

EPIGRAPHE

« Les caméras de surveillance ne vous quitteront pas des yeux. Elle sauront vous dissuader de piquer un petit somme ou de voler même une petite somme, de vous moucher dans le pain du client ou de vous mettre les doigts dans le nez. »

Anna Sam

DEDICACE

A vous chères parents, Richard KADIMBA et Hélène MBOMBO, à qui je dois tout ;

A mes chers frères et sœurs que j'aime tant.

REMERCIEMENTS

Je remercie mon Dieu, avant tout ; par son grand amour il ne cesse de renouveler mon souffle de vie et de déverser ses grâces dans ma vie tous les matins. Que gloire lui soit rendue.

A mes très chers parents, qui par leurs amour, leurs soutien tant morale, matériel que spirituel, m'ont encadré, élevé et instruit ; Je ne saurai leurs exprimer l'expression de ma gratitude en ces quelques lignes car ma vie entière retrace leur bonté et leurs amour à mon égard. Chers Richard KADIMBA et Hélène MBOMBO, veuillez recevoir mes remerciements distingués.

Je ne peux me passer de remercier ma chère école, l'Ecole Supérieure d'Informatique Salama pour son cadre et son corps professoral qui m'a formé des années durant, et je remercie particulièrement monsieur Patrick KASONGA, le secrétaire général académique, pour son aide et ses conseils durant mon parcours académique.

Pour ses conseils, sa disponibilité, son sens de l'écoute ainsi que les connaissances qu'il m'a transmis, je remercie mon directeur, monsieur Patient KASONGO; Sans n'oublie mon co-directeur, l'ingénieur Eddysha JABULANI à qui j'adresse mes sincères remerciements.

Je remercie également toutes les personnes qui m'ont aidé durant tout mon parcours académique.

KONGOLO KADIMBA Jean-Marc

LISTE DES FIGURES

Figure 1.1. Vidéosurveillance sur réseau ip.....	8
Figure 1.2. Kit de vidéosurveillance	9
Figure 1.3. La vidéosurveillance analogique	9
Figure 1.4. Méthode de déformation parallèle par David Beymer and Tomaso Poggio. 13	
Figure 1.5. KPPV	14
Figure 1.6. Réseau de neurones	14
Figure 1.7. Détection des visages avec l'algorithme de Viola et Jones.....	15
Figure 1.8. La sélection des points caractéristiques avec l'algorithme d'EBGM.....	15
Figure 1.9. Création d'un treillis avec l'algorithme d'EBGM.....	15
Figure 3.1. Modèle de conception basé sur les cas d'utilisations	27
Figure 3.2. Diagramme des cas d'utilisations système	30
Figure 3.3. Premier diagramme d'activité	36
Figure 3.4. Deuxième diagramme d'activité	37
Figure 4.1. Interface de connexion	39
Figure 4.2. Interface d'activation/désactivation de l'alarme	39
Figure 4.3. Interface d'activation/désactivation de la détection des mouvements	40
Figure 4.4. Interface d'ajout des admin	40
Figure 4.5. Interface de visualisation des caméras à distance	41
Figure 4.6. Interface d'accueil	42
Figure 4.7. Fenêtre principale de visualisation	42
Figure 4.8. Choix de la caméra à visualiser, activation/désactivation de l'alarme et la détection des mouvements	43
Figure 4.9. Reconnaissance faciale.....	44
Figure 4.10. Indicateur des mouvements	44
Figure 4.11. Détection des visages et des yeux	45
Figure 4.12. Envoyer d'un flux vidéo.....	46
Figure 4.13. Envoi des données Android à PHP.....	47

LISTE DES TABLEAUX

Tableau 3.1. Structuration des cas d'utilisations	35
---	----

LISTE DES ACRONYMES

UP : Unified Process

3D : Trois Dimension

API: Application Programming Interface

OPEN-CV: Open Source Computer Vision Library

SMS: Short Message Service

GSM: Global System for Mobile Communication

IP: Internet Protocol

CCTV: Closed Circuit of TV

PCA: Principal Component Analysis

LDA: Linear Discriminant Analysis

KPPV: K Plus Proches Voisins

EBGM : Elastic Bunch Graph Matching

BSD: Berkeley Software Distribution

OCR: Optical Character Recognition

UML: Unified Modeling Language

RUP: Rational Unified Process

XP: eXtreme Programming

2TUP: Two Trasks Unified Process

IDE: Integrated Development Environment

SGBD: Système de Gestion des bases de données

TABLE DES MATIERES

EPIGRAPHE	I
DEDICACE	II
REMERCIEMENTS.....	III
LISTE DES FIGURES	IV
LISTE DES TABLEAUX	V
LISTE DES ACRONYMES.....	VI
TABLE DES MATIERES	VII
AVANT-PROPOS	X
0. INTRODUCTION GENERALE.....	1
0.1. Problématique et hypothèse	1
0.1.1. <i>Problématique</i>	1
0.1.2. <i>Hypothèse</i>	2
0.2. Choix et intérêt du sujet	2
0.3. Méthodes et techniques	3
0.3.1. <i>Méthodes</i>	3
0.3.2. <i>Techniques</i>	3
0.4. Etat de l'art.....	3
0.4.1. <i>Reconnaissance faciale</i>	3
0.4.2. <i>Détection de mouvement</i>	4
0.4.3. <i>Surveillance</i>	4
0.5. Délimitation du travail	5
0.6. Subdivision du travail	5
0.7. Outils logiciels et équipements utilisés.....	5
Chapitre 1 : CONSIDERATIONS THEORIQUES.....	7
1.1. Introduction	7
1.2. La vidéosurveillance	7
1.2.1. <i>Historique</i>	7
1.2.2. <i>Les différents systèmes de vidéosurveillance</i>	8
1.2.2.1. La vidéosurveillance sur réseaux IP	8
1.2.2.2. Les kits de vidéosurveillance.....	9
1.2.2.3. La vidéosurveillance analogique	9

1.3.	La reconnaissance faciale.....	10
1.3.1.	<i>La détection des visages</i>	10
1.3.1.1.	Approches basées sur les connaissances acquises (Knowledge-based methods)	11
1.3.1.2.	Approches basées sur le « Template-matching »	11
1.3.1.3.	Approches basées sur l'apparence (Appearance-based methods)	11
1.3.1.4.	Approches basées sur les caractéristiques invariantes (Feature invariant approaches).....	12
1.3.2.	<i>L'extraction des caractéristiques</i>	12
1.3.3.	<i>La reconnaissance de visage</i>	12
1.4.	La bibliothèque Open CV	16
Chapitre 2 : ETUDE PREALABLE		17
2.1.	Introduction	17
2.2.	Présentation de l'ESIS.....	17
2.2.1.	<i>Historique</i>	17
2.2.2.	<i>Vision</i>	18
2.2.3.	<i>Mission</i>	18
2.2.4.	<i>Organigramme</i>	19
2.3.	Diagnostic de l'existant.....	20
2.4.	Solution proposée.....	22
Chapitre 3 : ANALYSE ET CONCEPTION DU SYSTEME INFORMATIQUE		24
3.1.	Introduction	24
3.2.	Présentation de la méthode UP	24
3.2.1.	<i>Processus Unifié : (Unified Process)</i>	26
3.2.1.1.	UP est piloté par les cas d'utilisation	26
3.2.1.2.	UP est centré sur l'architecture	28
3.2.1.3.	UP est itératif et incrémental	28
3.3.	Expression des besoins.....	29
3.3.1.	<i>Formalisation des besoins</i>	30
3.3.1.1.	Diagramme des cas d'utilisations.....	30
3.3.1.2.	Description textuelle.....	31
3.3.1.3.	Structuration des cas d'utilisations	35
3.3.1.4.	Diagramme d'activité	35
Chapitre 4 : IMPLEMENTATION		38

4.1.	Environnement logiciel	38
4.2.	Environnement matériel	38
4.3.	Présentation de l'application	39
4.3.1.	<i>Quelques captures d'écran de l'application Android.....</i>	39
4.3.2.	<i>Quelques captures d'écran de l'application Web.....</i>	42
4.3.3.	<i>Capture de l'illustration de la reconnaissance faciale et de la détection des mouvements</i>	44
4.3.4.	<i>Capture de quelques fonctions utilisées.....</i>	45
CONCLUSION GENERALE.....		48
BIBLIOGRAPHIE.....		49

AVANT-PROPOS

Grâce à l'évolution de la technologie, plusieurs problèmes courant de l'homme ont trouvés satisfactions ; Nous ne saurons énoncer ces différents problèmes résolus à cause de leurs multitudes, mais l'un de ces derniers a attiré particulièrement notre attention, il s'agit de la vidéosurveillance.

En 2016, il existait plus de 250 millions de caméra dans le monde, est l'une des villes ayant le plus grand nombre de caméra en activité jusqu'à ce jour est Londres, et cela a permis non seulement de diminuer le têt de crime ou d'incident non élucidé, mais aussi d'accroître le rendu de certaines entreprises.

Ce présent travail, qui est la conception d'un système de vidéosurveillance intégrant la reconnaissance faciale et la détection des mouvements, a pour but de permettre également de diminuer les cas de vol, crimes et incident non élucidé aux seins de différentes organisations congolaises et plus particulièrement des labos d'ESIS.

0. INTRODUCTION GENERALE

Dans le pays qui est le nôtre, à l'issue du premier cycle d'étude supérieur et universitaire, tout étudiant doit élaborer un travail de fin de cycle cadrant avec son domaine d'étude. C'est ainsi que nous nous sommes donné pour tâche la réalisation d'un système de vidéo surveillance intégrant la reconnaissance faciale et la détection des mouvements.

0.1.Problématique et hypothèse

0.1.1. Problématique

En République Démocratique du Congo (RDC) tout comme sous d'autre cieux, les cas de vol sont très fréquents, et pour y remédier, plusieurs entreprises font recours à des vigiles afin de contrôler le matériel voire le personnel ; les particuliers ne laissent presque jamais leur maisons vides, de peur d'être victime d'un vol ;

La majorité d'organisation et infrastructures sont butés à ce problème, « la surveillance » ; Certaines entreprises occidentales ont mis sur le marché des logiciels de surveillance afin d'apporter solution à ce problème mais ces derniers sont couteux et répondent plus aux besoins de leur société qu'à ceux des africains, plus précisément des congolais ; Ces dites solutions couteuse sont aussi très complexe à implémenter et demande presque toujours la présence d'un ingénieur formé dans le domaine et dont les services sont aussi couteux que la solution elle-même.

En ce qui concerne les labos de l'ESIS, plus particulièrement le labo génie logiciel, ce derniers est munis de divers matériels informatiques assez couteux, qui mérite d'être protégé et surveillé en permanence afin de ne pas être victime de vol, mauvaise utilisation ou destruction de matériel, destruction qui peut être révoqué ou imprévu.

Face à aux différentes difficultés et problèmes énoncés ci-haut, une question se pose, Que faire pour pallier aux cas de vol et/ou pertes, destruction de matériel dans le labo génie logiciel d'ESIS en particulier et en RDC en général ?

0.1.2. Hypothèse

Pour résoudre ce problème, nous allons concevoir un système de vidéo surveillance qui permettra de contrôler des pièces d'une maison, des bureaux, des écoles, universités etc. en détectant les intrusions et en envoyant des notifications en cas de besoin au responsable.

La détection des mouvements sera activer ou désactiver par reconnaissance faciale ou via une application Android. Cette dernière offrira en plus de cette option, la possibilité de contrôler les caméras à distance et de regarder les vidéos en temps réel.

0.2.Choix et intérêt du sujet

Les systèmes de vidéos surveillances ont fait leurs preuves dans plusieurs domaines aux quatre coins du monde ; dans des supers marchés et différents commerces, dans des entrepôts et espace de stockage, dans des laboratoires et bureaux,... .

La vidéo surveillance ne permet pas seulement d'assurer la sécurité des personnes et des biens, mais elle permet également d'accroître le rendement des responsables d'organisation en leurs permettant d'être à plusieurs endroit au même moment.

Associé à la reconnaissance faciale et la détection des mouvements, la vidéo surveillance monte d'un cran car elle permettra non seulement d'assurer la sécurité des personnes et des biens, d'accroître le rendement de travail ou encore de permettre à des vigiles de contrôler l'intégralité d'une installation, mais aussi de connaître avec précision les faits et gestes de toutes les personnes présentes dans des installations ainsi que de contrôler des zones interdites même en étant distrait ou pris par d'autre tâches.

De ces faits, nous avons jugé qu'il est d'une importance capital d'exploiter le domaine de la vidéo surveillance en l'associant à la reconnaissance faciale et la détection des mouvements.

0.3.Méthodes et techniques

0.3.1. Méthodes

Une méthode est définie comme étant : « un ensemble de démarches raisonnées, suivies pour parvenir à un but »¹.

Pour la réalisation de notre travail nous avons fait recours à la méthode UP qui est « un processus de développement logiciel itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques » [12, p.9].

0.3.2. Techniques

Une technique est un ensemble de procédés employés pour produire une œuvre ou obtenir un résultat déterminé.²

Pour la réalisation de notre travail, les techniques suivantes ont été utilisées :

L'interview : Grâce à cette technique, nous avons pu nous entretenir avec des professionnels dans le monde informatique, ainsi qu'avec les professeurs et le personnel de l'ESIS afin de mieux appréhender le sujet.

La documentation : La documentation nous a permis de rechercher tous les ouvrages relatifs à notre travail, et plus particulièrement des ouvrages sur la vision par ordinateur.

0.4.Etat de l'art

0.4.1. Reconnaissance faciale

90% des logiciels grands public intégrant la reconnaissance faciale se basent sur des photos statiques de hautes définitions et non sur des images en mouvements (Vidéo) ou des images ou vidéo 3D et ses derniers ne se limitent qu'à ça, la reconnaissance faciale. L'un des plus célèbres API de reconnaissance faciale est Kairos, et ce dernier est disponible dans plusieurs langage (C#, PHP, Python, C, C++, Java) mais sa plus grande faiblesse est que tout logiciel utilisant cette API doit être connecté à

¹ *Dictionnaire Le Robert Micro*, 2^{ème} Edition, Dictionnaires le Robert, Paris, 2006.

² *Dictionnaire Le Nouveau Petit Robert*, Dictionnaires le Robert, Paris, 2008.

internet afin de réaliser la reconnaissance faciale grâce à des applications annexes installées sur leur serveur. En dehors de Kairos, il existe plusieurs autres solutions mais la plus part d'entre elles sont basées que sur les photos statiques de hautes définitions.

Les solutions les plus complètes sont propriétaires, elles appartiennent à des grandes organisations et ne peuvent pas être utilisées par le grand public.

0.4.2. Détection de mouvement

En ce qui concerne la détection des mouvements, la majorité de solution sont basés sur des matériels (capteur infrarouge, capteur de présence, laser,...) qui reçoivent les informations concernant les mouvements et renvoient une certaine réponse. Et il existe aussi plusieurs API permettant de faire de la détection de mouvement à l'aide de tout type de caméra.

0.4.3. Surveillance

Les logiciels de surveillance existant sur le marché africain ou disponible sur internet intègrent la détection de mouvement et l'enregistrement des vidéos capturées, l'un des plus célèbre est EyeLine Vidéo Surveillance, il est gratuit et disponible sur plusieurs site web.

Ces derniers intègrent la détection de mouvement et non la reconnaissance faciale et ne sont pas légers, ils demandent des machines de bonnes performances afin que les vidéos soient fluides et que le logiciel tourne normalement.

Nous ne sommes pas les premiers à mettre en place un système de surveillance, lors de l'année académique 2013-2014, l'étudiant BAKANKENGESHA BAKENGE Augustin, de la filière Télécommunication et réseaux a eu à défendre un travail sur : « Etude de la mise en place d'un système de surveillance et de commande des équipements domestiques à distance via le réseau GSM ou internet ».

Notre travail n'a pas pour objectif la reproduction du travail ci-haut cité, mais la réalisation d'un système parallèle avec une tout autre vision.

0.5.Délimitation du travail

Le domaine de la vision par ordinateur étant vaste et en continuelle évolution, nous tenons à signaler que par manque de matériel adéquat, dans notre travail, la reconnaissance faciale ne passera que dans une pièce suffisamment éclairée et notre réseau de caméra sera composé de deux caméras uniquement pour une phase de test.

0.6.Subdivision du travail

Hormis l'introduction et la conclusion, notre travail comprendra quatre grands chapitres qui sont :

- Chapitre premier : **Considération théorique**, dans ce chapitre, nous parlerons en long et en large de tous les termes ainsi que toutes les notions théoriques importantes pour une meilleure compréhension de ce travail ;
- Chapitre deuxième : **Etude préalable**, ce chapitre présentera notre cadre référentiel et nous analyserons quelques systèmes de vidéosurveillance existants;
- Chapitre troisième : **Conception du système**, ce chapitre se focalisera sur la conception du système à mettre en place ;
- Chapitre quatrième : **Implémentation**, Dans ce dernier chapitre, nous nous attarderons sur la réalisation et le déploiement du système.

0.7.Outils logiciels et équipements utilisés

Pour la réalisation de notre travail, nous avons utilisé le langage de programmation Python comme langage principal, et le PHP ainsi que le c# comme langages secondaires.

L'application utilisera la bibliothèque Open CV pour la reconnaissance faciale, en ce qui concerne la vidéo surveillance et la détection de mouvement, nous allons utiliser nos propres algorithmes.

L'application sera installée sur un serveur et ce dernier distribuera certains services comme l'accès aux vidéos de surveillances qui seront consommées par des clients PHP. Les vidéos seront prises par des caméras IP.

Les alertes SMS seront faites grâce à un modem GSM qui sera connecté au serveur.

Nous avons choisis le langage Python à cause de sa puissance de calcul et de sa légèreté ; La bibliothèque Open CV parce qu'elle contient tous les éléments nécessaires pour le "Computer Vision" (La vision assisté par ordinateur).

Le modem GSM est pratique par le fait qu'il va nous permettre d'envoyer des sms sans passer par internet.

Chapitre 1 : CONSIDERATIONS THEORIQUES

1.1.Introduction

Ce présent chapitre nous présentera les différents concepts à connaître afin de comprendre au mieux notre travail ; Lesdits concepts seront répartis en trois grandes branches ou catégories, la première sera celle relative à la vidéosurveillance, la deuxième sera liée aux notions de reconnaissances faciale et la dernière à la bibliothèque Open CV, qui est la plaque tournante de notre travail.

1.2.La vidéosurveillance

La vidéosurveillance est un système de surveillance des lieux publics par caméras vidéo³. La vidéosurveillance ne date pas de ce jour, elle a été utilisée pour la première fois par les allemands, mais depuis lors, la vidéosurveillance a largement évolué.

Avant de parler des principes plus récent de vidéo surveillance, faisons un bond dans l'histoire, afin de connaître les origines de la vidéosurveillance.

1.2.1. *Historique*

La vidéo-surveillance fait partie de la famille des systèmes de sécurité électronique. Elle a vu le jour peu après la création de la télévision (dans les années 40) le premier système fut mis au point en Allemagne, en 1942, par la compagnie SIEMENS afin d'observer le lancement des fusées, avec cette invention il était désormais possible de visualiser des images captée par une caméra (en N&B à l'époque) et de les transmettre par câble sur un moniteur. Les principes de base de ce que l'on appellera plus tard la CCTV (Closed Circuit of TV ou TV en circuit fermé) étaient nés. [14]

L'objectif principal des systèmes de CCTV est de pouvoir visualiser à distance les endroits stratégiques d'un lieu et ce, en regardant les images sur un récepteur TV (ou moniteur). Cela a permis de réduire les coûts de gardiennage humain. En effet un gardien pouvait désormais superviser l'ensemble d'un site grâce aux caméras de surveillance. Au début il fallait relier chaque caméra à un moniteur unique, mais très vite sont apparu des systèmes de commutation. Ces derniers permettait de relier plusieurs

³ *Dictionnaire Le Nouveau Petit Robert*, Dictionnaires le Robert, Paris, 2009.

caméras à un commutateur analogique (aussi appelé matrice de commutation) et de pouvoir voir des caméras, seulement sur quelque moniteurs. Il suffisait de composer le numéro de la caméra et du moniteur sur un clavier. [14]

Très vite un autre besoin est née, la nécessité de pouvoir enregistrer ces images afin de les stockées. Il est en effet très utile de revoir un évènement qui a été filmé par les caméras de surveillance, surtout lorsque ce dernier implique des délits criminels. C'est la compagnie RCA qui en 1954 lança sur le marché le premier appareil qui permettait d'enregistrer des images sur bande magnétique de la même manière qu'un magnétophone enregistrerait la voix.

1.2.2. Les différents systèmes de vidéosurveillance

1.2.2.1. La vidéosurveillance sur réseaux IP

Ce système relie un réseau de caméras IP à un système d'enregistrement numérique. Cela permet de stocker une quantité importante d'images, sans perte de qualité, tout en offrant la possibilité de les consulter rapidement. Ce système permettra également de réduire les couts d'utilisation des câbles réseaux et offre la possibilité d'accéder aux vidéos de surveillance à distance.

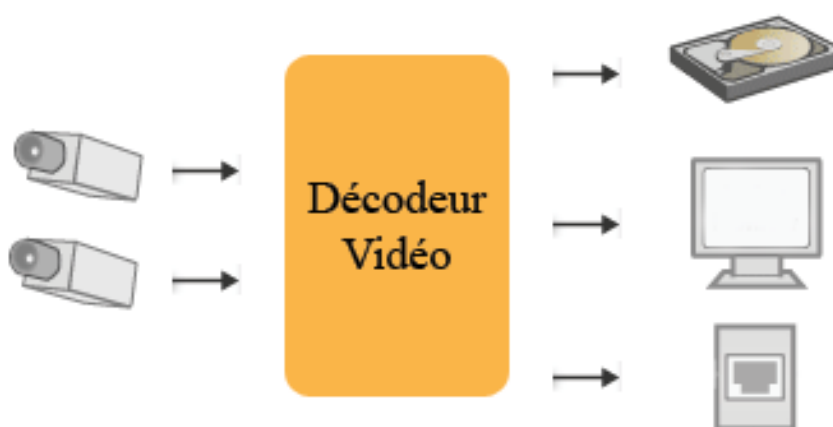


Figure 1.1. Vidéosurveillance sur réseau ip

1.2.2.2. Les kits de vidéosurveillance

Ce système est moins cher et est souvent utilisé dans des petits magasins, il est composé de quelques caméras, 3 ou 4, directement reliées à un moniteur ; Il n'offre pas la possibilité d'enregistrer les images, son but est de permettre « au vendeur d'un magasin » par exemple de surveiller en direct certains rayons de son magasin.

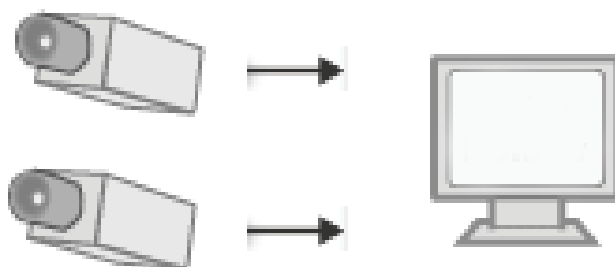


Figure 1.2. Kit de vidéosurveillance

1.2.2.3. La vidéosurveillance analogique

C'est l'un des plus anciens systèmes de vidéosurveillance et aussi le plus répandu dans le monde ; ce système fonctionne avec des caméras analogiques, qui peuvent être directement liées à un enregistreur ou à un moniteur, l'enregistrement fait avec ce système a une durée limitée.

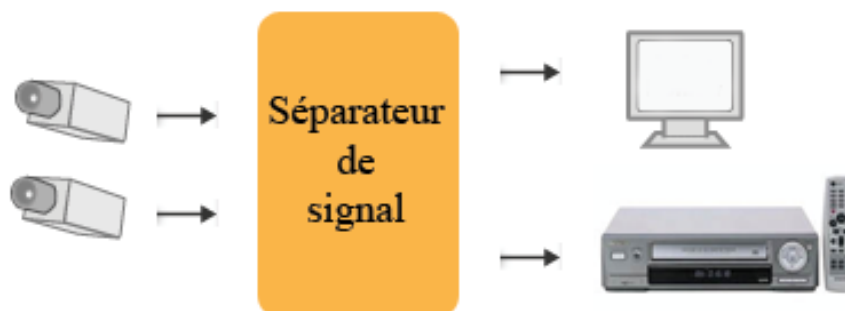


Figure 1.3. La vidéosurveillance analogique

Le plus récent des systèmes de vidéosurveillance est la vidéosurveillance sur réseau IP ; Ce système offre plusieurs avantages, par rapport aux autres systèmes.

Contrairement aux CCTV, la vidéosurveillance sur réseau IP compresse le flux vidéo et cela rend l'accès en direct à la vidéo beaucoup plus rapide ; la vidéosurveillance sur réseau IP permet également de faire plusieurs opérations sur les images en temps réel, comme le suivi des formes, la reconnaissance des personnes et des formes.

1.3.La reconnaissance faciale

La reconnaissance faciale est un domaine qui est en continuelle évolution, et de nos jours elle est de plus en plus présente dans nos vies au travers de nos téléphones mobiles et ordinateurs ; Malgré son évolution, la reconnaissance faciale fait encore l'objet de nombreuses études.

Il existe plusieurs algorithmes de reconnaissance faciale, et l'utilisation de ses algorithmes dépend du domaine d'application ; La reconnaissance faciale peut être divisée en trois grandes parties :

- La détection des visages ;
- L'extraction des caractéristiques ;
- La reconnaissance de visage.

1.3.1. La détection des visages

La détection des visages est la première phase de la reconnaissance faciale ; elle peut être considérée comme étant la phase la plus importante de la reconnaissance faciale car toutes les autres dépendent d'elle. Son efficacité influe directement sur les autres étapes de la reconnaissance faciale.

Il existe plusieurs méthodes de détection des visages, et ces dernières peuvent être réparties en quatre grandes catégories :

- Approches basées sur les connaissances acquises (Knowledge-based methods);
- Approches basées sur le « Template-matching » ;
- Approches basées sur l'apparence (Appearance-based methods);
- Approches basées sur les caractéristiques invariantes (Feature invariant approaches).

1.3.1.1. Approches basées sur les connaissances acquises (Knowledge-based methods)

Ces méthodes se basent sur la connaissance des différents éléments qui constituent un visage et des relations qui existent entre eux. Ainsi, les positions relatives de différents éléments clés tels que la bouche, le nez et les yeux sont mesurés pour servir ensuite à la classification 'visage', 'non visage'. Le problème dans ce type de méthode est qu'il est difficile de bien définir de manière unique un visage. Si la définition est trop détaillée, certains visages seront ratés tandis que si la description est trop générale le taux de faux positifs montera en flèche.

1.3.1.2. Approches basées sur le « Template-matching »

Des modèles caractéristiques d'un visage entier ou de sous-partie de visage (bouche, œil, nez) sont créées. La localisation se fait ensuite sur base de la corrélation de ces modèles avec les candidats.

1.3.1.3. Approches basées sur l'apparence (Appearance-based methods)

Ces approches appliquent généralement des techniques d'apprentissage automatique. Ainsi, les modèles sont appris à partir d'un ensemble d'images représentatives de la variabilité de l'aspect facial. Ces modèles sont alors employées pour la détection. L'idée principale de ces méthodes est de considérer que le problème de la détection de visage est un problème de classification (visage, non-visage). Une des approches les plus connues de détection de visage est l'Eigenface, et il en existe plusieurs autres comme Yalefaces,...

Les méthodes appartenant à cette catégorie sont les plus utilisées et les plus appréciées car elles s'exécutent beaucoup plus rapidement que les autres.

1.3.1.4. Approches basées sur les caractéristiques invariantes (Feature invariant approaches)

Ces approches sont utilisées principalement pour la localisation de visage. Les algorithmes développés visent à trouver les caractéristiques structurales existantes même si le point de vue, ou la condition d'éclairage changent. Puis ils emploient ces caractéristiques invariables pour localiser les visages. Nous pouvons citer deux familles de méthodes appartenant à cette approche : Les méthodes basées sur la couleur de la peau qu'on détaillera dans la section suivante, et les méthodes basées sur les caractéristiques de visage; elles consistent à localiser les cinq caractéristiques (deux yeux, deux narines, et la jonction nez/lèvre) pour décrire un visage typique.

1.3.2. L'extraction des caractéristiques

L'extraction des caractéristiques se base entièrement sur la détection des visages car les différentes caractéristiques se retrouvent sur le visage en question ; Cette extraction consiste à mettre de côté les yeux, le nez, les lèvres,... Une étude a révélé qu'il est très facile de reconnaître le visage d'une personne juste à partir de son nez et de ses lèvres, et la majorité des algorithmes de reconnaissance faciale assez récents exploitent cela.

1.3.3. La reconnaissance de visage

La reconnaissance de visage ne peut se faire sans la réalisation des deux étapes citées ci-haut ; Comme dit dans les points précédents, il existe plusieurs algorithmes aidant à la reconnaissance de visage, ces différents algorithmes ont vu le jour pour la plus part dans les années quatre-vingts dix grâce aux recherches de certains mathématiciens parmi lesquels nous pouvons citer M.A. Turk et A.P. Pentland du MIT Media Lab qui en 1991 ont créé l'un des plus grands algorithmes de reconnaissance de visage, le PCA (Principal Component Analysis) qui est beaucoup plus connu sous le nom de « Eigenfaces ». Comme la majorité, voire la totalité des algorithmes de reconnaissance de visage, Eigenfaces trouve ses bases dans les mathématiques, plus précisément des notions de statistique et d'algèbre (les vecteurs).

Le PCA a vu plusieurs autres algorithmes naître en s'inspirant de lui, nous citons l'un des plus connu, qui est le LDA (Linear Discriminant Analysis) plus connu sous le nom de Fisherfaces, il a vu le jour grâce à Belhumeur et Al de la Yale University (USA) en 1997.

Reposant plus sur des bases de programmation que des bases mathématiques, notre travail ne va pas s'attarder sur les formules mathématiques de reconnaissance faciale, mais il va néanmoins expliquer le principe fondamental de la reconnaissance de visage.

La reconnaissance de visage consiste à exprimer des images de départ selon une base de vecteurs orthogonaux particuliers contenant des informations indépendantes d'un vecteur à l'autre. Ces nouvelles données sont donc exprimées d'une manière plus appropriée à la reconnaissance du visage. En d'autre terme il s'agit d'extraire les informations caractéristiques d'une image de visage, pour l'encoder aussi efficacement que possible afin de la comparer à une base de données de modèles encodés de manière similaire.

Dans ses débuts, la reconnaissance faciale exigeait pour une seule personne, une série de photos prise sur plusieurs ongles, et ses dernières étaient utilisées comme base de référence, cela rendait les différents traitements très lent, car on devait faire plusieurs comparaison, mais actuellement, une seule photo suffit, et c'est l'unique photo qui sera modifier afin d'obtenir toutes les faces possible d'une personne ;

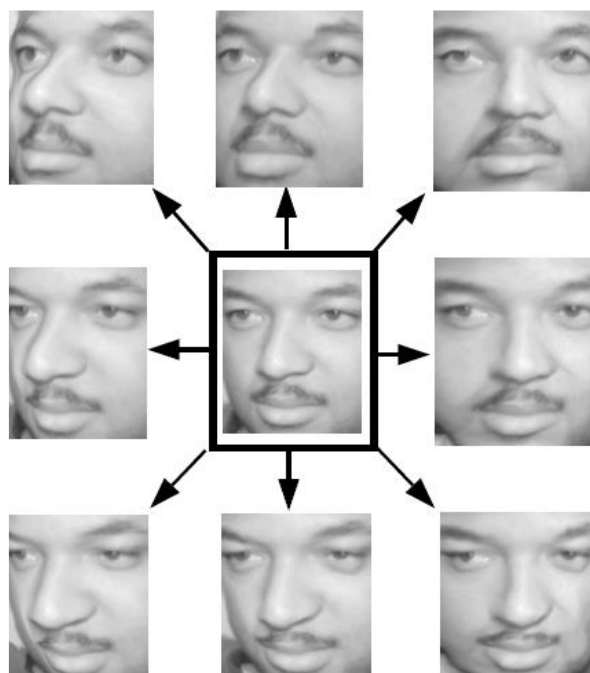


Figure 1.4. Méthode de déformation parallèle par David Beymer and Tomaso Poggio.

Voici une liste non exhaustive des algorithmes de reconnaissance de visage existant, en dehors du PCA et du LDA que nous avons énoncés ci-haut :

- Le KPPV (K Plus Proches Voisins)

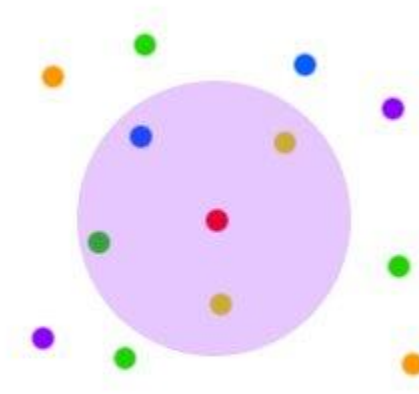


Figure 1.5. KPPV

- Le réseau de neurones

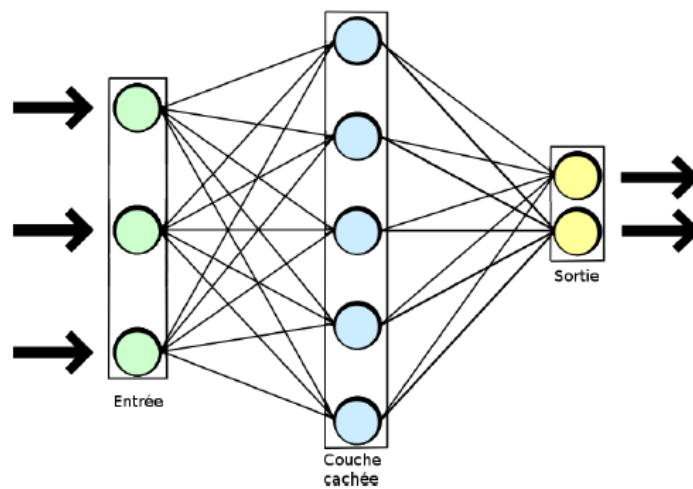


Figure 1.6. Réseau de neurones

➤ Viola et Jones



(a) Image initiale



(b) Visage extrait

Figure 1.7. Détection des visages avec l'algorithme de Viola et Jones

➤ L'EBGM (Elastic Bunch Graph Matching)



Figure 1.8. La sélection des points caractéristiques avec l'algorithme d'EBGM



Figure 1.9. Création d'un treillis avec l'algorithme d'EBGM

En ce qui concerne Open CV, toutes ses implémentations de la reconnaissance faciale est basé sur l'algorithme de Viola et Jones.

1.4.La bibliothèque Open CV

Open CV est le sigle de « Open Source Computer Vision Library », cette bibliothèque a vu le jour grâce aux travaux de recherche mener par Intel dans les années quatre-vingts dix ; Cesdits travaux avaient pour but de faciliter la reconnaissance d'image en mouvement.

Depuis plusieurs décennies, des algorithmes performant de traitement d'image ont vu le jour mais la plus part n'étaient pas ouverts au grand publique ; en 2000, Intel a rendu publique son projet, Open CV, afin de remédier à ce problème ; ce dernier est disponible sous la licence BSD actuellement

Open CV est essentiellement écrit en C et C++, il est multi-plateformes (Linux/Mac, Windows, iOS et Android), et il est également multi-langages (Python, Java, Matlab, c#, Perl, Ruby

Chapitre 2 : ETUDE PREALABLE

2.1.Introduction

Notre travail étant une solution apportée à l'Ecole Supérieure d'Informatique Salama afin de remédier au problème récurrent de vol dans ses labos, nous devrions parler des solutions existantes au sein de cette entité ainsi que des améliorations que nous apportons à cette dernière ; N'ayant jusqu'à ce jour aucun système de vidéo surveillance, notre solution sera la première à être installée dans les labos de l'ESIS.

De ce fait, dans ce chapitre, nous allons présenter l'ESIS, et analyser les systèmes de vidéos surveillances existants sous d'autres cieux vu que l'ESIS n'a aucun système similaire dans son sein.

2.2.Présentation de l'ESIS

L'ESIS est une institution d'enseignement supérieur et universitaire catholique d'obédience salésienne, située dans l'enclos de l'ITS (Institut Technique Salama) au numéro 5 de l'avenue femme congolaise, dans la commune de Lubumbashi ainsi que la ville portant le même nom. Sous l'initiative du père Joachim TSHIBANGU, salésien de Don Bosco, ESIS vu le jour en 2001 et son fonctionnement a été effectif qu'à partir du mois d'octobre 2002.

2.2.1. Historique

L'ESIS (école supérieure d'informatique Salama) fut créée en 2001 afin d'être le prolongement de l'ITS dans l'enseignement supérieur ; l'ITS offre des formations de niveau secondaire dans les domaines de la mécanique de fabrication et automobile, de l'électricité, de l'électronique, de la sérigraphie et de l'imprimerie. Ayant débuté avec une capacité de quatre-vingt-dix étudiants, l'ESIS compte à ce jour plus de mille étudiants.

La création de l'ESIS fut aussi influencée par les nombreuses demandes des parents d'élèves de l'ITS, qui se plaignaient du faible niveau d'encadrement au sein des établissements supérieurs publics.

L'ESIS a produit à ce jour des ingénieurs techniciens en cinq filières dont le taux d'insertion professionnel est supérieur à 70%. Il est à noter qu'ESIS est devenue

titulaire depuis mai 2010, de l'agrément définitif des instituts privés selon l'arrête ministériel n 0 078/2010, délivré par l'Etat congolais.

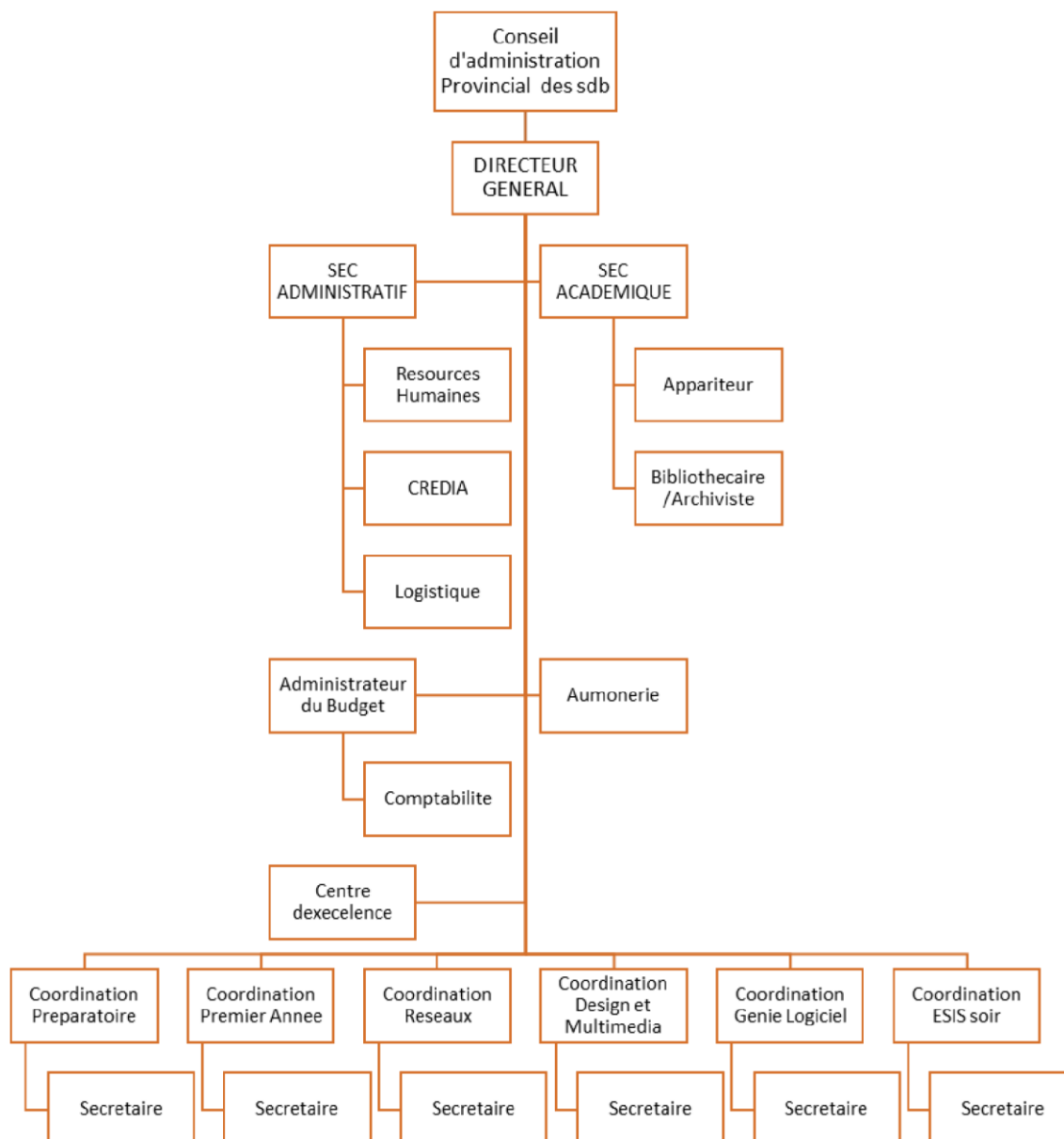
2.2.2. *Vision*

Etre incubateur d'entreprise, de société et des services informatique. ESIS forme des étudiants de telle manière qu'ils soient autant à la recherche de bons emplois, que de bonnes opportunités d'affaires pour se lancer dans la création de leurs propres entreprises.

2.2.3. *Mission*

Former des ingénieurs en informatique avec des solides bases scientifiques et des réelles habilités techniques et managériales dans le Design & Multimédia, Génie logiciel, et en Réseaux et Télécommunications.

2.2.4. Organigramme



2.3.Diagnostique de l'existant

Nous ne sommes pas les premiers à développer un système de surveillance, plusieurs systèmes similaires au notre existent, cependant, chacun de ces systèmes a des défauts et des qualités, et dans ce point, nous allons étudier les qualités ainsi que les défauts de certains systèmes de surveillances, et/ou de certaines technologies ayant trait à notre système.

Nous allons premièrement vous présenter les points forts ainsi que les points faibles des grandes plateformes de vision par ordinateur existante.

Google Vision API [15]

Cette API est une branche de la « Google's Cloud Plaform » et permet d'analyser les contenus des images ainsi que des vidéos.

Il permet principalement de détecter des contenus inadéquats sur des images ou vidéos selon certains critères, de classer très rapidement des images en des milliers de catégories, de détecter des émotions sur des visages, il intègre un OCR qui détecte automatiquement les langues.

Ses grandes faiblesses sont les suivantes :

- Il ne permet pas de reconnaissance faciale ;
- Il ne prend en charge que les images encodé en « base64 » ;
- La taille maximale des images à utiliser est de 4Mb (moins de 2 méga pixel) ;
- Cette API est disponible seulement sur le cloud de Google ;
- Certaines fonctionnalités de cette API sont payantes.

Microsoft Face API (Project Oxford) [16]

Face API est une partie de la plateforme « Cognitive Services » de Microsoft et permet d'analyser des visages sur des images et des vidéos.

Il permet de détecter des visages, de la reconnaissance faciale et de la détection des émotions.

Ses points faibles majeurs sont les suivants :

- La taille maximale des images à utiliser est de 4Mb (moins de 2 méga pixel) ;
- Il est payant, et coute assez cher.

IBM Watson Visual Recognition API [17]

Cette API est une partie du « IBM Watson™ Visual Recognition service », il permet de détecter des visages et d'identifier des célébrités sur des images.

Ses faiblesses sont les suivantes :

- L'identification de visage est limitée seulement aux célébrités ;
- Il est payant.

Cognitec [18]

Cognitec est plus des plus grandes entreprise au monde œuvrant dans le domaine de la vision par ordinateur, ses algorithmes de reconnaissance faciale, détection des visages et autres sont parmi les plus puissants au monde, mais hélas, Cognitec ne livre des services qu'à des gouvernement, des armées, ainsi que des grandes entreprises dans le domaine de la sécurité et des jeux vidéo.

Comme vous pouvez le constater, toutes les plateformes que nous avons mentionnées sont propriétaires et ne sont pas directement ouvert au grand public ; Il n'existe presque pas de plateforme de reconnaissance faciale Open Source, et 90% des plateformes existante dans le domaine de la vision par ordinateur sont basé sur la célèbre bibliothèque OPEN-CV.

Nous allons maintenant parler des logiciels de surveillance existant ; La majorité de logiciel de surveillance utilisé sont ceux qui sont livré à l'achat des caméras ip et ses logiciels ne renferment pas plusieurs options à part ceux qui sont primordiales comme visualisé la vidéo sur un post ou enregistrer les vidéos filmés ; En dehors de ceux-là, il existe plusieurs autres logiciel de surveillance disponible sur internet parmi lesquels nous pouvons citer « Eyeline Video Surveillance System » et « NCH Suite » qui sont les plus téléchargés et les plus utilisés.

« Eyeline Video Surveillance System » et « NCH Suite » sont des logiciels de surveillance gratuits et disponibles sur plusieurs sites web, leurs options principales sont les suivantes :

- La lecture de la vidéo en temps réel ;
- L'activation/désactivation de la détection des mouvements ;
- L'enregistrement des vidéos ;

Ces logiciels présentent néanmoins quelques défauts :

- Ce sont des logiciels « stand alone » ;
- Lors de la détection des mouvements, l'alerte est faite uniquement sur le poste où le logiciel tourne, et ce n'est pas une alerte sonore mais plutôt du texte ;

2.4.Solution proposée

Notre application de vidéo surveillance peut être divisée en trois grandes parties : le noyau du système qui sera installée sur le serveur, une application Android qui communiquera directement avec le noyau, et des clients web qui permettront également d'être en contact avec le noyau se trouvant sur le serveur.

Le noyau du système permettra de :

- Distribuer les vidéos de toutes les caméras en temps réel;
- Détecter les mouvements lors que la détection des mouvements est activée ;
- Lancer l'alarme lors de la détection des mouvements ;
- Envoyer des notifications ;
- Faire une vérification des personnes présentes dans la salle (reconnaissance faciale) ;

L'application Android et les clients web permettront de :

- Activer/Désactiver l'alarme ainsi que la détection des mouvements ;
- Visionner les vidéos en temps réel et lancer l'enregistrement des vidéos en cas de besoins
- Avoir recours aux différents registres de détection des mouvements et du retentissement de l'alarme ;

- Ajouter ou supprimer les personnes autorisés à gérer le système

L'application Android permettra en plus, de recevoir des notifications en temps réel, et le plus des clients web sera de permettre l'accès aux vidéos enregistrées.

Chapitre 3 : ANALYSE ET CONCEPTION DU SYSTEME INFORMATIQUE

3.1.Introduction

Dans ce chapitre, nous allons parler en long et en large de l'analyse de notre système ainsi que sa conception ; nous parlerons aussi de la méthode de développement logiciel utilisée pour la réalisation de notre application.

Premièrement nous allons présenter la méthode UP, qui est la méthode de développement logiciel que nous avons utilisée pour la réalisation de notre application, et par la suite, nous allons nous attarder sur l'expression des besoins et l'analyse proprement dit.

3.2.Présentation de la méthode UP

Avant d'entrer dans le vif du sujet, montrons brièvement la différence qui existe entre l'UP et l'UML, car ses deux concepts prêtent quelque fois confusion.

UP, qui est le sigle de « Unified Process », est un processus de développement logiciel alors que UML, qui est le sigle de « Unified Modeling Language » est un langage de modélisation objet et il permet de modéliser des systèmes.

La modélisation d'un système se fait selon trois axes sur lesquels sont répartis des diagrammes :

- L'axe fonctionnel qui est utilisé pour décrire ce que fait le système à réaliser,
Diagramme des cas d'utilisations
 (Diagramme d'activités)
 (Diagramme de séquences)
- L'axe structurel et statique qui est relatif à la structure du système,
Diagramme de classes
 Diagramme de composants
 Diagramme de déploiement
 Diagramme d'objets
- L'axe dynamique qui est relatif à la construction des fonctionnalités du système
Diagramme de séquences
 Diagramme de collaboration
 Diagramme d'activités

(Diagramme d'états/transitions)

Si UML permet de modéliser un système, il ne définit pas le processus d'élaboration des modèles.

- Dans quel ordre doit-on utiliser les neufs types de diagrammes?
- A quel moment de la conception d'un système doivent-ils intervenir?

Seul un processus de développement peut répondre à ces questions.

Le processus de développement régit les activités de production du logiciel selon deux aspects.

- L'aspect statique qui représente le processus en termes de tâches à réaliser.

L'aspect statique définit :

- Le « qui ». Les intervenants
- Le « comment ». Les activités à réaliser
- Le « quoi ». Les résultats d'une activité

- L'aspect dynamique qui représente la dimension temporelle du processus.

L'aspect dynamique représente :

- Le nom et le nombre de phases du cycle de vie du processus
- L'organisation et l'ordre de réalisation des activités de développement

Un processus gère généralement les activités suivantes:

- **L'expression des besoins**, consiste pour le client à élaborer un cahier des charges décrivant:
 - Les fonctionnalités du système à étudier.
 - La façon d'utiliser le système
- **La spécification des besoins**, permet aux utilisateurs, experts et aux informaticiens de finaliser le cahier des charges en levant les ambiguïtés et en éliminant les redondances
- **L'analyse des besoins**, vise à faire définir, par les experts et les utilisateurs les entités métier concernées par le système indépendamment de toutes considérations:
 - Techniques
 - et informatiques
- **La conception**, concerne les experts informatiques. On y détermine la manière de résoudre techniquement le problème posé. Comment réaliser les fonctionnalités attendues.
- **L'implémentation** consiste :
 - A construire les programmes dans un langage de programmation donné.

- A organiser logiquement les programmes en fonction de l'architecture logique choisie.
- A distribuer les programmes sur le système informatique selon l'architecture physique retenue.
- **Les tests** fonctionnels et techniques
 - Fonctionnels. Ils vérifient que le système implémente bien les fonctionnalités attendues.
 - Techniques. Ils vérifient que l'implémentation des fonctionnalités est techniquement correcte.
- **La maintenance**

3.2.1. Processus Unifié : (Unified Process)

Le processus unifié est un processus de développement logiciel construit sur UML. Il est itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. Il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel.

C'est un patron de processus pouvant être adaptée à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.

Caractéristiques essentielles du processus unifié :

- Le processus unifié est à base de composants,
- Le processus unifié utilise le langage UML (ensemble d'outils et de diagramme),
- Le processus unifié est piloté par les cas d'utilisation,
- Centré sur l'architecture,
- Itératif et incrémental.

Les adaptations d'UP les plus connues sont :

- RUP : Rational Unified Process
- XP : eXtreme Programming
- 2TUP : Two Tracks Unified Process

3.2.1.1. UP est piloté par les cas d'utilisation

L'objectif principal d'un système logiciel est de rendre service à ses utilisateurs ; il faut par conséquent bien comprendre les désirs et les besoins des futurs utilisateurs. Le

processus de développement sera donc centré sur l'utilisateur. Le terme utilisateur ne désigne pas seulement les utilisateurs humains mais également les autres systèmes. L'utilisateur représente donc une personne ou une chose dialoguant avec le système en cours de développement.

Les cas d'utilisation permettent d'illustrer les besoins. Ils détectent puis décrivent les besoins fonctionnels (du point de vue de l'utilisateur), et leur ensemble constitue le modèle de cas d'utilisation qui dicte les fonctionnalités complètes du système.

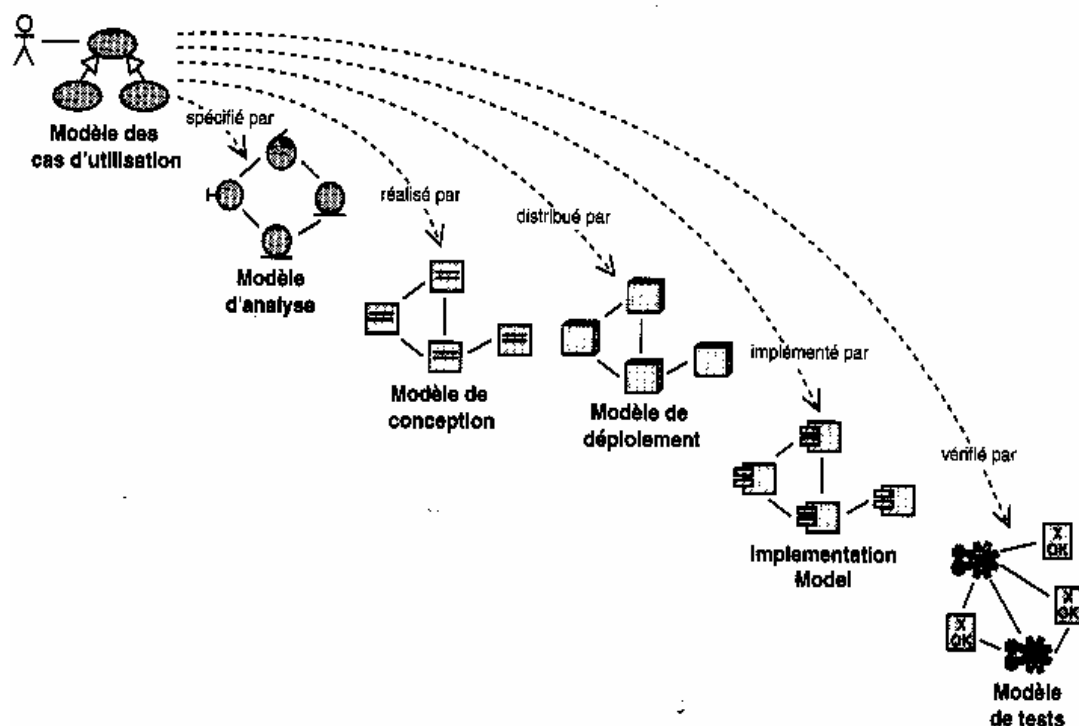


Figure 3.1. Modèle de conception basé sur les cas d'utilisations

- A partir du modèle des cas d'utilisation, les développeurs créent une série de modèles de conception et d'implémentation réalisant les cas d'utilisation.
- Chacun des modèles successifs est ensuite révisé pour en contrôler la conformité par rapport au modèle des cas d'utilisation.
- Enfin, les testeurs testent l'implémentation pour s'assurer que les composants du modèle d'implémentation mettent correctement en œuvre les cas d'utilisation.

3.2.1.2. *UP est centré sur l'architecture*

Dès le démarrage du processus, on aura une vue sur l'architecture à mettre en place. L'architecture d'un système logiciel peut être décrite comme les différentes vues du système qui doit être construit. L'architecture logicielle équivaut aux aspects statiques et dynamiques les plus significatifs du système. L'architecture émerge des besoins de l'entreprise, tels qu'ils sont exprimés par les utilisateurs et autres intervenants et tels qu'ils sont reflétés par les cas d'utilisation.

3.2.1.3. *UP est itératif et incrémental*

L'itération est une répétition d'une séquence d'instructions ou d'une partie de programme un nombre de fois fixé à l'avance ou tant qu'une condition définie n'est pas remplie, dans le but de reprendre un traitement sur des données différentes. Elle qualifie un traitement ou une procédure qui exécute un groupe d'opérations de façon répétitive jusqu'à ce qu'une condition bien définie soit remplie.

Une itération prend en compte un certain nombre de cas d'utilisation et traite en priorité les risques majeurs.

Avantages d'un processus itératif contrôlé :

- Permet de limiter les coûts, en termes de risques, aux strictes dépenses liées à une itération.
- Permet de limiter les risques de retard de mise sur le marché du produit développé (identification des problèmes dès les premiers stades de développement et non en phase de test comme avec l'approche « classique »).
- Permet d'accélérer le rythme de développement grâce à des objectifs clairs et à court terme.
- Permet de prendre en compte le fait que les besoins des utilisateurs et les exigences correspondantes ne peuvent être intégralement définis à l'avance et se dégagent peu à peu des itérations successives.

3.3.Expression des besoins

« La formalisation des besoins est un bon préalable à la conduite réussie d'un projet informatique.

Pour la maîtrise d'ouvrage, savoir exprimer toutes ses exigences, tant implicites qu'explicites, c'est l'assurance d'avoir décrit de manière complète ce qu'on attend.

Pour la maîtrise d'œuvre, au-delà de l'intérêt d'un contrat clair qu'elle aura contribué à construire avec sa maîtrise d'ouvrage, le référentiel d'exigences est un outil puissant de traçabilité tout au long des phases du projet.».⁴

⁴ Oxiane, *UML2 expression des besoins*, 20 juillet 2017, p.1.

3.3.1. Formalisation des besoins

3.3.1.1. Diagramme des cas d'utilisations

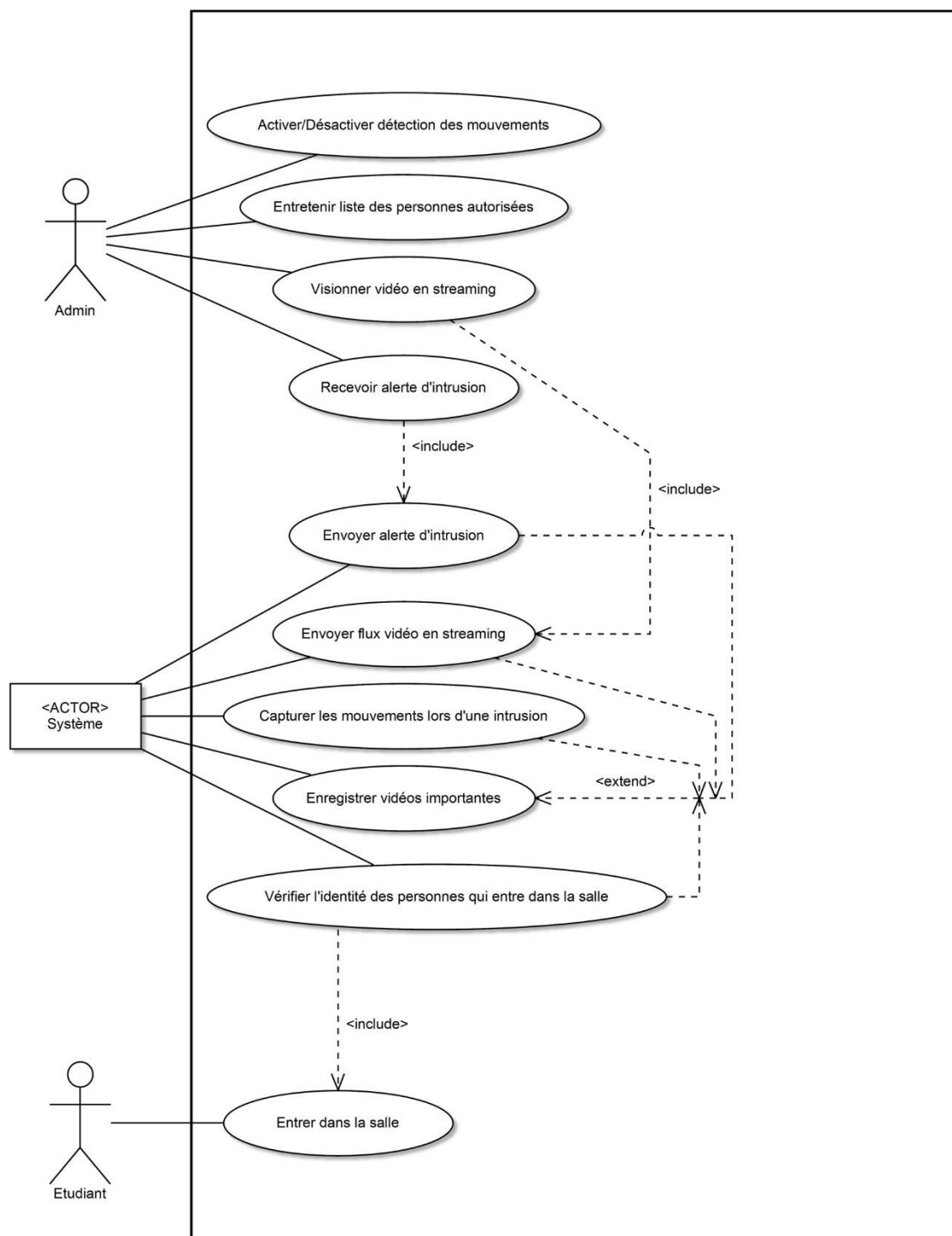


Figure 3.2. Diagramme des cas d'utilisations système

3.3.1.2. Description textuelle

✓ Premier cas

- Nom du cas : Activer / désactiver détection des mouvements
- Objectif : Permettre ou empêcher l'envoi de l'alerte d'intrusion
- Acteur principal : Admin
- Acteur secondaire : -
- Précondition : -
- Scénarios :
 - Nominal : Cliquer sur le bouton d'activation ou de désactivation
 - Alternatif : -
 - Exception : -
- Post-Condition : Etat de la détection des mouvements changé

✓ Deuxième cas

- Nom du cas : Visionner vidéo en streaming
- Objectif : Prendre connaissance de tout ce qui se passe dans la salle à distance
- Acteur principal : Admin
- Acteur secondaire : Système
- Précondition : Envoyer flux vidéo en streaming
- Scénarios :
 - Nominal : Lire le flux vidéo reçu via un navigateur
 - Alternatif :
 - Exception : Mauvaise connexion internet
- Post-Condition : Vidéo lue

✓ Troisième cas

- Nom du cas : Recevoir alerte d'intrusion
- Objectif : Prendre connaissance des intrusions
- Acteur principal : Admin
- Acteur secondaire : Système
- Précondition : Envoyer alerte d'intrusion

- Scénarios :
 - Nominal :
 - Alternatif : Retentissement de l'alarme
 - Exception : Téléphone de l'administrateur indisponible
- Post-Condition : Notification reçue

✓ Quatrième cas

- Nom du cas : Entretenir liste des personnes autorisées
- Objectif : Ajouter ou retirer une personne de la liste de ceux qui sont autorisées à entrer dans la salle
- Acteur principal : Admin
- Acteur secondaire : Système
- Précondition : -
- Scénarios :
 - Nominal : Avoir l'identité de la personne
 - Alternatif : -
 - Exception : -
- Post-Condition : Liste des personnes mise à jour

✓ Cinquième cas

- Nom du cas : Envoyer alerte d'intrusion
- Objectif : Envoyer une notification à l'admin en cas d'intrusion non autorisée
- Acteur principal : Système
- Acteur secondaire : Admin
- Précondition : -
- Scénarios :
 - Nominal : -
 - Alternatif : Retentissement de l'alarme
 - Exception :
 - ❖ Pas de sms dans la Sim
 - ❖ Pas de connexion internet
- Post-Condition : Notification envoyée

✓ Sixième cas

- Nom du cas : Envoyer flux vidéo en streaming
- Objectif : Permettre à l'admin de voir les vidéos des caméras en temps réel
- Acteur principal : Système
- Acteur secondaire : Admin
- Précondition : -
- Scénario :
 - Nominal : -
 - Alternatif : enregistrer des vidéos sur un disque dur externe
 - Exception : Pas de connexion internet
- Post-Condition : Flux vidéo envoyé

✓ Septième cas

- Nom du cas : Capturer les mouvements lors d'une intrusion
- Objectif : Prendre des photos de la personne non désirée et l'envoyer à l'admin
- Acteur principal : Système
- Acteur secondaire : Admin
- Précondition : -
- Scénario :
 - Nominal : -
 - Alternatif : sauvegarder la photo en local
 - Exception : pas de connexion internet
- Post-Condition : Photo envoyée et enregistrée

✓ Huitième cas

- Nom du cas : Enregistrer vidéo importante
- Objectif : Avoir des traces vidéo des mouvements pendant un certain moment
- Acteur principal : Système
- Acteur secondaire : -
- Précondition : -
- Scénario :
 - Nominal : -
 - Alternatif : -

- Exception : -
- Post-Condition : Vidéo enregistrée

✓ Neuvième cas

- Nom du cas : Vérifier l'identité des personnes qui entre dans la salle
- Objectif : Vérifier si la personne entrante est permise d'entrer
- Acteur principal : Système
- Acteur secondaire : Etudiant
- Précondition : -
- Scénario :
 - Nominal : -
 - Alternatif : -
 - Exception : -
- Post-Condition : Identité vérifiée

✓ Dixième cas

- Nom du cas : Entrer dans la salle
- Objectif : Accéder à la salle
- Acteur principal : Etudiant
- Acteur secondaire : -
- Précondition : -
- Scénario :
 - Nominal : -
 - Alternatif : -
 - Exception : -
- Post-Condition : Etudiant entré dans la salle

3.3.1.3. Structuration des cas d'utilisations

Tableau 3.1. Structuration des cas d'utilisations

Cas d'utilisation	Acteur	Package
Visionner vidéo en streaming	Admin Système	Gestion des étudiants
Entretenir la liste des personnes autorisées		
Envoyé flux vidéo en streaming		
Activer / Désactiver détection des mouvements	Admin Système	Gestion des intrusions
Recevoir alerte d'intrusion		
Envoyer alerte d'intrusion		
Capturer les mouvements		
Vérifier l'identité des personnes	Système	Gestion des informations
Enregistrer vidéo importante		
Entrer dans la salle	Etudiant	Etudiant

3.3.1.4. Diagramme d'activité

Le diagramme d'activité représente les règles d'enchaînement des actions et décisions au sein d'une activité. Il peut également être utilisé comme alternative au diagramme d'états pour décrire la navigation dans un site web.⁵

Notre application ayant deux point d'entrés, celui de l'étudiant ainsi que celui de l'administrateur, nous avons réalisé deux diagramme d'activité relatifs à ces derniers.

⁵ Pascal ROQUES, *UML 2 Modéliser une application web*, 4^{ème} Edition, Eyrolles, Paris, 2008, p.7.

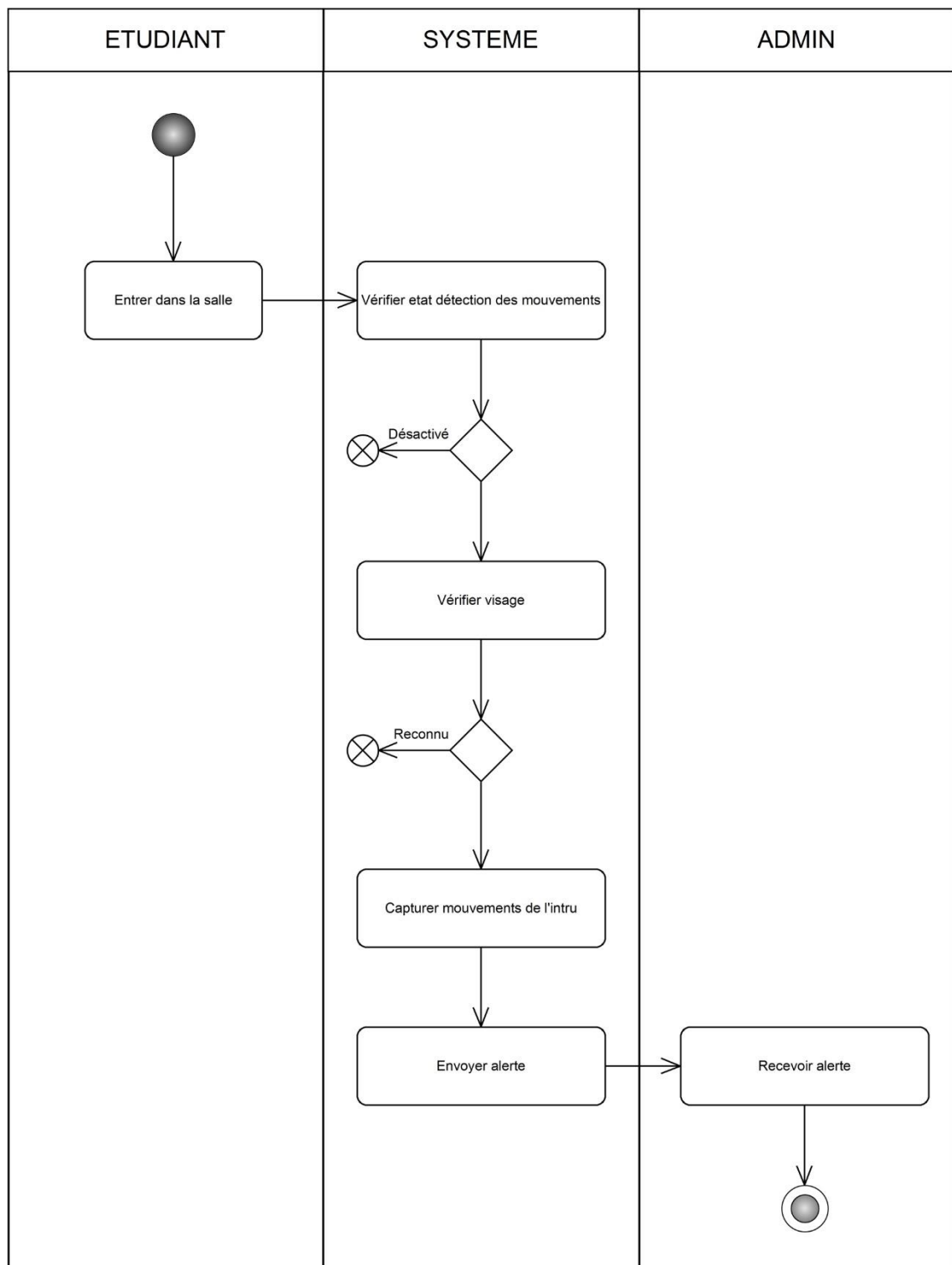


Figure 3.3. Premier diagramme d'activité

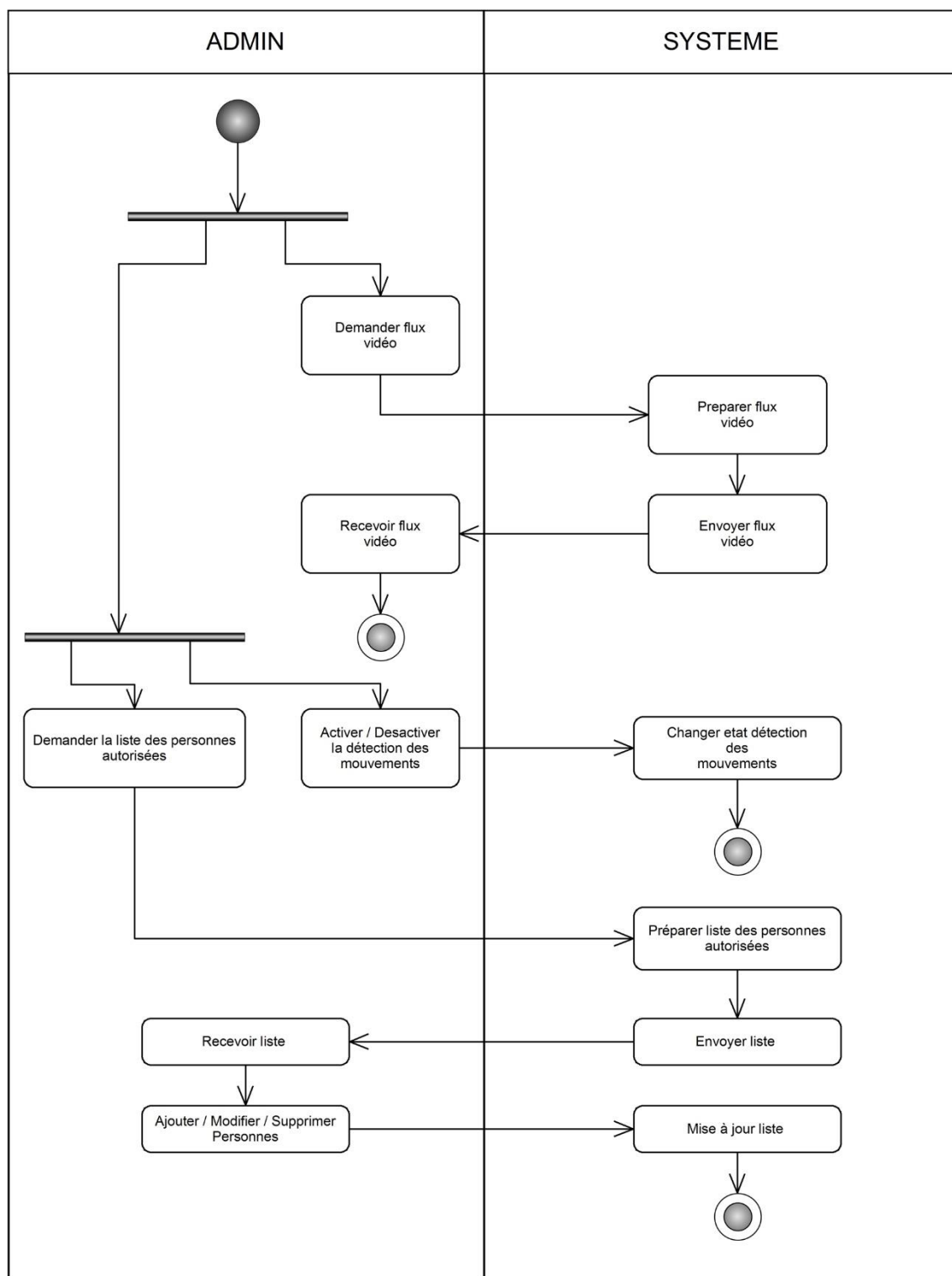


Figure 3.4. Deuxième diagramme d'activité

Chapitre 4 : IMPLEMENTATION

Dans ce présent chapitre, nous présenterons les différents outils qui nous ont permis de réaliser notre application.

4.1. Environnement logiciel

Du point de vue logiciel, notre application peut être subdivisée en quatre grande partie, qui sont, les modules fait en python, de gestion des caméras, de mouvements et de reconnaissance faciale, du module fait en c#, de gestion des notifications et alertes, du module fait en php, de gestion générale de l'application et du module Android de gestion générale de l'application.

Pour la réalisation des modules fait en python, nous nous sommes servis de l'IDE : JetBrains PyCharm Community Edition 4.0.6, pour le c# Visual Studio 2013, pour le php, Netbeans 8.0.2 et pour Android, Android studio 2.1.2.

En ce qui concerne la base de données, nous nous sommes servis de l'SGBD MySQL.

4.2. Environnement matériel

Les différents matériels utilisés pour la réalisation de notre travail sont les suivants :

- Ordinateur ;
- Caméra USB ;
- Routeur wifi ;
- Modem GSM ;
- Téléphones Android.

4.3.Présentation de l'application

4.3.1. Quelques captures d'écran de l'application Android

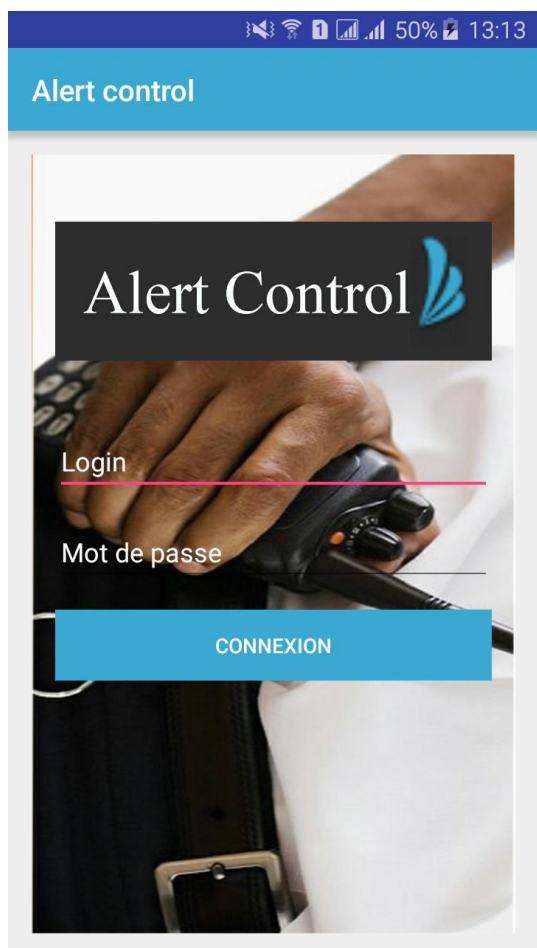


Figure 4.1. Interface de connexion



Figure 4.2. Interface d'activation/désactivation de l'alarme

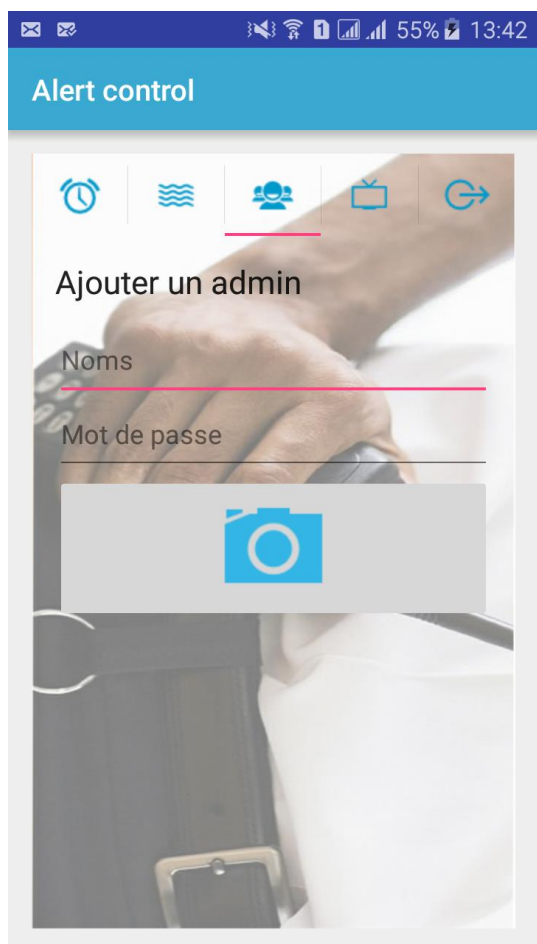


Figure 4.4. Interface d'ajout des admin



Figure 4.3. Interface d'activation/désactivation de la détection des mouvements

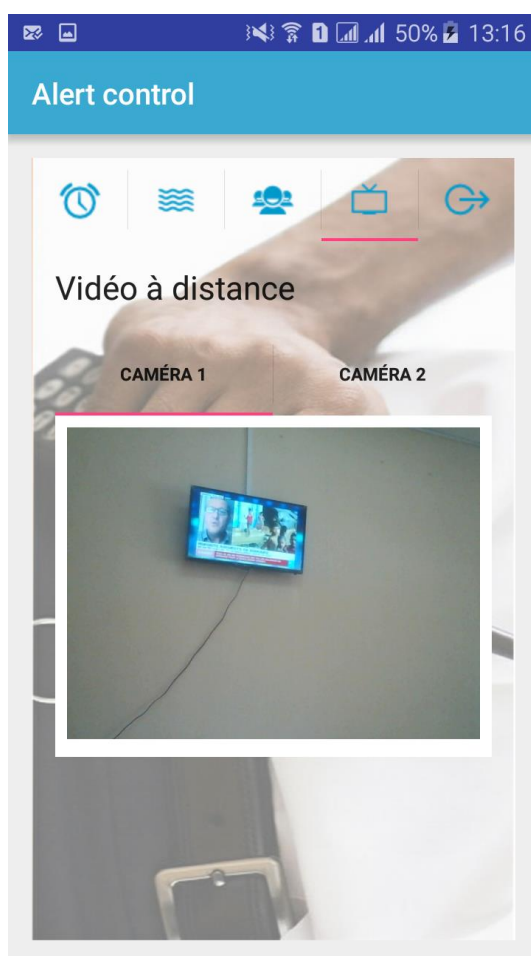


Figure 4.5. Interface de visualisation des caméras à distance

4.3.2. Quelques captures d'écran de l'application Web

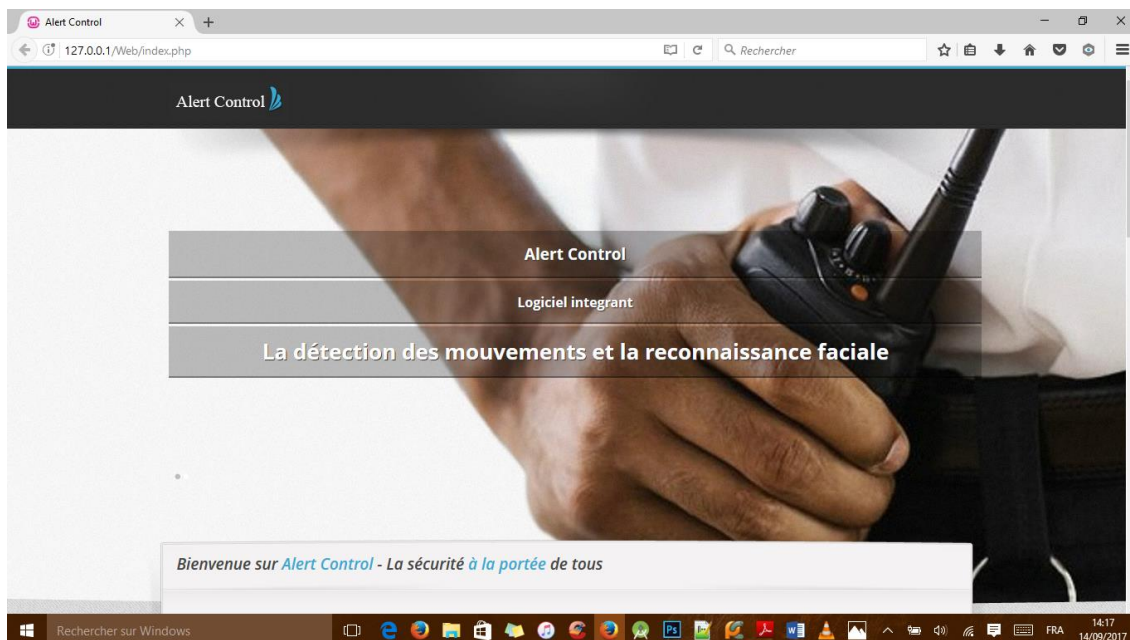


Figure 4.6. Interface d'accueil

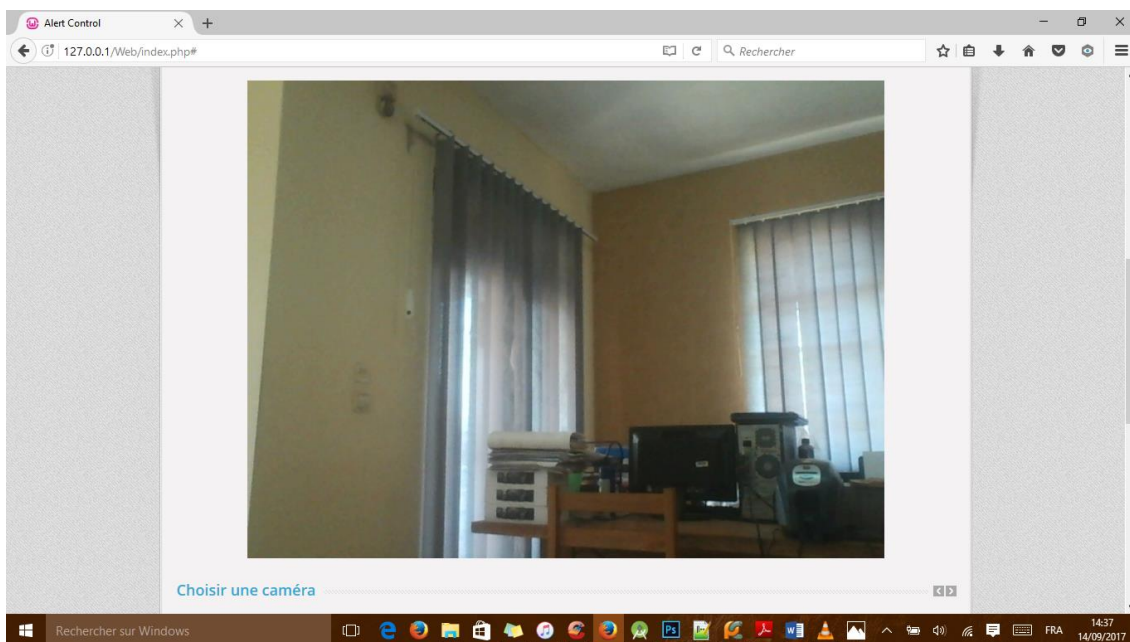


Figure 4.7. Fenêtre principale de visualisation

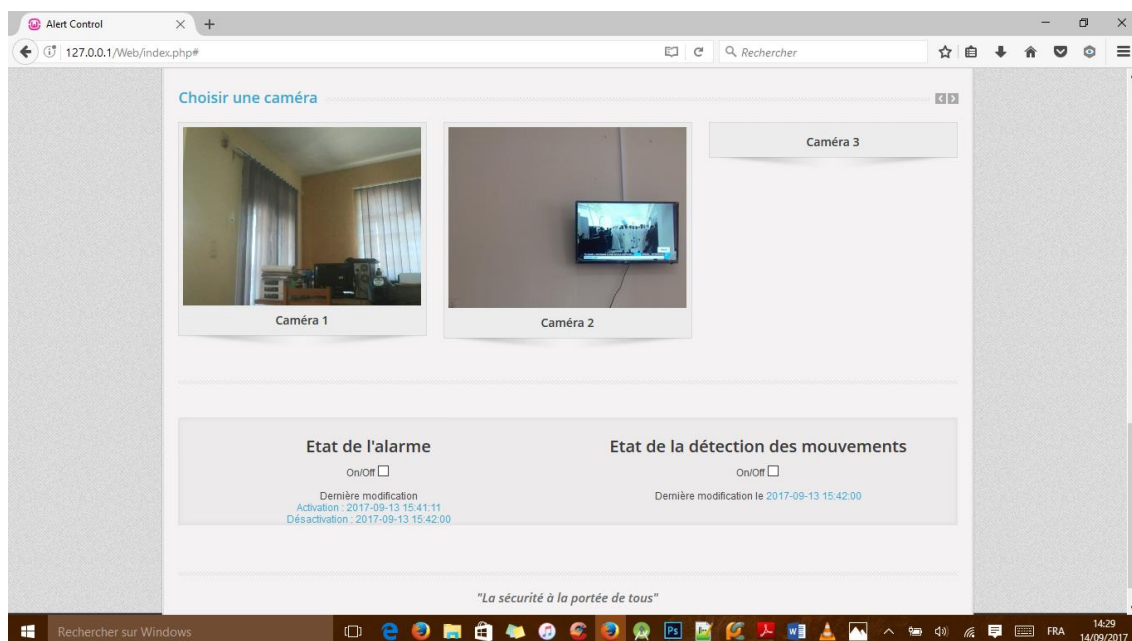


Figure 4.8. Choix de la caméra à visualiser, activation/désactivation de l'alarme et la détection des mouvements

4.3.3. Capture de l'illustration de la reconnaissance faciale et de la détection des mouvements

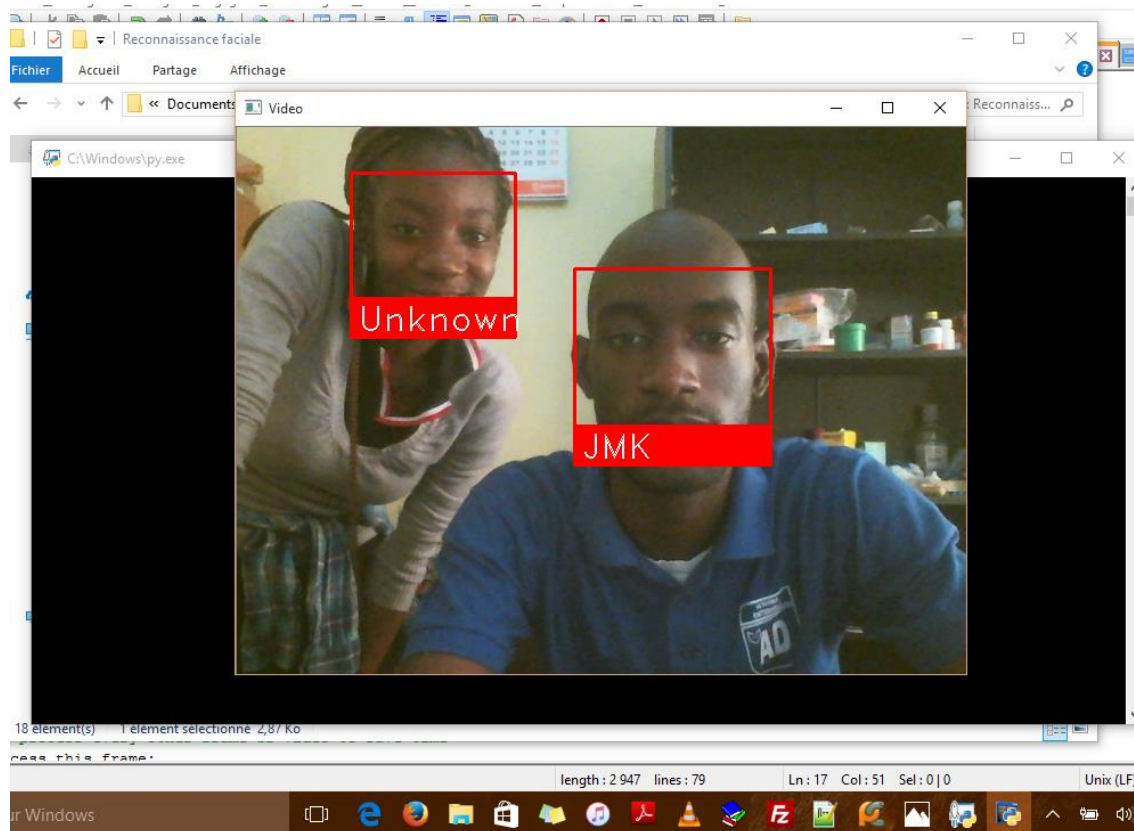


Figure 4.9. Reconnaissance faciale

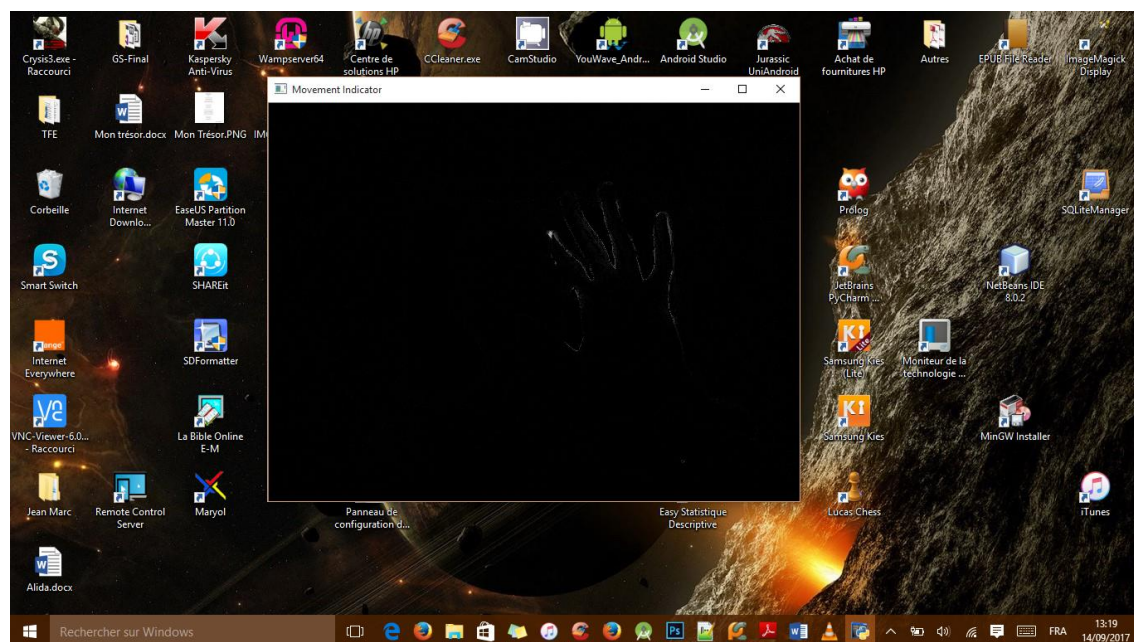


Figure 4.10. Indicateur des mouvements

4.3.4. Capture de quelques fonctions utilisées

```
@click.command()
@click.option('--eyes', help='add eyes detection', type=bool, default=False)
@click.option('--face_cascade', help='frontal face classifier', type=click.Path(exists=True))
@click.option('--eyes_cascade', help='eyes classifier', type=click.Path(exists=True))
def main(eyes, face_cascade, eyes_cascade):

    face_text = ""
    # Definition des bases de connaissance (Visage et yeux)
    face_cascade = cv2.CascadeClassifier(face_cascade)
    eye_cascade = cv2.CascadeClassifier(eyes_cascade)

    video_capture = cv2.VideoCapture(1)

    while True:
        # Capture image par image
        ret, frame = video_capture.read()

        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        faces_data = face_cascade.detectMultiScale(gray, 1.2, 5)

        # encadrer le visage par un rectangle
        for (x, y, w, h) in faces_data:
            cv2.rectangle(img=frame,
                          pt1=(x, y),
                          pt2=(x + w, y + h),
                          color=(255, 0, 0),
                          thickness=2)
```

Figure 4.11. Détection des visages et des yeux

```

from flask import Flask, render_template, Response
from camera import VideoCamera
import time

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    return Response(gen(VideoCamera()),
                    mimetype='multipart/x-mixed-replace; boundary=frame')

while True:
    if __name__ == '__main__':
        app.run(host='192.168.0.100', debug=True, threaded=True)

```

Figure 4.12. Envoyer d'un flux vidéo


```

private boolean envoyerAdmin() {

    new Thread(new Runnable() {
        @Override
        public void run() {
            ArrayList<NameValuePair> donnees = new ArrayList<NameValuePair>();
            donnees.add(new BasicNameValuePair("noms", noms.getText().toString()));
            donnees.add(new BasicNameValuePair("pwd", pwd.getText().toString()));
            donnees.add(new BasicNameValuePair("photo", photo));

            try {
                HttpClient httpclient = new DefaultHttpClient();
                HttpPost httpost = new HttpPost(nomsAdmin);

                httpost.setEntity(new UrlEncodedFormEntity(donnees));
                HttpResponse reponse = httpclient.execute(httpost);
                HttpEntity entity = reponse.getEntity();

            } catch (Exception e) {
                // TODO Auto-generated catch block
                Log.e("alert_control", "Erreur : " + e.toString());
                repp = false;
            }
        }
    }).start();
    return repp;
}

```

Figure 4.13. Envoi des données Android à PHP

CONCLUSION GENERALE

Le présent travail a porté sur la conception d'un système de vidéosurveillance intégrant la reconnaissance faciale et la détection des mouvements afin de remédier aux cas de vol et perte de matériel dans le labo génie logiciel.

Notre système est basé sur un réseau de caméras IP qui fournissent des renseignements au cœur du système afin de gérer les intrusions indésirable, et/ou l'accès à des endroits interdits ;

Ce dit système est principalement établi avec le langage python et il est centré sur les options que nous offre la librairie OPEN-CV.

Notre système n'a pas la prétention d'être le meilleur système de vidéosurveillance existant, et nous ne pouvons pas prétendre qu'il est complet ; De ce fait, nous espérons que ce travail aidera les étudiants et chercheurs qui auront à œuvrer le domaine de la vidéosurveillance à dépasser les limites auxquelles nous nous sommes butées.

BIBLIOGRAPHIE

- [1] LOÏC GOUARIN, *Les bases du langage Python*, Orsay, 6 Decembre 2010 ;
- [2] GERARD SWINNEN, *Apprendre à programmer avec Python 3*, <http://inforef.be/swi/python.htm> , 2012 ;
- [3] ALEXANDER MORDVINTSEV & ABID K, *OpenCV-Python Tutorials Documentation Release 1*, 12 Avril 2017 ;
- [4] PHILIPP WAGNER, *Face Recognition with Python*, http://www.byte_sh.de , 18 Juillet 2012;
- [5] ALEXANDRE BONHOMME, FLORENT MAUFRAS, TIMOTHEE PLANTE, *Implémentation d'un système de détection et de reconnaissance faciale* ;
- [6] SOUHILA GUERFI ABABSA, *Authentification d'individus par reconnaissance de caractéristiques biométriques liées aux visages 2D/3D*, Université d'Evry val d'Essonne, France ,03 octobre 2008 ;
- [7] FERJANI MOHAMMED, SERIAI ABDERRAHMANE, HENNANI HAKIM, SEDDIK ANNES, *Rapport du projet TER, La Vidéosurveillance à distance*, Université de Montpellier 2, France ;
- [8] OLIVIER SCHWANDER, *Identification de visages*, Ecole Normale Supérieure de Lyon, France, 4 juin - 27 juillet 2007 ;
- [9] Découverte d'Open CV, <http://www.pobot.org>, 16 décembre 2010;
- [10] NICOLAS MORIZET, THOMAS EA, FLORENCE ROSSANT, FREDERIC AMIEL, AMARA AMARA, *Revue des algorithmes PCA, LDA et EBGM utilisés en reconnaissance 2D du visage pour la biométrie*, Institut Supérieur d'Electronique de Paris, France, 2006 ;
- [11] MATHIEU VAN WAMBEKE, *Reconnaissance et suivi de visages et implémentation en robotique temps-réel*, Université Catholique de Louvain, France, 2009-2010 ;
- [12] Pascal ROQUES, *UML 2 Modéliser une application web*, 4ème Edition, Eyrolles, Paris, 2008 ;
- [13] Pascal ROQUES, Franck Vallée, *UML en action, de l'analyse des besoins à la conception en Java*, deuxième édition, Eyrolles, Paris, 2003.

- [14] Vidéosurveillance–Historique, <https://securivision.wordpress.com/2011/07/22/la-videosurveillance-historique/amp/>
- [15] Google vision API, <https://cloud.google.com/vision/>
- [16] Microsoft Face API (Project Oxford) <https://www.microsoft.com/cognitive-services/en-us/>
- [17] IBM Watson Visual Recognition API
<https://www.ibm.com/watson/developercloud/visual-recognition>
- [18] Cognitec, <http://www.cognitec.com/>