l'ancre.

3.2.6 Configuration des entrées

La configuration des entrées peut avoir un impact significatif sur les performances du modèle siamois. Parmi les paramètres importants à considérer, on trouve :

- **Taille d'entrée** : La taille des images d'entrée doit être compatible avec l'architecture du réseau et les exigences de calcul.
- **Prétraitement des images** : Le prétraitement des images d'entrée, comme la normalisation et la correction de l'éclairage, peut améliorer la robustesse du modèle.
- **Augmentation des données** : L'augmentation des données, comme le recadrage aléatoire et la rotation des images, peut artificiellement augmenter la taille de l'ensemble de données et améliorer les performances du modèle.

3.2.7 Tests et évaluation

Pour évaluer les performances d'un modèle siamois entraîné, il est important de définir des paramètres de test appropriés. Ces paramètres incluent :

- **Ensemble de données de test** : Un ensemble de données de test distinct et non vu pendant l'entraînement est utilisé pour évaluer la capacité de généralisation du modèle.
- **Métriques d'évaluation** : Des métriques d'évaluation telles que la précision, le rappel et la F1-score sont utilisées pour quantifier les performances du modèle sur l'ensemble de données de test.
- **Seuils de décision** : Le choix d'un seuil de décision approprié pour la sortie du modèle est crucial pour déterminer si deux images représentent la même personne ou non.

4 Conclusion

En conclusion, ce chapitre a exploré en profondeur les deux modèles de l'intelligence artificielle clés utilisés dans notre projet : un modèle de détection de visage pré-entraîné reconnu pour sa robustesse, ainsi qu'un modèle siamois développé spécifiquement pour la reconnaissance faciale. Dans le prochain chapitre, nous examinerons de manière détaillée les méthodes et les ressources utilisées pour la réalisation concrète de notre projet, mettant en lumière les étapes de développement et les outils essentiels qui ont contribué à sa mise en œuvre réussie.

Chapitre 5

Réalisation

1 Introduction

Dans ce dernier chapitre, nous exposerons la réalisation du projet en présentant d'abord l'environnement de travail, comprenant les environnements logiciel et matériel. Nous détaillerons ensuite la mise en œuvre de notre système et de l'interface graphique, en expliquant les principes et démarches suivis lors de chaque sprint.

2 Environnement de travail

La section suivante présente en détail l'environnement de travail utilisé pour le développement de notre projet, englobant à la fois les aspects logiciels et matériels

2.1 Environnement matériel

Pour la réalisation du projet, j'ai utilisé un PC de la marque Acer avec la configuration suivante :

- **Système d'exploitation** : Windows 11 Professionnel
- **Processeur** : Intel(R) Core(TM) i5-1135G7, 11^{ème} génération (2.40GHz)
- **Mémoire RAM** : 8 Go
- **Disque Dur** : MSI SPATIUM-M480 1Tb SSD (lecture—écriture : 7000 mb/s)

2.2 Environnement logiciel

Cette section vise à décrire l'environnement logiciel du projet, y compris les technologies, langages, bibliothèques, et logiciel utilisés.

2.2.1 Environnement de développement

- **Outil de développement** : Visual Studio Code



Figure 21 : logo Visual Studio Code

Visual Studio Code est un éditeur de code source libre et multiplateforme développé par Microsoft. Il offre une multitude de fonctionnalités telles que le debugging, le contrôle de

version intégré avec Git, la coloration syntaxique, l'auto-complétion intelligente, ainsi qu'un large éventail d'extensions pour divers langages de programmation

- Outil de modélisation : draw.io



Figure 22 : logo draw.io

Draw.io est un outil de modélisation gratuit, en versions web et desktop, qui permet de créer facilement des diagrammes de flux, des organigrammes, des diagrammes UML, des maquettes et bien plus encore. Il est particulièrement apprécié pour son interface intuitive et ses nombreuses fonctionnalités, telles que la collaboration en temps réel, l'intégration avec divers services cloud comme Google Drive et OneDrive, et une vaste bibliothèque de formes et de modèles.

2.2.2 Langages et bibliothèques

- Python



Figure 23 : logo Python

Python est un langage de haut niveau, largement reconnu pour sa simplicité et sa lisibilité. Pour ce projet, nous l'avons utilisé comme langage de programmation principal. Il est particulièrement adapté aux systèmes de détection d'objets et de reconnaissance faciale grâce à des bibliothèques dédiées telles que TensorFlow, Keras et OpenCV. Ces bibliothèques permettent une implémentation efficace et rapide des algorithmes de deep learning et de vision par ordinateur. De plus, Python offre des bibliothèques comme Tkinter et PyQt pour le développement d'interfaces graphiques, facilitant ainsi la création d'applications interactives et conviviales.

- OpenCV / CV2



Figure 24 : logo OpenCV

OpenCV est une bibliothèque open source spécialisée dans le traitement d'images et de vidéos. Elle offre un vaste ensemble de fonctions pour des tâches telles que la manipulation d'images, la détection d'objets, et la reconnaissance de formes. Dans notre projet, OpenCV joue un rôle crucial pour le traitement des flux vidéo, permettant de capturer, analyser et manipuler les images en temps réel. Nous avons intégré le modèle de

détection de visage Haar Cascade avec OpenCV pour détecter les visages dans les vidéos. Il est à noter que le package cv2 est l'interface d'OpenCV qui permet de l'utiliser avec Python.

- TensorFlow



Figure 25 : logo TensorFlow

TensorFlow est une bibliothèque open source développée par Google pour le calcul numérique et l'apprentissage automatique à grande échelle. Elle fournit une infrastructure robuste et flexible pour déployer des modèles de machine learning et de deep learning. Dans notre projet, TensorFlow est utilisé pour la construction et l'entraînement du modèle siamois de reconnaissance faciale. Grâce à ses capacités de traitement efficace des données et à son support pour l'exécution sur GPU, TensorFlow permet de gérer de grands volumes d'images et d'optimiser les performances du modèle pour une reconnaissance précise et rapide des visages.



Figure 26 : logo Keras

- Keras

Keras est une bibliothèque open source de haut niveau, intégrée dans TensorFlow, qui facilite la création et l'entraînement de réseaux de neurones profonds, ce qui est crucial pour le développement de notre modèle siamois de reconnaissance faciale. Son intégration transparente avec TensorFlow permet un entraînement et une évaluation efficaces du modèle.



- Tkinter

Figure 27 : logo Tkinter

Tkinter est une bibliothèque standard de Python utilisée pour créer des interfaces graphiques (GUI). Elle permet de développer des applications interactives et conviviales grâce à une variété de widgets tels que des boutons, des étiquettes, des champs de texte, et des cadres. Dans notre projet, nous avons utilisé Tkinter pour développer une interface graphique permettant de visualiser les données sur les intrusions enregistrées. Cette interface affiche les informations pertinentes telles que les images capturées, les personnes

identifiées, les horaires des intrusions et les données statistiques associées, offrant ainsi un outil intuitif pour la surveillance et l'analyse des incidents de sécurité.

2.2.3 Base de données

- MySQL Workbench



Figure 28 : logo MySQL Workbench

MySQL Workbench est un outil visuel complet pour la gestion de bases de données MySQL. Il offre des fonctionnalités telles que la conception de schémas, l'administration du serveur, et la gestion des requêtes SQL. Dans notre projet, nous avons utilisé MySQL Workbench pour concevoir, administrer et interagir avec notre base de données. Il a facilité la création de tables pour stocker les données relatives aux intrusions, à la répartition des caméras sources par zone et aux zones autorisées par employé. Grâce à son interface intuitive, nous avons pu gérer efficacement les requêtes et optimiser les performances de notre base de données, assurant ainsi une gestion fluide et structurée des données du projet.

3 Mise en œuvre

Dans cette section, nous allons expliquer concrètement la mise en œuvre de notre solution en détaillant la réalisation de chaque composante de notre projet. Cette description comprendra les aspects techniques et les choix méthodologiques qui ont guidé notre travail, offrant une vision complète et précise de notre démarche.

Il est important de noter que les employés dont les noms et images apparaissent au cours de ce chapitre ont donné leur consentement pour l'utilisation de leurs données à des fins de démonstration. Toutefois, pour des raisons de sécurité, aucune adresse IP ni données de configuration réelles des caméras installées ne seront affichées.

3.1 La détection de visages

L'objectif à ce niveau est de détecter la présence d'un visage et d'en capturer des images. Afin de l'atteindre, nous avons couplé OpenCV pour la lecture des flux vidéo au modèle pré-entraîné Haar Cascade Frontal Face pour la détection des visages dans ces images. [14] Nous avons utilisé la détection et la capture à 2 niveaux différents selon lesquels le nombre d'images à capturer varie.

Dans un premier temps, nous visons à collecter un grand nombre d'images pour un employé donné, afin de les utiliser d'une part pour l'entraînement du modèle de reconnaissance faciale et d'autre part comme images de vérification au niveau de la fonction de vérification du système.

Ensuite, quand le système est en marche, nous souhaitons capturer une image unique à chaque fois qu'un nouveau visage est détecté. Cette image servira d'entrée pour la composante de reconnaissance faciale.

Ainsi, en enregistrant les champs cadrés avec précision par le modèle Haar Cascade, nous aurons obtenus à partir d'images vidéo une ou plusieurs images présentant uniquement le visage qui y apparaît.

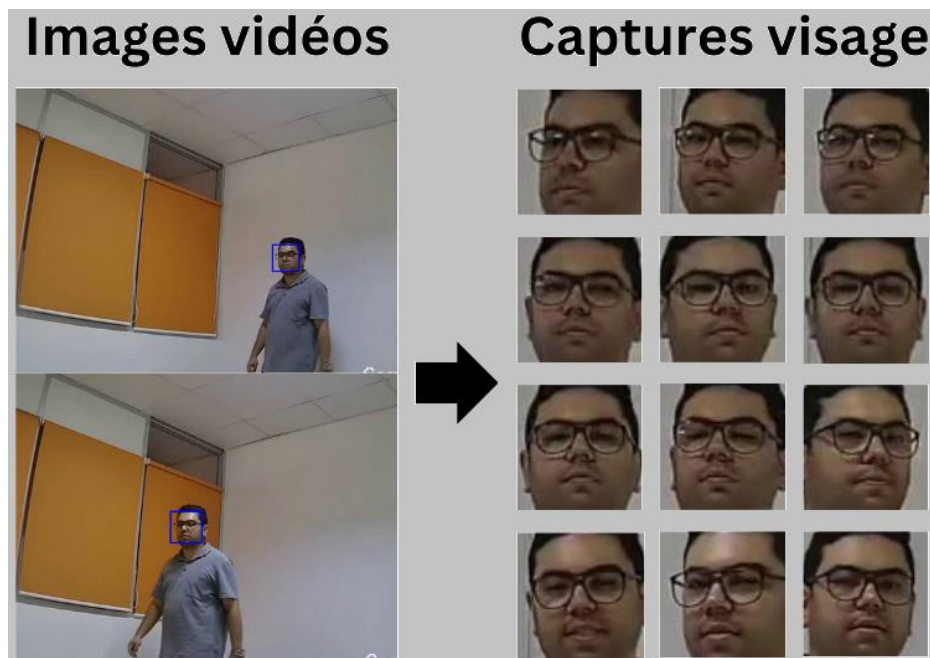


Figure 29 : Détection et extractions de visages à partir d'images vidéo

3.2 La reconnaissance faciale

Dans cette section, nous allons présenter la réalisation de la composante de reconnaissance faciale. Nous commencerons par expliquer la méthode de création et d'entraînement du modèle siamois. Ensuite, nous détaillerons la fonction de vérification intégrée au système, qui utilise ce modèle pour vérifier l'identité de l'employé détecté.

3.2.1 Création du modèle siamois

Le modèle siamois nécessite des données spécifiques pour un entraînement efficace. Nous disposons déjà d'un grand nombre d'images faciales enregistrées précédemment, que nous allons répartir équitablement en images positives et images ancrées.

En plus de ces images, nous utiliserons un nombre élevé d'images négatives provenant de la base de données LFW (Labeled Faces in the Wild), une base de données publique contenant plus de 13 000 images de visages de personnes célèbres et inconnues, collectées dans des conditions variées. [15]

Cette combinaison de données permettra au modèle d'apprendre à distinguer efficacement entre les visages des employés et ceux des autres individus.



Figure 30 : Extrait d'images faciales de la base de données LFW

Pour garantir une entrée cohérente et optimisée pour le modèle, il est crucial d'effectuer un prétraitement des images qui comprend le redimensionnement pour uniformiser les tailles et la mise à l'échelle des valeurs des pixels entre 0 et 1.

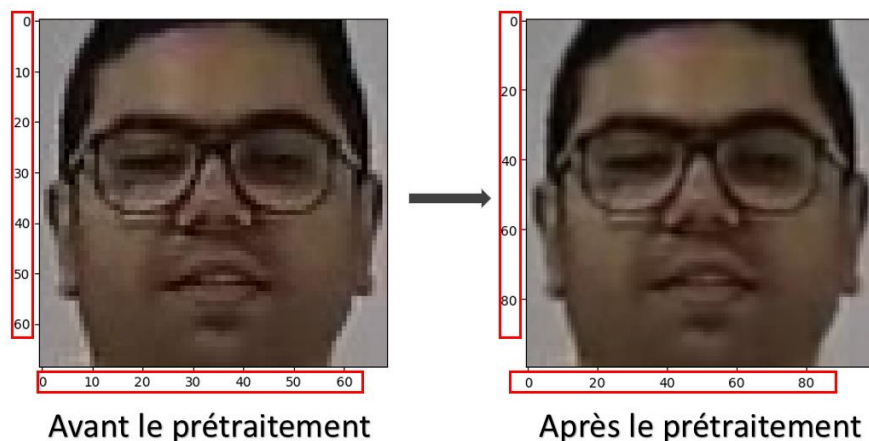


Figure 31 : Résultat de la fonction de prétraitement des images

Le modèle est composé de trois parties principales : le réseau d'intégration (Embedding Network), la partie distance et la partie classification.

La fonction « Embedding » définit un modèle de réseau de neurones convolutif pour créer des « embeddings » d'images, qui sont des représentations vectorielles de faible dimension utilisées pour la comparaison des images dans un modèle siamois. En définissant cette fonction, on obtient un modèle avec un réseau de neurones convolutif (CNN) qui prend des images de 100x100x3 en entrée et produit un vecteur d'embedding de taille 4096. Il est composé de plusieurs blocs de couches convolutives et de max-pooling, suivis d'une couche dense. Le modèle compte un total de 38,960,448 paramètres entraînables, comme le montre la figure ci-dessous.

```
embedding.summary()
✓ 0.0s
```

Model: "embedding"

Layer (type)	Output Shape	Param #
input_image (InputLayer)	[(None, 100, 100, 3)]	0
conv2d (Conv2D)	(None, 91, 91, 64)	19264
max_pooling2d (MaxPooling2D)	(None, 46, 46, 64)	0
conv2d_1 (Conv2D)	(None, 40, 40, 128)	401536
max_pooling2d_1 (MaxPooling2D)	(None, 20, 20, 128)	0
conv2d_2 (Conv2D)	(None, 17, 17, 128)	262272
max_pooling2d_2 (MaxPooling2D)	(None, 9, 9, 128)	0
conv2d_3 (Conv2D)	(None, 6, 6, 256)	524544
flatten (Flatten)	(None, 9216)	0
dense (Dense)	(None, 4096)	37752832

=====
Total params: 38,960,448
Trainable params: 38,960,448
Non-trainable params: 0

Figure 32 : Interprétation de la couche « embedding »

Ensuite, nous passons à la partie distance du modèle qui calcule la distance entre les deux embeddings (l'embedding de l'image d'entrée et l'embedding de l'image de validation). Cette partie utilise une couche L1Dist personnalisée qui calcule la distance de Manhattan (également appelée distance L1) entre les deux vecteurs d'embedding.

Enfin, la dernière partie du modèle est la classification qui prend en entrée la distance calculée et prédit si les deux images sont similaires ou non. Cette partie est constituée d'une couche Dense avec une seule sortie et une activation sigmoïde qui prédit la similarité entre les deux images.

Une fois que nous avons défini la couche L1Dist et la couche Dense, on obtient un modèle qui est constitué de deux entrées d'images de 100x100x3, traitées par un sous-modèle d'embedding partageant les mêmes poids. Les embeddings produits sont ensuite comparés par une couche L1Dist qui calcule la distance L1 entre eux. Enfin, une couche dense produit la sortie finale du réseau, indiquant la similarité entre les deux images.

```
siamese_network.summary()
```

Model: "SiameseNetwork"			
Layer (type)	Output Shape	Param #	Connected to
input_img (InputLayer)	(None, 100, 100, 3)	0	input_img[0][0]
validation_img (InputLayer)	(None, 100, 100, 3)	0	validation_img[0][0]
embedding (Functional)	(None, 4096)	38960448	input_img[0][0], validation_img[0][0]
l1_dist_1 (L1Dist)	(None, 4096)	0	embedding[0][0], embedding[1][0]
dense_1 (Dense)	(None, 1)	4097	l1_dist_1[0][0]
Total params: 38,964,545			
Trainable params: 38,964,545			
Non-trainable params: 0			

Figure 33 : Interprétation du réseau siamois

Après avoir mis en place les prérequis nécessaires, nous définissons une fonction d'entraînement et spécifions un nombre d'époques. Une époque représente un cycle complet à travers tout le jeu de données d'entraînement. [16]

La figure ci-dessous illustre le processus d'entraînement avec le nombre d'époque fixé.


```
Epoch 1/40
Tensor("binary_crossentropy/weighted_loss/value:0", shape=(), dtype=float32)
Tensor("binary_crossentropy/weighted_loss/value:0", shape=(), dtype=float32)
43/44 [=====>.] - ETA: 21sTensor("binary_crossentropy/weighted_loss/value:0", shape=(), dtype=float32)
44/44 [=====] - 937s 21s/step

Epoch 2/40
44/44 [=====] - 928s 21s/step

Epoch 3/40
44/44 [=====] - 928s 21s/step

Epoch 4/40
44/44 [=====] - 926s 21s/step

Epoch 5/40
44/44 [=====] - 925s 21s/step

Epoch 6/40
44/44 [=====] - 930s 21s/step

Epoch 7/40
44/44 [=====] - 928s 21s/step
...
44/44 [=====] - 857s 19s/step

Epoch 40/40
44/44 [=====] - 857s 19s/step
```

Figure 34 : Processus d'entraînement du modèle siamois

À l'issue de l'entraînement du modèle, nous pouvons à présent l'évaluer. Les trois principales métriques sont le rappel (recall), la précision et le F1-score.

```
# 1. Instantiate Precision and Recall metrics
precision = tf.keras.metrics.Precision()
recall = tf.keras.metrics.Recall()

# 2. Update their states with your data
precision.update_state(y_true, y_hat)
recall.update_state(y_true, y_hat)

# 3. Calculate Precision, Recall, and then F1-score
precision_result = precision.result().numpy()
recall_result = recall.result().numpy()

# Calculate F1-score
f1_score = 2 * (precision_result * recall_result) / (precision_result + recall_result)

print(f"precision_result: {precision_result}")
print(f"recall_result: {recall_result}")
print(f"F1-score: {f1_score}")

✓ 0.0s

precision_result: 0.75
recall_result: 0.78
F1-score: 0.764
```

Figure 27 : Calcul des métriques d'évaluation

Avec des valeurs de précision et de rappel de 0.75 et 0.78, le modèle présente une capacité raisonnable à éviter les prédictions erronées tout en identifiant une majorité des instances positives. Cela suggère une performance équilibrée où le modèle est relativement précis

dans ses prédictions positives tout en réussissant à capturer une proportion significative des cas positifs réels.

3.2.1 La reconnaissance faciale dans le cadre du système

La fonction de vérification est essentielle dans notre système pour pouvoir effectuer la tâche de la reconnaissance faciale. Elle prend en entrée le modèle siamois, ainsi que des seuils de détection et de vérification.

Les images de vérification utilisées sont issues des données collectées, tel que décrit dans la section de la détection de visages.

Cette fonction compare les embeddings faciaux extraits des images de vérification avec ceux des employés déjà enregistrés. Si la similarité entre les embeddings dépasse le seuil de vérification spécifié, l'identité de l'employé est confirmée

```
1/1 [=====] - 0s 344ms/step
1/1 [=====] - 0s 324ms/step
1/1 [=====] - 0s 299ms/step
1/1 [=====] - 0s 279ms/step
1/1 [=====] - 0s 271ms/step
1/1 [=====] - 0s 311ms/step
1/1 [=====] - 0s 238ms/step
1/1 [=====] - 0s 236ms/step
1/1 [=====] - 0s 252ms/step
1/1 [=====] - 0s 240ms/step
1/1 [=====] - 0s 247ms/step
1/1 [=====] - 0s 241ms/step
1/1 [=====] - 0s 246ms/step
1/1 [=====] - 0s 249ms/step
1/1 [=====] - 0s 236ms/step
1/1 [=====] - 0s 245ms/step
1/1 [=====] - 0s 219ms/step
1/1 [=====] - 0s 228ms/step
1/1 [=====] - 0s 251ms/step
1/1 [=====] - 0s 246ms/step
1/1 [=====] - 0s 245ms/step
1/1 [=====] - 0s 258ms/step
1/1 [=====] - 0s 233ms/step
1/1 [=====] - 0s 256ms/step
...
1/1 [=====] - 0s 235ms/step
Employé détecté: Adam Lassoued
```

Figure 35 : Exécution de la fonction de vérification

3.3 Le contrôle

Une fois qu'un employé est détecté et reconnu, la composante de contrôle vérifie si cet employé est autorisé à accéder à la zone surveillée par la caméra source.

En cas de présence non autorisée, la composante de contrôle du système doit archiver les données relatives à l'infraction sous plusieurs formes. Tout d'abord, elle enregistre une image du cadre entier où l'employé est apparu pour une documentation visuelle précise. Ensuite, elle met à jour le fichier journal qui enregistre les détails de chaque infraction détectée, assurant ainsi une traçabilité complète des événements. De plus, elle actualise la table de la base de données qui compte le nombre d'apparitions de chaque employé dans les zones non autorisées.

Nous commencerons par explorer l'architecture de la base de données MySQL qui se compose principalement de trois tables essentielles :

- une table qui associe chaque caméra à une zone spécifique
- une table qui détaille les zones autorisées et non autorisées pour chaque employé, les zones autorisées étant représentées par des « 1 » et les zones non autorisées étant représentées par des « 0 »
- une table qui enregistre le nombre d'apparitions d'un employé dans chaque zone (les zones autorisées étant toujours à 0)

address	zone
rtsp://{username1}:{password1}@adresse_ip_1/stream	etage_technique
rtsp://{username2}:{password2}@adresse_ip_2/stream	etage_administratif
rtsp://{username3}:{password3}@adresse_ip_3/stream	rh_et_finance
rtsp://{username4}:{password4}@adresse_ip_4/stream	etage_technique
rtsp://{username5}:{password5}@adresse_ip_5/stream	rh et finance

Table des caméras et des zones associées

employe	etage_technique	etage_administratif	rh_et_finance
Adam Lassoued	0	1	1
Ahmed Rebai	1	0	0
Samar Mezzi	0	1	1
Sofiane Menzli	1	0	0

Table des autorisations des employés par zone

employe	detect_non_auto	etage_technique	etage_administratif	rh_et_finance
Adam Lassoued	5	5	0	0
Ahmed Rebai	5	0	3	2
Samar Mezzi	3	3	0	0
Sofiane Menzli	6	0	4	2

Table de comptabilisation d'apparitions dans chaque zone

Figure 36 : Aperçu des tables de la base de données

La tâche de vérification est assurée par deux fonctions étroitement liées : la première vérifie dès le démarrage du système la zone attribuée à la caméra source, tandis que la deuxième fonction vérifie si l'employé est autorisé à cette même zone une fois détecté et reconnu.

Une fois qu'une personne est identifiée et qu'il est vérifié qu'elle n'est pas autorisée à apparaître dans la zone de la caméra qui l'a détecté, le système doit enregistrer les détails de l'évènement d'infraction sous ses différentes formes de données.

```
if verified:

    if check_autorisation(person_name, zone)==False:

        cursor = mydb.cursor()
        sql = f"""
        UPDATE dashboard
        SET detect_non_auto = detect_non_auto + 1
        WHERE personne = %s
        """
        cursor.execute(sql, (person_name,))
        mydb.commit()
        # Mettre à jour le nombre de détections non autorisées

        cursor = mydb.cursor()
        sql = f"""
        UPDATE dashboard
        SET {zone} = {zone} + 1
        WHERE personne = %s
        """
        cursor.execute(sql, (person_name,))
        mydb.commit()
        # Mettre à jour le nombre d'apparitions dans la zone

        now = time.strftime("%Y-%m-%d_%H-%M-%S")
        input_frame = frame
        filename_frame = os.path.join('application_data', 'input_frame', f"frame_{cap.get(cv2.CAP_PROP_BACKEND)}-{now}.jpg")
        cv2.imwrite(filename_frame, input_frame)
        # Enregistrer l'image du cadre entier de l'infraction

        with open('verification_log.txt', 'a') as log_file:
            modelname = os.path.splitext(os.path.basename(model_path))[0]
            log_file.write(f"Person: {person_name}, Date and Time: {now}, Image: {filename_frame}, Zone: {zone}\n")
        # Mettre à jour le fichier journal avec les détails de l'infraction
```

Figure 37 : Opérations d'archivage des données

3.4 L'interface de visualisation des données

Dans cette dernière section, nous allons aborder l'interface de visualisation des données archivées. Étant donné le caractère critique de ces données, leur accès est strictement réservé à un responsable de sécurité, muni d'un identifiant unique et d'un mot de passe. Ainsi, la première action à entreprendre pour accéder aux données consiste à s'authentifier.

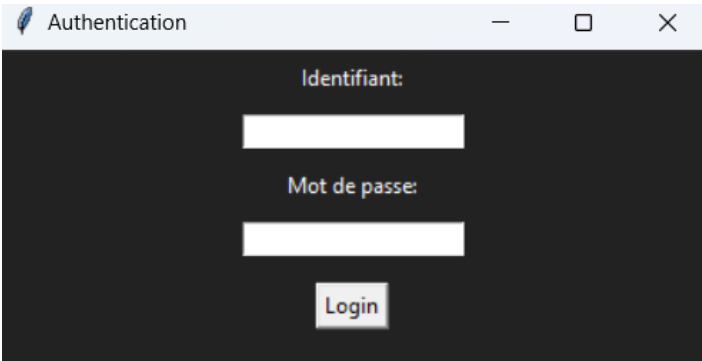


Figure 38 : Fenêtre d’authentification

Une fois authentifié, l'utilisateur accède à l'interface principale du système. Cette interface est structurée en deux cadres distincts :

- Le cadre gauche : Il est dédié au « tracking général » : il présente une série d'images identifiées, horodatées et localisées des infractions récentes, organisées dans un ordre chronologique décroissant.
- Le cadre droit : Il affiche la date du jour, le nombre d'infractions enregistrées pour la journée ainsi que la liste des employés ayant commis des infractions, également classées par ordre chronologique décroissant.

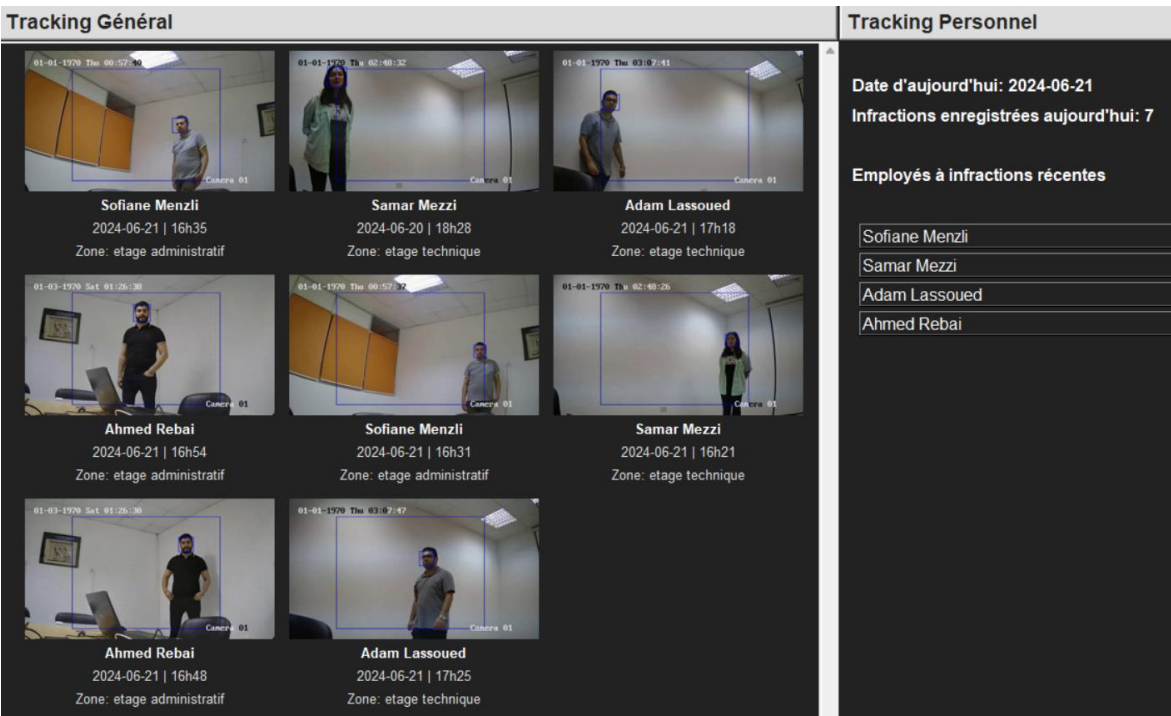


Figure 39 : Vue globale de l’interface de visualisation des données

Le champ vierge situé sous la liste des employés est dédié à afficher des détails spécifiques sur un employé sélectionné. En effet, lorsque l'utilisateur sélectionne un employé, trois éléments apparaissent :

- une mini carte montrant les zones autorisées et interdites pour cet employé
- un tableau indiquant le nombre de fois où l'employé est apparu dans des zones non autorisées
- un bouton permettant de visualiser l'historique personnel des infractions en images.

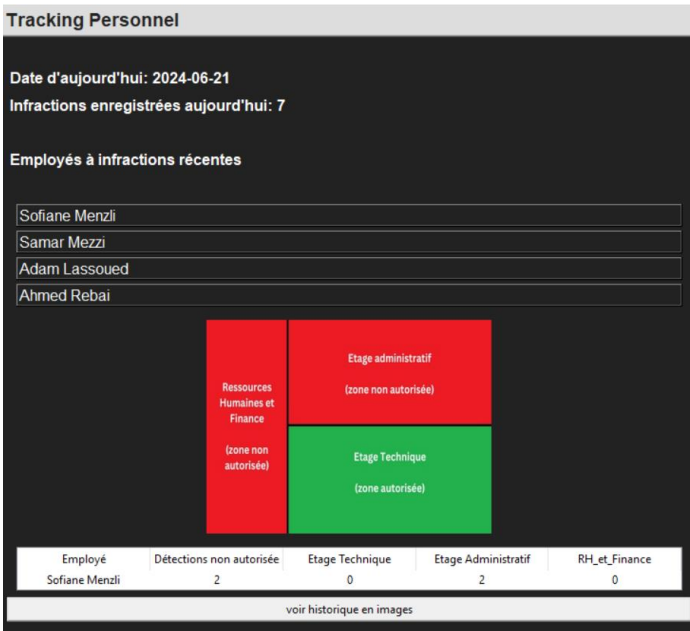


Figure 40 : Exemple 1 d'un employé sélectionné

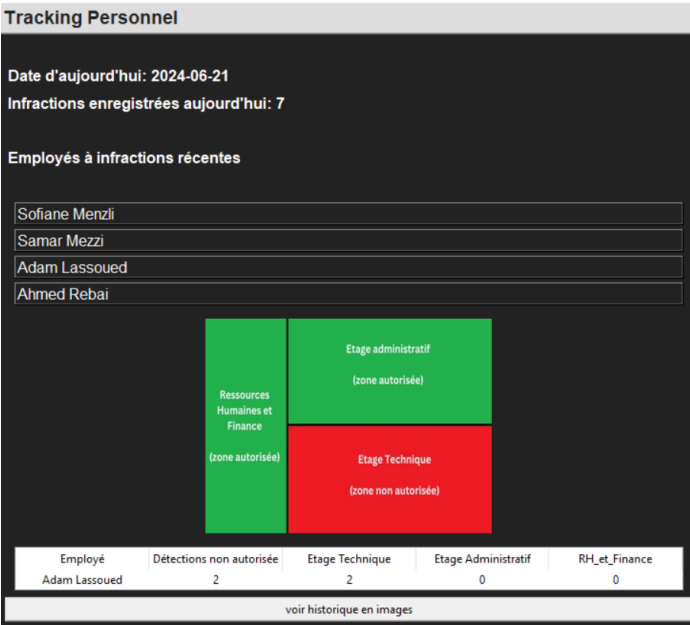


Figure 41 : Exemple 2 d'un employé sélectionné

Pour consulter l'historique personnel complet de la personne sélectionnée, comprenant des images et un résumé du nombre total et journalier des infractions, il suffit de cliquer sur le bouton "Voir historique en images". Cela ouvrira un pop-up contenant toutes ces informations.

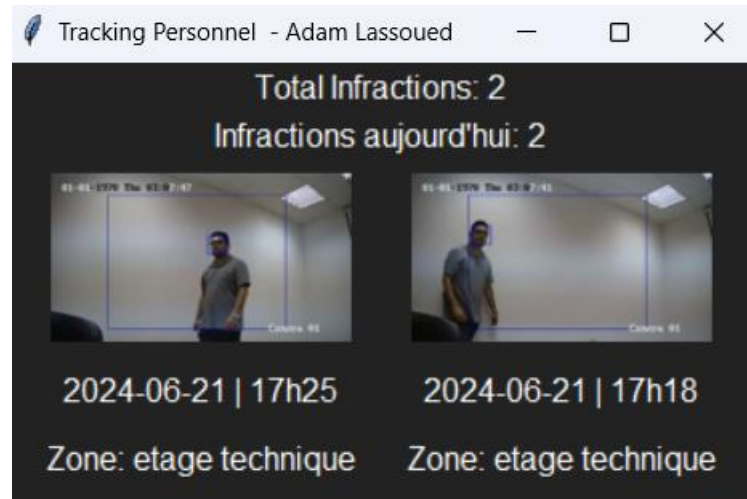


Figure 42 : Fenêtre pop-up d'un historique personnel

5 Conclusion

En conclusion, ce projet a abouti à la réalisation d'un système complet et d'une interface graphique intuitive, grâce à une planification rigoureuse et une mise en œuvre itérative par sprints. L'utilisation d'un environnement logiciel et matériel bien choisi a été déterminante pour assurer la stabilité et l'efficacité du développement. Enfin, les principes de design centrés sur l'utilisateur ont permis de créer une expérience utilisateur optimale et répondant aux attentes initiales.

Conclusion générale

Dans le cadre de ce projet de fin d'études, nous avons travaillé à améliorer le niveau de sécurité au sein de l'entreprise d'accueil, Opalia Pharma. Nous avons tiré parti de l'infrastructure de sécurité existante tout en créant un système capable de surveiller les déplacements des employés à l'intérieur de l'établissement et d'archiver toute infraction. Ces opérations de contrôle ont été mises en œuvre grâce à des techniques de deeplearning, incluant la détection d'objets et la reconnaissance faciale.

Au cours de ce stage, l'objectif était de concevoir et de mettre en œuvre une solution composée de deux grandes parties. La première partie, un système automatisé, est un script capable de lire les images vidéo en direct provenant des caméras, d'effectuer la détection et la reconnaissance faciale, puis de vérifier les autorisations des individus identifiés pour accéder à la zone surveillée par la caméra. La seconde partie est destinée à un acteur humain, notamment le responsable de la sécurité, qui peut, grâce à une interface simple et intuitive, visualiser les données archivées concernant toute infraction détectée et enregistrée dans la base de données.

Tout au long de cette expérience, l'implication de plusieurs employés a été cruciale. Ils ont consacré de leur temps et donné leur consentement pour m'aider à collecter un jeu de données destiné à l'entraînement du modèle de reconnaissance, ainsi qu'à participer à des simulations en temps réel et à fournir des vidéos pour des simulations ultérieures.

Ainsi, nous avons développé un système capable de détecter les employés et de reconnaître un certain nombre d'entre eux, tout en contrôlant leurs déplacements. Cela a permis au responsable de la sécurité d'accéder à un historique clair, de surveiller toute activité suspecte et de prendre les décisions nécessaires.

À cette étape, l'objectif suivant serait donc d'impliquer la totalité des employés en collectant leurs données en images afin d'entraîner le modèle de reconnaissance à reconnaître toutes les personnes détectables sur les images vidéos. Une autre optimisation serait de mettre en place, en plus de l'historisation et de la visualisation postérieure, un système d'alerte en temps réel. Ce système servirait à signaler un événement suspect au moment précis où il a eu lieu. Une dernière avancée consisterait à développer une application mobile pour l'interface de visualisation des données. Cela impliquerait de passer d'une application locale lisant les données archivées à partir d'un serveur local à une version web avec une base de données en ligne, accessible via l'application mobile.

Conclusion générale

Pour conclure, cette expérience professionnelle a été extrêmement enrichissante. Elle nous a permis, d'une part, de mettre en pratique les notions théoriques acquises durant notre cursus, et d'autre part, de développer nos compétences en matière de technologies de détection d'objets et de reconnaissance faciale. Nous avons également eu l'opportunité de nous initier à la vie professionnelle au sein d'une grande entreprise, où nous avons pu nouer des relations avec d'autres stagiaires et employés, et partager une éthique de travail précieuse. Enfin, nous souhaitons exprimer notre entière satisfaction d'avoir travaillé dans d'excellentes conditions matérielles et dans un environnement agréable.

Références

Bibliographie

[9] <https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf>

Gregory Koch , Richard Zemel et Ruslan Salakhutdinov (2015), Siamese Neural Networks for One-shot Image Recognition, ICML deep learning workshop, vol. 2.

Netographie

[1] <https://www.opaliarecordati.com/fr/>

[2] <https://blog.swann.com/dvr-vs-nvr-whats-the-difference/>

[3] <https://us.dahuasecurity.com/product/smartpss/>

[4] <https://litum.com/>

[5] <https://litum.com/staff-workflow-employee-tracking-rtls/>

[6] <https://www.sooyoos.com/publication/scrum/>

[7] <https://herbertograca.com/2017/08/03/layered-architecture/>

[8] <https://pyimagesearch.com/2020/11/30/siamese-networks-with-keras-tensorflow-and-deep-learning/>

[10] https://docs.opencv.org/4.x/db/d28/tutorial_cascade_classifier.html

[11] https://blog.vaniila.ai/Reconnaissance_faciale/

[12] <https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels>

[13] https://github.com/rohanrao619/Face_Recognition_using_Siamese_Network

[14] <https://www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/>

[15] <https://vis-www.cs.umass.edu/lfw/>

[16] <https://medium.com/@rinkinag24/a-comprehensive-guide-to-siamese-neural-networks-3358658c0513>

Table des matières

Introduction générale	1
Chapitre 1	3
Cadre général du projet	3
1 Introduction	3
2 Présentation de l'organisme d'accueil.....	3
2.1 Présentation générale	3
2.2 Secteurs d'activité	4
2.3 Infrastructure de l'entreprise et de l'installation industrielle	4
2.3.1 Infrastructure physique.....	4
2.3.2 Infrastructure de la sécurité	5
2.4 Organigramme	7
3 Présentation du projet.....	8
3.1 Problématique et objectifs du projet.....	8
3.2 Travail demandé.....	8
3.3 Solution proposée	9
3.4 Solution similaire.....	10
3.5 Etude et critique de l'opération classique	12
4 Conclusion.....	13
Chapitre 2	15
Analyse et spécification des besoins	15
1 Introduction	15
2 Identification des acteurs.....	15
3 Spécifications des besoins	15
3.1 Besoin fonctionnels.....	15
3.2 Besoin non-fonctionnels	16
Les besoins non-fonctionnels comprennent :	16
3.3 Diagramme de cas d'utilisation global	17
4 Pilotage du projet avec SCRUM	18
4.1 Répartition des rôles	18
4.2 Planification et plan des sprints	18
4.3 Product Backlog.....	19
5 Conclusion.....	21

Chapitre 3	22
Conception	22
1 Introduction	22
2 Architecture	22
3 Conception détaillée	23
3.1 Sprint 1 : Intégration de la détection des visages dans le système	23
3.1.1 Backlog du Sprint 1	23
3.1.2 Diagramme de séquence du Sprint 1	24
3.2 Sprint 2 : Intégration de la reconnaissance faciale dans le système	24
3.2.1 Backlog du Sprint 2	24
3.2.2 Diagramme de séquence du Sprint 2	25
3.3 Sprint 3 : Implémentation de la partie contrôle dans le système	25
3.3.1 Backlog du Sprint 3	25
3.3.2 Diagramme de séquence du sprint 3	26
3.3.3 Synthèse du fonctionnement global du système : Diagramme d'activités	26
3.4 Sprint 4 : Développement de l'interface graphique de visualisation	28
3.4.1 Backlog du Sprint 4	28
3.4.2 Diagramme de séquence du Sprint 4	28
4 Conclusion	29
Chapitre 4	30
Étude des modèles de l'intelligence artificielle adoptés	30
1 Introduction	30
2 Modèle de détection de visages	30
2.1 Définition de la détection d'objets	30
2.2 Le modèle déployé : Haar Cascade Frontal Face	30
2.2.1 Le fonctionnement du modèle	31
2.2.2 Avantages et limites	32
3 Modèle de reconnaissance faciale	32
3.1 Définition de la reconnaissance faciale	32
3.2 Le modèle développé : Le modèle siamois	33
3.2.1 Introduction aux réseaux de neurones siamois	33
3.2.2 Principes des réseaux de neurones siamois	33
3.2.3 Architecture du réseau	34
3.2.4 Données d'entraînement	34
3.2.5 Processus d'entraînement	35

3.2.6 Configuration des entrées.....	35
3.2.7 Tests et évaluation.....	35
4 Conclusion.....	36
Chapitre 5.....	37
Réalisation.....	37
1 Introduction.....	37
2 Environnement de travail.....	37
2.1 Environnement matériel.....	37
2.2 Environnement logiciel.....	37
2.2.1 Environnement de développement.....	37
2.2.2 Langages et bibliothèques.....	38
2.2.3 Base de données.....	40
3 Mise en œuvre.....	40
3.1 La détection de visages.....	40
3.2 La reconnaissance faciale.....	41
3.2.1 Création du modèle siamois.....	42
3.2.1 La reconnaissance faciale dans le cadre du système.....	46
3.3 Le contrôle.....	47
3.4 L'interface de visualisation des données.....	48
5 Conclusion.....	51
Conclusion générale.....	52
Références.....	54

Résumé

Le présent document constitue une synthèse du travail réalisé dans le cadre du projet de fin d'études effectué au sein de l'entreprise Opalia Pharma pour l'obtention du diplôme de licence en génie logiciel et développement des systèmes d'information à l'Université Internationale de Tunis. L'objectif de ce projet est de concevoir et mettre en œuvre un système autonome capable de contrôler le déplacement des employés au sein des zones opérationnelles et administratives de l'établissement. Le travail a été réalisé grâce au déploiement de technologies axées sur le deep learning pour la détection d'objets et la reconnaissance faciale. Enfin, une application bureau permet de visualiser une archive en images et en données textuelles et statistiques de toutes les infractions détectées.

Mots clés : surveillance intelligente, suivi en temps réel, analyse images vidéos, deep learning, détection d'objets, haar cascade, reconnaissance faciale, TensorFlow, réseau de neurones siamois, interface Tkinter

Abstract

This document presents a synthesis of the work carried out as part of the end-of-studies project at Opalia Pharma, aimed at obtaining a Bachelor's degree in Software Engineering and Information Systems Development from the International University of Tunis. The objective of this project is to design and implement an autonomous system capable of monitoring employees' movements within operational and administrative areas of the establishment. The project leverages deep learning technologies for object detection and facial recognition. Additionally, a desktop application is developed to visualize an archive of images and textual/statistical data of all detected infractions.

Keywords: smart surveillance, real-time monitoring, video image analysis, deep learning, object detection, Haar cascade, facial recognition, TensorFlow, Siamese neural network, Tkinter interface