

FACULTE DES SCIENCES AIN CHOCH
DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

PROJET DE FIN D'ETUDES

THEME

Application de reconnaissance
faciale

FILIÈRE

Sciences Mathématiques et informatique

OPTION

Base de Données

RÉALISÉ PAR

- BENAOU Mohammed
- EL KHADIRI Mohamed Amine
- BENOUDA Mohammed

ENCADRÉ PAR

Pr. BENABELLAH

Année universitaire :

2015 / 2016

REMERCIEMENTS

Au terme de notre projet de fin d'étude, nous tenons à remercier toutes les personnes, qui par leurs conseils, leurs suggestions ou par leur simple présence nous ont permis de rendre notre travail aussi instructif et efficace que plaisant.

Nos chers parents, que nulle dédicace ne puisse exprimer nos sincères sentiments, pour leur patience illimitée, leur encouragement continue, leur aide, en témoignage de notre profond amour et respect pour leurs grands sacrifices.

Nous remercions tout spécialement notre encadrente Madame *Samia BENABDELLAH* pour son encadrement tout au long de ce projet, sa patience, sa disponibilité et ses conseils toujours avisés.

Enfin, nos remerciements vont à tous les enseignants du département de mathématiques et informatique pour la qualité de la formation qu'ils nous ont fournie et tous les membres du jury pour avoir accepté de juger ce modeste travail.

LISTE DES ABREVEATIONS

CV	: COMPUTER VISION
CVPR	: COMPUTER VISION AND PATTERN ROCOGNITION
UML	: UNIFIED MODELING LANGUAGE
NSF	: NATIONAL SCIENCE FONDATION
PCA	: PRINCIPALE COMPONENT ANALYSIS
BSD	: BERKELEY SOFTWARE DISTRIBUTION
TBB	: THREADING BUILDING BLOCKS
SLN	: SOLUTION
GCC	: GNU COMPILER COLLECTION

LISTE DES FIGURES

Figure 1 : Plan Prévisionnel	6
Figure 2 : Plan Réel.....	6
Figure 3 : Diagramme de cas d'utilisation	8
Figure 4 : Diagramme de séquence détection de visage	9
Figure 5 : Diagramme de séquence Enregistrement	10
Figure 6 : Diagramme de séquence Reconnaissance	10
Figure 7 : Conception du système.....	11
Figure 12 : Algorithme de l'application	12
Figure 13 : Les Eigenfaces.....	22
Figure 8 : Face recognition	25
Figure 9 : Training System	26
Figure 10 : Facial Expression	27
Figure 11 : Tableau de comparaison des outils.....	28
Figure 14 : Listes des bibliothèques.....	31
Figure 15 : Exécution des commandes	32
Figure 16 : Exécution des commandes 2	32
Figure 17 : Exécution des commandes 3	33
Figure 18 : Installation des bibliothèques sur Visual Studio	34
Figure 19 : Installation de Open CV et TBB avec Cmake.....	35
Figure 20 : Installation de TBB	35
Figure 21 : Installation de OpenCV sur Visual Studio	36
Figure 22 : Installation de OpenCV	37
Figure 23 : Installation de OpenCV	37
Figure 24 : Installation de OpenCV sur Visual Studio	38
Figure 25 : La disparition des erreurs d'inclusion	39
Figure 26 : Test d'application 1.....	39
Figure 27 : Test d'application 2.....	40
Figure 28 : Test d'application 3.....	40
Figure 29 : Test d'application 4.....	41
Figure 30 : Test d'application 5.....	41

RESUMÉ

Jusqu'à nos jours, la reconnaissance des personnes en utilisant des critères biologiques est un sujet de recherche très actif. L'idée générale de ce projet est la mise en place d'une application de reconnaissance qui se base sur les visages, pour les besoins de sécurité et de la facilité des tâches de la vie quotidienne, en faisant les recherches et l'implémentation d'un système qui répond aux besoins demandés.

Dans le cadre de l'amélioration de nos compétences et l'enrichissement de nos connaissances, un projet est essentiel pour mettre notre formation en pratique afin d'acquérir les savoirs nécessaires pour une bonne intégration professionnelle.

Ce rapport résume le travail réalisé dans le Projet de Fin d'études, pour l'obtention du diplôme du Licence en Mathématiques et Informatique, au sein de l'Université Hassan II de Casablanca, Faculté des sciences Ain-Chock.

Le présent rapport montre l'essentiel des travaux effectué dans ce projet.

Mots clés : Reconnaissance faciale, Détection de visage, Open CV, Imagerie, PCA.

ABSTRACT

Until today the recognition of individuals using biological criteria is a subject of active researches, the general idea of this project is the establishment of a humans recognition application based on faces, for security needs and the facility needs of daily life, doing the researches and implementation of a system that responds to the requested needs.

As part of improving our skills and enrichment of our knowledge, a project is essential to practice our studies to acquire the necessary knowledge for a good professional integration.

This report summarizes the work realized in the study Final Project for graduation of "Licence Mathematiques et Informatique", within the University Hassan 2 Casablanca, Faculty sciences of Ain-Chock.

This report shows most of the work done in this project.

Keywords: Face Recognition, Face Detection, OpenCV, Imaging.

Table de Contenu

INTRODUCTION GENERALE	1
PARTIE 1.....	2
CHAPITRE 1 : ÉTUDE GENERALE DU PROJET	2
<i>INTRODUCTION</i>	3
<i>1.1. 1. Etude de l'existant</i>	3
<i>1. 1. 2. Problèmatique</i>	4
<i>1. 1. 3. - Solution / Objectif</i>	5
<i>1. 1. 4. Planification</i>	6
<i>CONCLUSION</i>	6
PARTIE 2.....	7
CHAPITRE 1 : ANALYSE ET CONCEPTION.....	7
<i>INTRODUCTION</i>	8
<i>2. 1. 1. Analyse et Conception</i>	8
<i>2. 1. 2. Architecture du système</i>	11
<i>2. 1. 3. Méthodes utilisées</i>	11
<i>2. 1. 4. Outils existants</i>	24
<i>2. 1. 5. Comparaison et prise de décision</i>	27
<i>CONCLUSION</i>	28
CHAPITRE 2 : RÉALISATION DU PROJET	29
<i>INTRODUCTION</i>	30
<i>2. 2. 1. Inclusion des bibliothèques</i>	30
<i>2. 2. 2. Teste de l'application</i>	39
<i>CONCLUSION</i>	41
CONCLUSION GENERALE.....	42
REFERENCES	43

INTRODUCTION GENERALE

Pour améliorer nos compétences et mettre en œuvre nos connaissances afin d'assurer une bonne intégration professionnelle, nous avons effectué un projet de fin d'études qui entre dans le cadre de l'obtention de notre diplôme du Licence en Mathématiques et Informatique, Ce projet a pour objectif l'implémentation d'un logiciel de reconnaissance faciale.

Savoir déterminer de manière à la fois efficace et exacte l'identité d'un individu est devenu un problème critique car de nos jours l'accès sécurisé et la surveillance constituent un sujet de très grande importance. En effet bien que nous ne nous en rendions pas toujours compte, notre identité est vérifiée quotidiennement par de multiples organisations : lorsque nous accédons à notre lieu de travail, lorsque nous utilisons notre carte bancaire, lorsque nous nous connectons à un réseau informatique.

Plusieurs difficultés de la vie quotidienne ont donné naissance à l'idée d'utiliser des caractéristiques biométriques comme moyens d'identification. En effet, chaque individu possède des caractéristiques qui lui sont propres : sa voix, ses empreintes digitales, les traits de son visage, la forme de sa main, sa signature et jusqu'à son ADN. Ces données dites « biométriques » peuvent ainsi être utilisées pour l'identifier.

Il convient en dépit de tout ce qui précède de ne pas renoncer aux bénéfices de l'informatisation, et pour cela de faire appel à des outils informatiques qui apportent une garantie pour répondre à nos besoins.

Le projet présenté dans ce rapport consiste à utiliser une application de détection et de reconnaissance de visage. Le présent rapport est constitué de deux parties : La première contient un chapitre d'intitulé « ETUDE DE PROJET », à pour objectif de mettre le projet dans son contexte en présentant les outils, les méthodes existantes pour résoudre les différentes problématiques du projet existants dans ces champs d'étude et proposer notre solution qui est l'objectif de notre travail, ainsi que le besoin qui a donné naissance à ce projet. La deuxième partie est constitué de deux chapitres, le chapitre 1 d'intitulé « ANALYSE ET CONCEPTION », va nous permettre de donner une idée générale sur la conception du système et une phase d'analyse des outils existants avec une comparaison entre-autres, et puis un chapitre d'intitulé « RÉALISATION DU PROJET » qui donnera les principaux démarches de réalisation de notre système.

PARTIE 1

CHAPITRE 1 :

ÉTUDE GENERALE DU PROJET

INTRODUCTION

La première partie consiste à l'étude des outils existants dans le domaine de notre projet, nous présentons la problématique qui donne naissance à l'idée d'un système de reconnaissance faciale et nous proposons une solution qui est notre objectif.

1. 1. 1. Etude de l'existant

Ils existent différents travaux approchant notre sujet et parlant de l'identification des personnes par reconnaissance de visage, nous en citons :

- **Face Recognition** : C'est un projet d'une société multi nationalité nommée PACKT PUBLISHING de développement visuelle sous le nom " Mastering OpenCV with Practical Computer Vision Projects ", ce projet est le résultat des recherches faites sur les bibliothèques Open CV et ses applications dans le domaine de l'imagerie et la détection des objets.
- **Visual Control** : Un projet élaboré par un développeur en C++. L'auteur du présent travail aboutit au résultat qui est la conception d'une application de reconnaissance de visage dans une capture en temps réel avec une webcam.
- **Open Face** : est une implémentation Python et Torch de la reconnaissance faciale avec des réseaux neuronaux profonds et est basé sur le CVPR 2015 du FaceNet : A Unified Embedding for Face Recognition and Clustering par Florian Schroff, Dmitry Kalenichenko, et James Philbin à Google. Torch permet au réseau d'être exécuté sur un CPU ou avec CUDA. préparé par Brandon Amos dans le groupe de recherche de Satya à l'Université Carnegie Mellon.
- **Training System** : Il s'agit d'un projet de fin d'études élaboré par gihan tharanga d'école d'ingénieurs Rajalakshmi de Chennai à l'inde. Cet auteur présente dans ce travail un nouveau capteur atténuant l'illumination ambiante. Ce capteur se base sur la réduction de temps de pose pour ne capturer qu'une faible quantité de la lumière ambiante mais aussi facilite la délivrance en simultané d'une image en proche infrarouge et d'une image normale acquise avec la lumière ambiante, grâce à ce capteur et avec utilisation des points caractéristiques détectés et l'image de contours, un algorithme de reconnaissance de visage s'inspirant de l'elastic graph matching pour construire un modèle du visage a été mis au point. Quant à nous, avons tenu compte de l'environnement changeant et par conséquent avons converti toutes les images de figures détectées en niveau gris égalisé.
- **Facial Expression Recognition V2** : Une application payante qui se base sur le principe AdaBoost, développée et concevée par Yoav Freund et Robert Schapire, « A desicion-theoretic generalization of on-line learning and an application to boosting, Les expressions faciales transmettent les signaux non verbaux, qui jouent un rôle important dans les relations interpersonnelles. La reconnaissance automatique des expressions faciales peut être une composante importante des interfaces homme-machine naturelles, il peut également être utilisé dans les

sciences du comportement et dans la pratique clinique. Bien que les hommes reconnaissent les expressions faciales pratiquement sans effort ou de retard, la reconnaissance de l'expression fiable par la machine est toujours un défi. Ceci est un aperçu de haut niveau de la reconnaissance de l'expression automatique du projet s'il met en évidence les principaux composants du système et des défis de la recherche.

1. 1. 2. Problématique

De nos jours on parle de plus en plus de l'insécurité dans divers secteurs ainsi que des moyens informatiques à mettre en œuvre pour contrer cette tendance : le contrôle d'accès à nos ordinateurs, l'e-commerce, les opérations bancaires basées sur l'identification du demandeur, etc.

Parlant des opérations bancaires, une institution bancaire est amenée à effectuer ses activités quotidiennes en assurant des services à ses clients. Ces services rendus par une institution bancaire la conduisent à être en contact physique fréquent (à part quelques exceptions à l'exemple du commerce par internet) avec ses clients.

Les clients qui vont déposer, retirer des sommes d'argent, effectuer des virements bancaires, des transferts d'argent, etc. doivent se présenter physiquement dans cette institution afin d'être servis. Ceci fait que cette institution bancaire doit recevoir nombreuses gens de toute catégorie, de tout genre, de toute couche de la population par jour et encore plus par semaine, par mois, par an.

Les mesures de sécurité mises en place par nos institutions bancaires ne leur garantissent pas la protection totale contre les espions, contre les braqueurs, contre les escrocs. nous avons assisté à un braquage en pleine journée de l'une des institutions bancaires à Tanger. Ces braqueurs ne se sont pas inquiétés des mesures de sécurité utilisées par cette institution bancaire (comme tant d'autres institutions bancaires qui utilisent seulement les agents de sécurité en uniforme).

Ces auteurs, les braqueurs pouvaient être découverts (s'ils ne le sont pas encore bien sûr) le plus vite possible, si les mesures préventives de sécurité pouvaient être mises au point avant cet événement de braquage.

Une préoccupation nous vient en première vue :

- Comment nos institutions bancaires, en particulier peuvent-elles assurer le control, la surveillance, le suivi de toutes les visites dans ces dernières de personnes considérées comme clients pour afin garantir leur sécurité ?
- Mais aussi, même si elles arrivaient à contrôler toutes ces visites, quelle est le moyen efficace à utiliser pour mieux identifier toutes les personnes visitant ces institutions bancaires et ainsi assurer la sécurité individuelle à chacune de ces institutions ?

Dans un autre contexte, à présent chaque'un d'entre nous a besoin d'une clé pour ouvrir la porte de sa maison et une autre pour démarrer sa voiture, et d'un mot de passe ou une empreinte digitale pour passer un appel et utiliser son Smartphone. En effet il est difficile de maintenir tous ses outils avec nous, les exigences du futur peuvent être une sorte de

généralisation de nos clés et mots de passe, la solution est notre visage. Comment peut-on garantir l'accès à nos propriétés sans cassé nos têtes d'apprendre des mots de passe ou avoir des clés en poche tout le temps ?

Nous tacherons de répondre à ces questions tout au long de notre travail et ceci pourra même constituer la base de notre sujet.

1. 1. 3. - Solution / Objectif

Alors que l'informatique est devenue pour nous un outil incontournable de gestion, d'organisation, de production et de communication, les échanges internes et externes dans nos vies sont exposés aux actes de malveillance de différentes natures et sans cesse changeants.

D'une manière concrète pour lutter contre ces visites incontrôlées ou bien des clés et mots de passe inutiles, nous proposons comme hypothèses ce qui suit :

- La mise en place d'un système de contrôle des "personnes extérieures" (ici les clients) aux institutions bancaires peut les aider à assurer et à garantir leur sécurité. Ceci pourra être possible grâce à l'utilisation des cameras de surveillance placées à l'intérieur comme à l'extérieur, dans différents coins (dans toutes les positions) de bâtiment des institutions bancaires et à leurs différentes entrées pour accéder à l'intérieur de ces institutions.

Dans un autre contexte nous proposons l'intégration d'une caméra dans la portière de la maison, la porte du chauffeur du voiture ou même l'utilisation du camera frontale des Smartphone pour garantir l'accès facile et l'utilisation avec un maximum de sécurité.

Notre investissement dans le domaine de la reconnaissance de visages est sans doute motivé par la multiplicité et la variété des champs d'application entre-autres la haute sécurité, la télésurveillance ou encore le contrôle d'accès, etc.

Parmi ses applications les plus concrètes rencontrées dans la vie courante, nous pouvons citer à titre d'exemple la recherche des individus dans une foule par reconnaissance de leur visage par des caméras de surveillance, étude des émotions par analyse des expressions faciales, la protection d'identité des personnes pour livrer les cartes d'identités surs et fiables mais aussi de permis de conduire.

1. 1. 4. Planification

1. 1. 4. 1. Plan Prévisionnel :

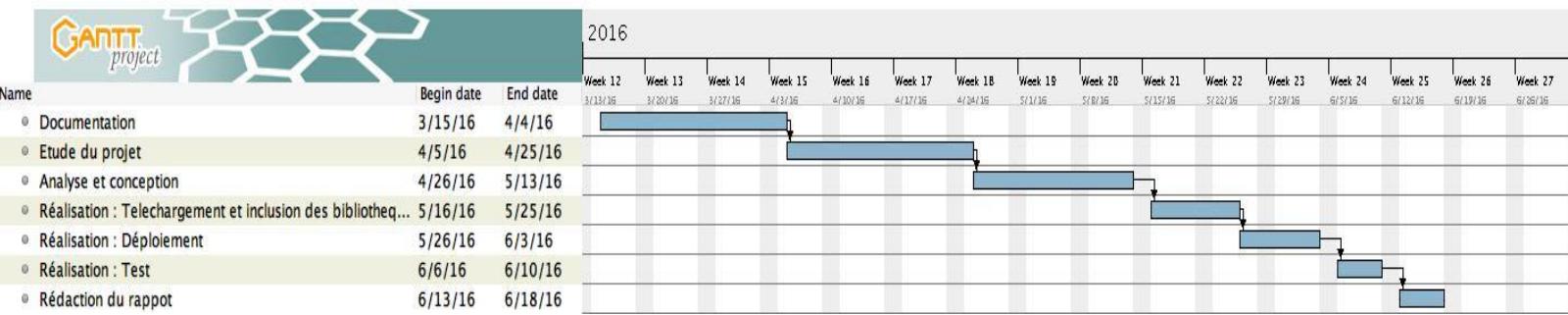


Figure 1 : Plan Prévisionnel

1. 1. 4. 2. Plan réel

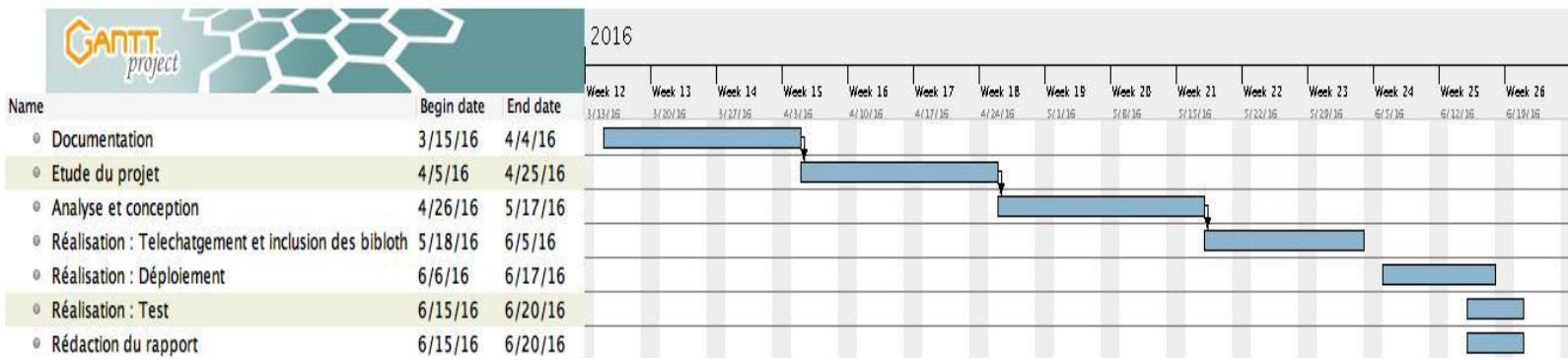


Figure 2 : Plan Réel

Lors des travaux réalisés en cours du projet, nous avons rencontré plusieurs difficultés dans différentes étapes.

Au niveau de la gestion de temps, nous avons eu des problèmes par rapport aux cours et aux mini-projets des autres modules ainsi que les examens, au niveau de l'installation du système, la préparation de l'environnement de l'application était un vrai défi qui a engendré des obstacles : Fichiers manquants, durée d'installation, erreurs de système.

CONCLUSION

Dans ce chapitre, nous avons présenté l'étude générale du projet et la problématique ainsi qu'une proposition de solution qui présentera notre objectif.

PARTIE 2

CHAPITRE 1 : ANALYSE ET CONCEPTION

INTRODUCTION

Cette partie est consacrée à la conception et la modélisation et les méthodes utilisées qui sont des étapes fondamentales dans l'étude de notre application. Nous avons choisi de modéliser en s'appuyant sur le formalisme UML qui offre une flexibilité marquante et s'exprime par l'utilisation des diagrammes.

2. 1. 1. Analyse et Conception

2. 1. 1. 1. Diagramme de cas d'utilisation

Les cas d'utilisations est un outil important d'UML. Il permet d'identifier les possibilités d'interaction entre le système et les acteurs, c'est-à-dire toutes les fonctionnalités que doit fournir le système, il permet aussi de délimiter le système.

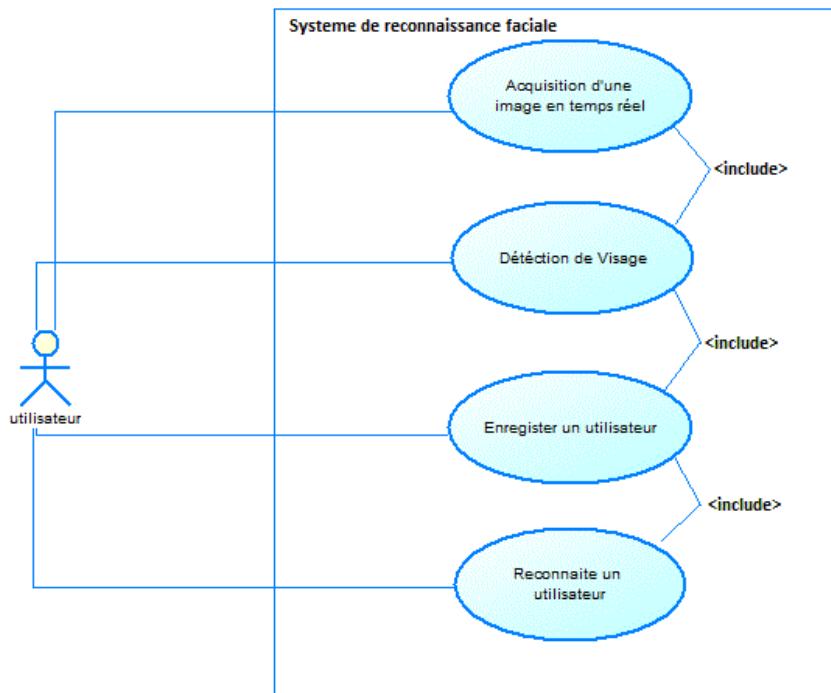


Figure 3 : Diagramme de cas d'utilisation

Le système permet d'accuser une image en temps réel, détecté un visage, enregistrer un utilisateur et reconnaître cet utilisateur.

2. 1. 1. 2. Diagramme de séquences

Le diagramme de séquence permet de cacher les interactions d'objet dans le cadre d'un scenario d'un diagramme des cas d'utilisations dans un souci de simplification, on représente l'acteur principal à gauche du diagramme et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroule les actions entre les acteurs ou objets.

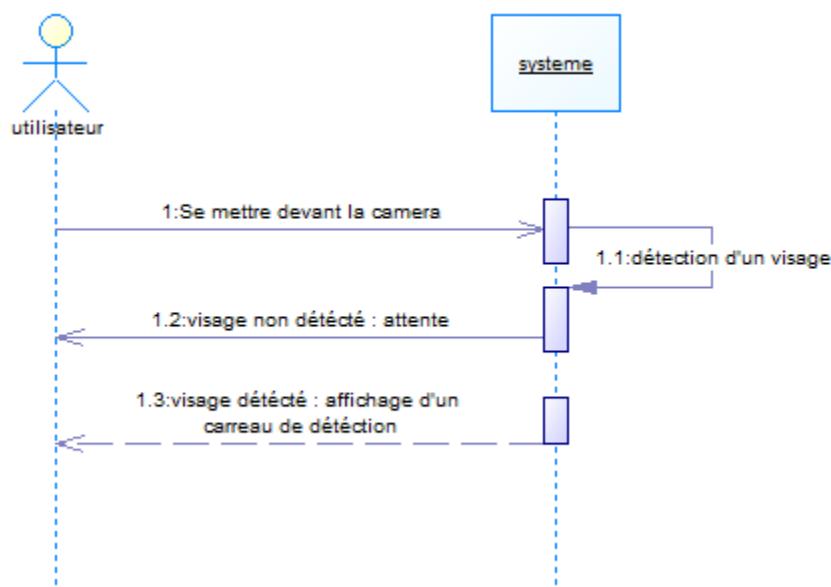


Figure 4 : Diagramme de séquence détection de visage

L'utilisateur se met devant la caméra pour permettre au système de détecter le visage et afficher les carreaux de la détection.

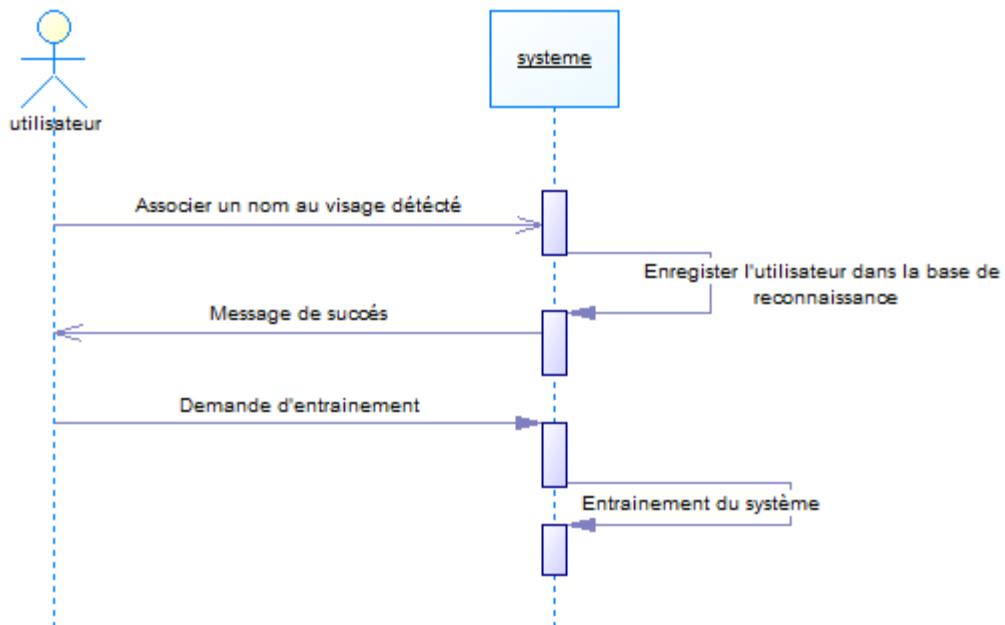


Figure 5 : Diagramme de séquence Enregistrement

L'utilisateur associe un nom pour chaque visage de la personne enregistrée dans la base de reconnaissance.

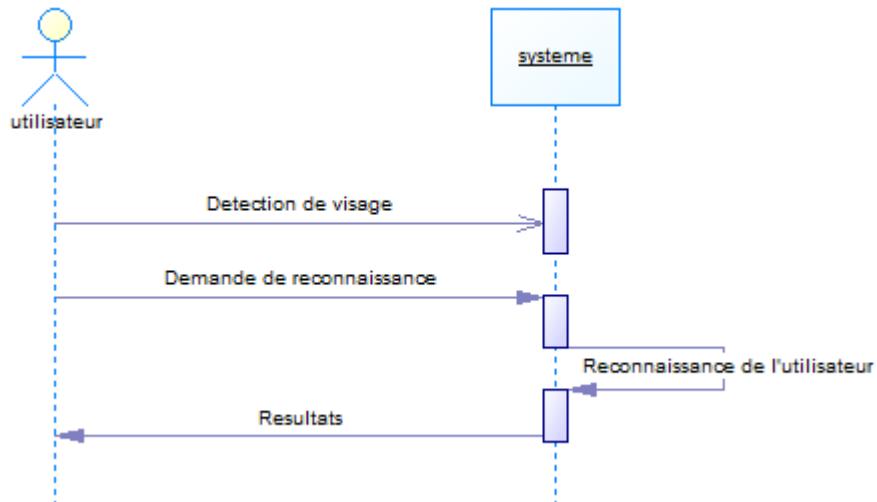


Figure 6 : Diagramme de séquence Reconnaissance

Après avoir été détecté, l'utilisateur demande au système de lui reconnaître, le système réagi et donne les résultats de la reconnaissance.

2. 1. 2. Architecture du système

Le système de reconnaissance faciale se base d'abord sur la détection de visage dans une vidéo en temps réel, plusieurs étapes viennent après : Normalisation du visage, Extraction des caractéristiques et la comparaison de ces caractéristiques avec ceux qui existent dans notre base de données, avant de produire le résultat : Identité du Visage.

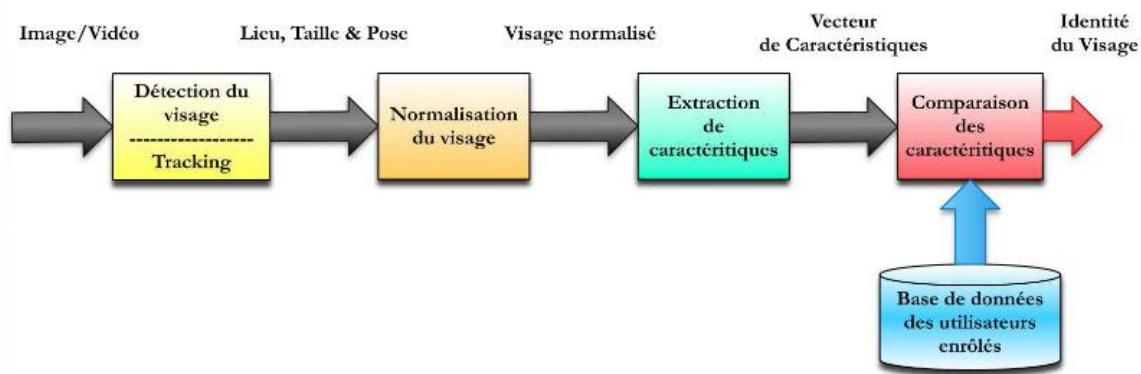


Figure 7 : Conception du système

2. 1. 3. Méthodes utilisées

Les systèmes de reconnaissance faciale se basent en général sur les étapes suivantes :

1. L'acquisition de l'image
2. La détection du Visage (Algorithme de Viola et Jones)
3. L'Alignement du Visage (Calcule de l'angle du Visage avec la Camera)
4. L'Extraction des éléments importants du Visage (Algorithme du PCA)
5. Reconnaissance (Comparaison avec les données dans la base de reconnaissance)
6. Résultat.

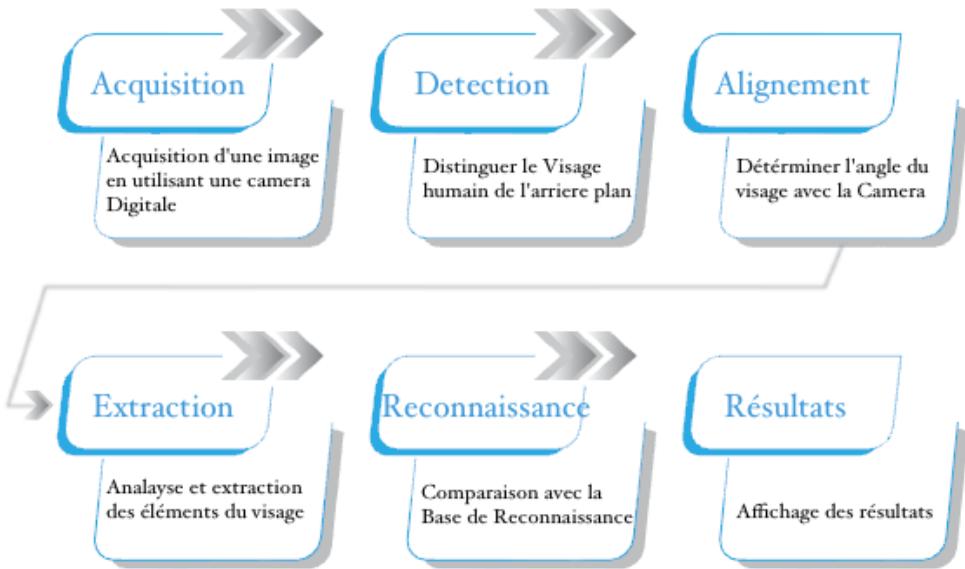


Figure 8 : Algorithme de l'application

Ces étapes diffèrent de chaque système et de son but d'utilisation.

Un système de reconnaissance faciale nous permet de reconnaître une personne qui est déjà existante dans notre base de données, c'est clair que ça sera une comparaison, une étude dans le domaine de l'imagerie et un travail purement mathématique et algébrique en ce qui concerne le calcul matriciel et les espaces vectoriels.

Il faut d'abord construire une base de reconnaissance, en anglais “ Training Set “ c'est un entraînement de notre système pour reconnaître un visage avec un ensemble d'images permettant au système d'avoir plusieurs informations et de calculer des valeurs ressortissant de la personne ciblé, des vecteurs de caractéristiques dont on va éclaircir plus tard, ici dans ce système, nous voulons connaître l'identité de la personne, à travers l'image de cette personne passer au système, appelé dans ce cas cette image fournie pour l'apprentissage nommé "test image".

Afin d'attirer les vecteur des caractéristiques des images dans ce projet, nous nous appuierons sur l'algorithme de PCA, dans cet article on va essayer de donner une idée sur l'utilisation de l'algorithme PCA (Principale Component Analysis).

2. 1. 3. 1. Acquisition de l'image

L'utilisation d'une caméra intégrée permettra d'avoir une image en temps réel pour assurer le fonctionnement.

2. 1. 3. 2. Détection du visage : Algorithme / Méthode de Viola et Jones

Pour cet algorithme, On ne va pas beaucoup rentrer dans ces détails, c'est un sujet de recherche qui est loin de notre objectif, la méthode de Viola et Jones est une méthode de détection d'objet dans une image numérique, proposée par les chercheurs Paul Viola et Michael Jones en 2001.

Elle fait partie des toutes premières méthodes capables de détecter efficacement et en temps réel des objets dans une image. Inventée à l'origine pour détecter des visages, elle peut également être utilisée pour détecter d'autres types d'objets comme des voitures ou des avions. La méthode de Viola et Jones est l'une des méthodes les plus connues et les plus utilisées, en particulier pour la détection de visages et la détection de personnes.

En tant que procédé d'apprentissage supervisé, la méthode de Viola et Jones nécessite de quelques centaines à plusieurs milliers d'exemples de l'objet que l'on souhaite détecter, pour entraîner un classifieur. Une fois son apprentissage réalisé, ce classifieur est utilisé pour détecter la présence éventuelle de l'objet dans une image en parcourant celle-ci de manière exhaustive, à toutes les positions et dans toutes les tailles possibles.

Considérée comme étant l'une des plus importantes méthodes de détection d'objet, la méthode de Viola et Jones est notamment connue pour avoir introduit plusieurs notions reprises ensuite par de nombreux chercheurs en vision par ordinateur, à l'exemple de la notion d'image intégrale ou de la méthode de classification construite comme une cascade de classificateurs boostés.

Cette méthode bénéficie d'une implémentation sous licence BSD dans OpenCV, une bibliothèque très utilisée en vision par ordinateur.

2. 1. 3. 3. Alignement / Extraction : Algorithme du PCA (Approche mathématique)

Introduction

Cet Algorithme est l'une des techniques les plus efficaces qui ont été utilisés dans le domaine de la reconnaissance d'image et dans le domaine de la compression d'image.

PCA est classée comme l'une des méthodes statistiques dans ce domaine.

L'objectif principal dans l'algorithme de PCA consiste à réduire les grandes dimensions des données spatiales aux dimensions les plus petits espaces. Habituellement, les nouveaux espaces sont des espaces caractéristiques (contenant les caractéristiques fondamentales et importantes des données dans l'espace original) et par conséquent, en réduisant les dimensions de cela, nous avons décrit les données de façon plus économique qui nous aidera plus tard.

Les problèmes apparaissent généralement dans les systèmes de reconnaissance de visage lorsqu'ils traitent des grands espaces de systèmes-dimensionnelles (par exemple, le traitement des images).

On peut faire de nombreuses améliorations et à travers la mise en correspondance et de transférer les données existantes pour les données d'un espaces de moins de dimensions. Ainsi, nous avons, nous diminuons les dimensions de l'espace d'origine avec de grandes dimensions au nouvel espace avec de plus petites dimensions.

Parlons Matrices maintenant

Que ce soit, par exemple, nous avons le vecteur

$$\mathbf{x} = [x_1, x_2, \dots, x_N]^T$$

dans un espace de N dimension, nous réduisons les dimensions en se déplaçant à un espace

$$\mathbf{y} = [y_1, y_2, \dots, y_K]^T$$

Constitué de K dimensions de sorte que $K < N$.

En principe, la diminution dimensionnelle à son tour conduit à la perte d'information, mais le but de l'algorithme du PCA est de réduire les dimensions des données tout en conservant autant que possible l'information contenue dans les données originales, dans ce contexte, le PCA calcule la transformation linéaire, ce qui conserve les données contenues dans l'espace de grandes dimensions dans un sous-espace moins-dimensionnel, comme la position ci-dessous :

$$\begin{cases} y_1 = t_{11}x_1 + t_{12}x_2 + \cdots + t_{1N}x_N \\ y_2 = t_{21}x_1 + t_{22}x_2 + \cdots + t_{2N}x_N \\ \cdots \\ y_K = t_{K1}x_1 + t_{K2}x_2 + \cdots + t_{KN}x_N \end{cases}$$

$$\mathbf{y} = \mathbf{T}\mathbf{x}$$

d'autre façon :

avec

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1N} \\ t_{21} & t_{22} & \cdots & t_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ t_{K1} & t_{K2} & \cdots & t_{KN} \end{bmatrix}$$

La transformation \mathbf{T} est la transformation dans laquelle la valeur $\|\mathbf{x} - \mathbf{y}\|$ est minimale.

Selon la théorie du PCA, on peut définir l'espace de moins de dimension K en tirant profit des vecteurs propres de la matrice de covariance et de variation.

Je m'explique de plus, supposons que

$$\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_M$$

Est un ensemble de M vecteurs, chaque vecteur est de dimension N x 1.

Ci-dessous, nous avons mentionné et expliqué les étapes de base pour l'algorithme de PCA :

étape 1 :

On calcule le vecteur central des vecteurs donnés

$$\bar{\mathbf{I}} = \frac{1}{M} \sum_{i=1}^M \mathbf{I}_i$$

étape 2 :

On fait une normalisation de chaque vecteur, en faisant une soustraction de ce vecteur et du vecteur centrale calculé dans l'étape 1

$$\Phi_i = \mathbf{I}_i - \bar{\mathbf{I}}$$

étape 3 :

La construction du matrice des dimensions N x M tel que :

$$\mathbf{A} = [\Phi_1, \Phi_2, \dots, \Phi_M]$$

étape 4 :

On calcule la matrice de covariance associée à l'ensemble des vecteurs, la matrice est la suivante :

$$\mathbf{C} = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = \mathbf{A} \mathbf{A}^T$$

c'est une matrice de dimensions N x N.

étape 5 :

On calcule les valeurs propres et les vecteurs propres de la matrice C :

$$\lambda_1, \lambda_2, \dots, \lambda_N \text{ et } \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N$$

Sachant que la matrice C est une matrice symétrique, donc elle est constituée des vecteur de la base, donc le vecteur I est dans le même espace d'où on peut le représenter comme une combinaison linéaire des vecteurs de cette base tout en utilisant les vecteur normalisé dans l'étape 2, alors on aura :

$$\mathbf{I} - \bar{\mathbf{I}} = y_1 \mathbf{u}_1 + y_2 \mathbf{u}_2 + \dots + y_N \mathbf{u}_N = \sum_{i=1}^N y_i \mathbf{u}_i$$

étape 6 (réduction des dimensions) :

Nous allons dans cette étape représenter chaque vecteur I en conservant seulement les valeurs correspondant à la plus grande valeur significatif K :

$$\hat{\mathbf{I}} - \bar{\mathbf{I}} = y_1 \mathbf{u}_1 + y_2 \mathbf{u}_2 + \cdots + y_K \mathbf{u}_K = \sum_{i=1}^K y_i \mathbf{u}_i$$

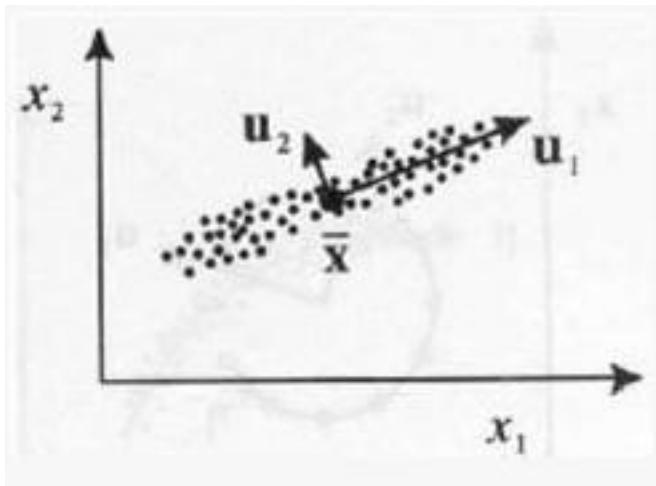
Avec biensur $K < N$, dans ce cas $\hat{\mathbf{I}}$ est proche de \mathbf{I} tel que la valeur $\|\mathbf{I} - \hat{\mathbf{I}}\|$ est minimale, pour cela, la transformation linéaire \mathbf{T} de l'algorithme du PCA est reconnu pas les composante principale du matrice de covariance.

$$\mathbf{T} = \begin{bmatrix} u_{11} & u_{21} & \cdots & u_{K1} \\ u_{12} & u_{22} & \cdots & u_{K2} \\ \vdots & \vdots & \ddots & \vdots \\ u_{1N} & u_{2N} & \cdots & u_{KN} \end{bmatrix}$$

Nous allons expliquer et représenter l'algorithme PCA géométriquement afin que nous puissions mieux comprendre le mécanisme.

Le PCA baisse les données le long des directions dans lesquelles les données sont différentes que d'autres, ces tendances sont identifiées grâce aux vecteurs propres de la matrice de covariance aux plus grandes valeurs propres.

L'ampleur de la taille et l'importance des valeurs propres est compatible avec la variation des données le long des vecteurs propres.



Pour décider ce qui est le nombre de composants de base que nous avons besoin de garder (valeur de K moyenne), vous pouvez utiliser le critère suivant :

$$\frac{\sum_{i=1}^K \lambda_i}{\sum_{i=1}^N \lambda_i} > t$$

Où t est un seuil "Threshold" (par exemple ça prend les valeurs de 0,8 ou 0,9), Et la valeur de t détermine la quantité d'information qui sera conservée dans les données, une fois on détermine la valeur de t, on peut alors déterminer la valeur de K.

Nous pouvons dire que l'erreur de qui résulte en réduisant les dimensions est donné par :

$$error = \frac{1}{2} \sum_{i=K+1}^N \lambda_i$$

Dans tout ce qui précède-on a expliqué le principe de fonctionnement de l'algorithme PCA, là en ce qui suit, on va introduire l'utilisation de cette méthode dans la reconnaissance faciale dans des images.

2. 1. 3. 4. Algorithme Eigenface (Application du PCA)

Si vous avez bien saisi ce qu'on a dit a propos du PCA, Je vais résumer ce que je vais expliquer a propos du "Eigenface", Le "Eigenface" est l'approche du PCA qui utilise un algorithme pour représenter les visages dans le sous-espace de faible dimension, et les extraire à travers cet espace pour utiliser les vecteurs propres, les matrices de covariance

des images de visage. Malgré que la voie en termes de l'approche est la même, mais il y a des problématiques pratiques qui doivent être étudiés et des considérations et des situations très particulières.

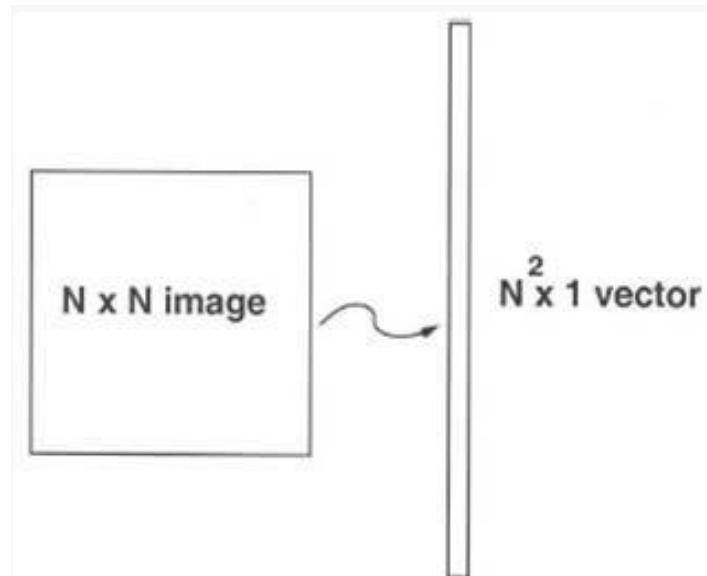
Supposons que nous avions un groupe de M - visage d'entraînement de dimensions NxN :

$$\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_M$$

Ci-dessous, nous avons cité les étapes de base nécessaires pour mettre en œuvre l'algorithme PCA sur un groupe de visage d'entraînement :

étape 1 :

Il faut convertir chaque image \mathbf{I}_i des dimensions NxN en un seul vecteur Γ_i des dimensions $N^2 \times 1$. Ce processus peut être réalisé simplement par ranger les lignes un par un, pour tourner à partir d'une matrice à un seul faisceau comme dans la figure suivante :



étape 2 :

On calcule le vecteur centrale par la relation suivante :

$$\bar{\Psi} = \frac{1}{M} \sum_{i=1}^M \Gamma_i$$

étape 3 :

Dans cet étape on fait une normalisation de chaque vecteur Γ_i comme on a déjà expliqué en faisant une soustraction au vecteur centrale :

$$\Phi_i = \Gamma_i - \Psi$$

étape 4 :

On construit la matrice $\mathbf{A} = [\Phi_1, \Phi_2, \dots, \Phi_M]$ de dimension $N2 \times M$.

étape 5:

La matrice de covariance qui contient les variations visages est calculé selon la méthode de PCA, nous avons besoin de calculer les vecteurs propres U_i de la matrice AAT , mais cette matrice est très grande (de dimensions $N2 \times N2$), par conséquent, il est impossible de calculer ses vecteur propres. Au lieu de cela, nous allons tenir compte les vecteur propres V_i de la matrice ATA qui à son tour est beaucoup plus petite que notre matrice, (c.-à-dimensions $M \times M$).

étape 6 :

On peut montrer facilement la relation entre U_i et V_i , tant que V_i sont les vecteurs propres de la matrice ATA , donc elle satisfait à la relation suivante :

$$\mathbf{A}^T \mathbf{A} \mathbf{v}_i = \mu_i \mathbf{v}_i$$

tel que μ_i représente les valeurs propres appropriés .

Donc si on multiplie les deux parties de l'équation par la matrice A on aura :

$$\mathbf{A} \mathbf{A}^T \mathbf{A} \mathbf{v}_i = \mathbf{A} \mu_i \mathbf{v}_i$$

$$\mathbf{C} \mathbf{A} \mathbf{v}_i = \mathbf{A} \mu_i \mathbf{v}_i$$

$$\mathbf{C} \mathbf{u}_i = \mu_i \mathbf{u}_i$$

Ensuite, la valeur $\mathbf{u}_i = \mathbf{A}\mathbf{v}_i$ est la valeur propre des deux matrices ATA et AAT

Il faut qu'on remarque que la matrice AAT peut avoir presque N2 vecteur propres, alors que la matrice ATA n'aura que M vecteurs propres.

étape 7 :

On calcule les vecteurs propres \mathbf{U}_i de la matrice AAT en utilisant la relation :

$$\mathbf{u}_i = \mathbf{A}\mathbf{v}_i$$

remarque : il faut normaliser les \mathbf{U}_i tel que $\|\mathbf{U}_i\| = 1$.

étape 8 (réduction des dimensions) :

On représente chaque visage Γ en conservant uniquement les valeurs qui correspondent au plus grand K valeur propres :

$$\hat{\Gamma} - \Psi = y_1 \mathbf{u}_1 + y_2 \mathbf{u}_2 + \cdots + y_K \mathbf{u}_K = \sum_{i=1}^K y_i \mathbf{u}_i$$

L'image ci-dessous nous donne une approche simulée à la méthode "Eigenface".

Dans la première rangée, on remarque plusieurs visage "Eigenface" (autrement dit, les images qui correspondent aux plus grandes valeurs propres), le mot "Eigenface" vient du fait que les images apparaissent comme des photos fantomiques.

Dans la deuxième rangée, on remarque des autres visages qui sont une combinaison linéaire du visage cible.

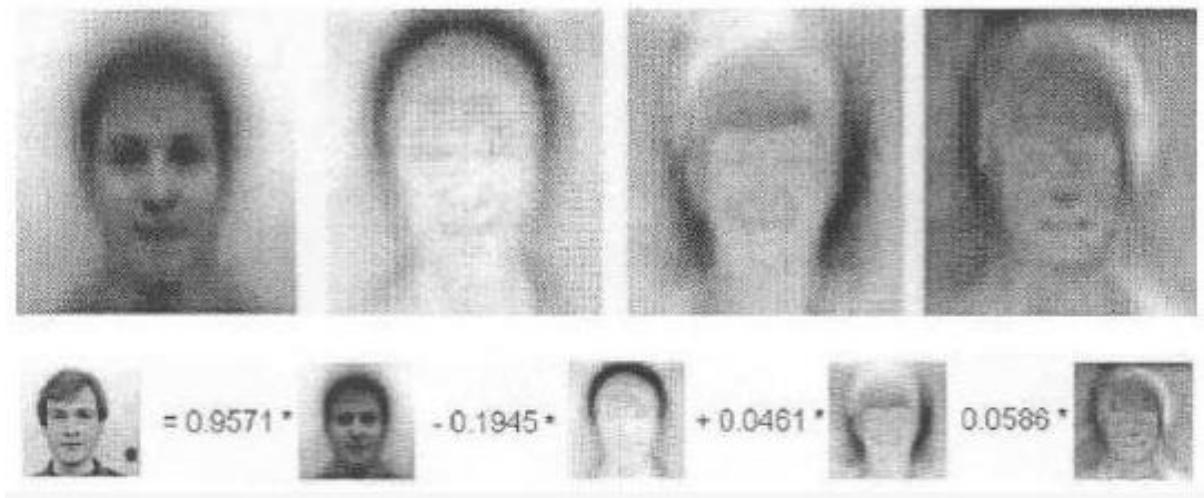


Figure 9 : Les Eigenfaces

En se basant sur le PCA, chaque visage Γ peut être représenté dans un espace de dimensions inférieures aux dimensions de l'image originale, en utilisant des coefficients de dilatation longitudinales:

$$\Omega = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{bmatrix}$$

2. 1. 3. 5. Reconnaissance et Résultats

Pour compléter la tâche de la reconnaissance des visages, tout d'abord, nous devons représenter tous les visages du groupe d'entraînement au sein de l'espace de moins de dimensions en utilisant l'algorithme de PCA.

$$\Omega_i = \begin{bmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{iK} \end{bmatrix}, i = 1, \dots, M$$

Supposons que le visage cible à reconnaître est I des dimensions $N \times N$ (l'alignement de cette visage de la même manière de l'alignement et la normalisation des images d'entraînement définies)

On applique les étapes suivantes pour faire la reconnaissance de ce visage :

étape 1 :

On représente l'image I comme un seul vecteur de dimension $N^2 \times 1$ et en le nommant Γ

étape 2 :

Il faut normaliser le vecteur Γ en faisant la soustraction suivante :

$$\Phi = \Gamma - \Psi$$

Ψ étant la valeur centrale du vecteur.

étape 3 :

La projection de Φ dans l'espace de PCA (c.-à-d. dans l'Eigenspace) de la manière suivante :

$$\Phi = y_1 \mathbf{u}_1 + y_2 \mathbf{u}_2 + \cdots + y_K \mathbf{u}_K = \sum_{i=1}^K y_i \mathbf{u}_i$$

avec $y_i = \mathbf{u}_i^T \Phi$

étape 4 :

on trouve le visage Φ_i le plus proche du visage d'entraînement dans la base de reconnaissance Φ , de tel façon que l'erreur suivant :

$$e_r = \min_l \|\Omega - \Omega_l\|$$

sera minimale.

étape 5 :

Si $e_r < T_r$ avec T_r est un seuil, donc le visage Γ à été identifié comme étant le visage Γ_i .
 e_r appelée erreur à " la distance dans l'espace de visages".

On utilise habituellement la distance euclidienne pour calculer l'erreur, en tout état de cause, il a été démontré que l'utilisation d'autres moyens de calculer la distance comme "la distance de Mahalanobis" montrent de meilleurs résultats, comme indiqué ci-dessous :

$$\|\boldsymbol{\Omega} - \boldsymbol{\Omega}_l\| = \sum_{i=1}^K \frac{1}{\lambda_i} (y_i - y_{li})^2$$

Nous avons terminé l'explication d'une des méthodes modernes et efficaces utilisés dans la reconnaissance faciale.

2. 1. 4. Outils existants

Nous avons brièvement présenté chaque outil existant dans la partie de l'étude de l'existant. Il est intéressant de présenter en détail les outils afin de renforcer la comparaison.

2. 1. 4. 1. Face Recognition

Il s'agit d'une application qui se présente comme trois buttons en haut à gauche, une qui permet d'enregistrer un nouveau visage qui apparaît dans la liste des visages reconnus dans la barre à droite, l'application fournit le taux de ressemblance entre le visage détecté dans le temps réel (carré jaune) et visage dans la base de reconnaissance.

au dessous en vert le mode de fonctionnement de l'application, ils existent 2 modes : ADDING PERSON (Ajouter personne), RECOGNITION (Reconnaissance).

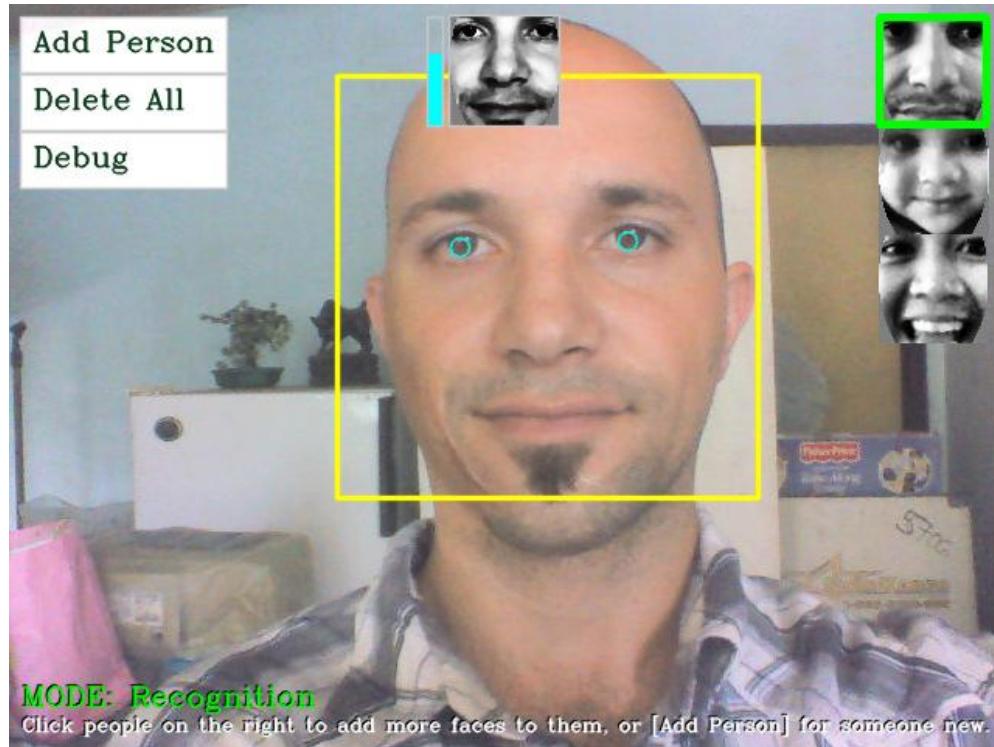


Figure 10 : Face recognition

2. 1. 4. 2. Visual Control :

Une belle interface graphique, facile à utiliser contient une case de commentaire au dessous, donnant une idée sur le fonctionnement de l'application l'angle du visage par rapport au caméra de détection, et la reconnaissance des personnes connu.

2. 1. 4. 3. Open Face :

Cette recherche a été soutenue par la National Science Foundation (NSF) sous le numéro de subvention CNS-1518865. Soutenu par les sociétés : Intel, Google, Vodafone, NVIDIA, et le fonds de la famille Conklin Kistler, c'est une application qui s'intègre au niveau des autres applications ou sitewebs, codé en C++ et Python avec un code relativement complexe.

2. 1. 4. 4. Training System :

Sans aucune interface graphique, cette application est conçue d'une manière à détecter le visage et afficher le nom si la personne est reconnue. La difficulté dans cette application est qu'il n'y a aucune méthode intégré pour reconnaître une nouvelle personne, sauf en passant par le code, changer le mode de fonctionnement.

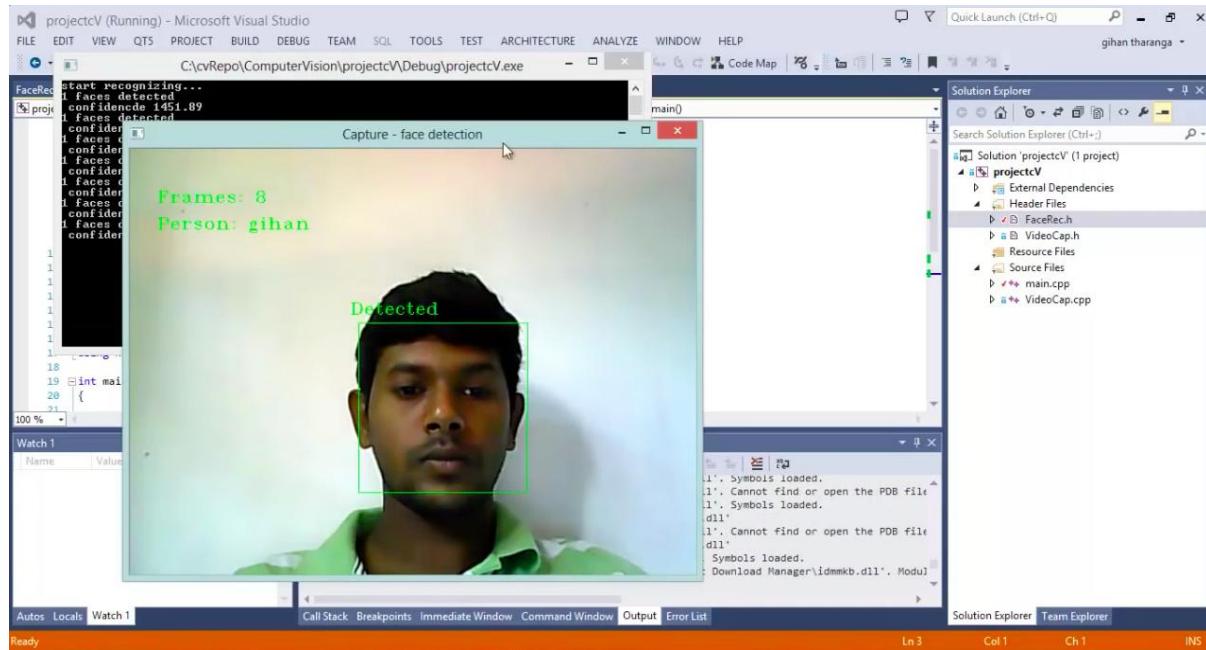


Figure 11 : Training System

2. 1. 4. 5. Facial Expression:

Concevée par Yoav Freund et Robert Schapire, cet application utilise Matlab et le même principe du Eigenface, l'avantage de ce projet est qu'il permet de reconnaître l'humeur de la personne détecté après avoir donné son identité, c'est une application complète très riche au niveau de programmation et de l'interface graphique, le problème qui existe : c'est payant.

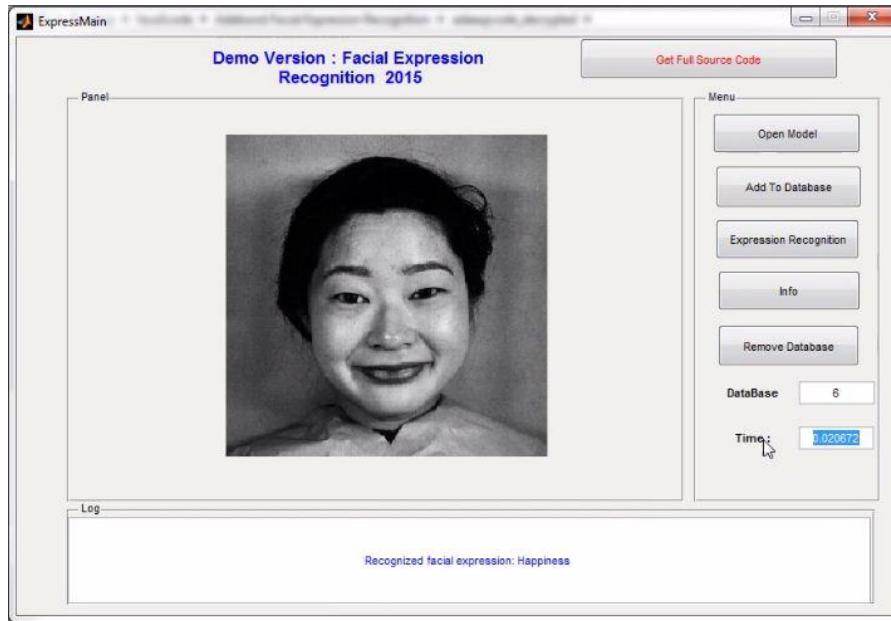


Figure 12 : Facial Expression

2. 1. 5. Comparaison et prise de décision

Pour choisir l'outil adéquat pour notre projet, nous avons réalisé une comparaison que nous présentons sous forme de tableau des caractéristiques permettant de récapituler l'ensemble des spécificités de chaque outil :

- Algorithme : L'algorithme utilisé par l'application
- Complexité : La complexité du réalisation de l'outil
- Interface Graphique : l'interface utilisée par l'application
- Détection d'humeur : La possibilité de détecté l'humeur de la personne (heureux, malheureux, neutre)
- Bibliothèques utilisées : La bibliothèque contenant les fonctions de la reconnaissance.
- Accessibilité : L'utilisation par rapport aux utilisateurs.
- Environnement : L'environnement dans lequel l'application fonctionne

Outils \ Critères	Algorithme	La Complexité	L'interface Graphique	Langage Utilisé	Detection d'humeur	Biblioth Utilisé	Accessibilité	Environnement
Face Recognition	Eigenface	Moins complexe	N'est pas bien conçu	C++	Non inclus	Open CV	Difficile à utiliser	Desktop
Visual Control	Eigenface	N'est pas complexe	Bien conçu	C++	Possible	Open CV	Clair et facile	Desktop
Open Face	Eigenface	complexe	intgré aux applications	Python et C++	Possible	Open CV	Difficile	WEB
Training System	Eigenface	N'est pas complexe	Simple	C++	Non inclus	Open CV	Facile	Desktop
Facial Expression	Eigenface	complexe	Bien conçu	Java	Possible	Matlab	Facile	Desktop

Figure 13 : Tableau de comparaison des outils

Le choix de l'outil qui va être l'objet de notre projet se base sur les critères de comparaisons qu'on a cité dans le tableau de la figure 11, on remarque que le projet d'intitulé " Visual Control " a plus de points forts (coloré en bleu) que les autres au niveau du complexité du code et l'interface graphique ainsi que l'accessibilité aux utilisateurs et la possibilité d'élargir l'utilisation du logiciel pour permettre en plus de la reconnaissance des personnes, la détection de leurs humeurs.

CONCLUSION

Nous avons terminé l'explication d'une des méthodes modernes et efficaces utilisés dans la reconnaissance faciale, et nous avons pris notre décision de l'outil adopté pour la réalisation de notre projet.

PARTIE 2

CHAPITRE 2 : RÉALISATION DU PROJET

INTRODUCTION

Après avoir achevé l'étape de la conception et de la modélisation de l'application, nous allons entamer dans ce chapitre la réalisation qui constitue le dernier volet de ce rapport et qui a pour objectif d'exposer le travail réalisé.

2. 2. 1. Inclusion des bibliothèques

L'implémentation de notre application nécessite une préparation de l'environnement du système, on a besoin de plusieurs bibliothèques :

2. Boost : est une collection de bibliothèques logicielles utilisées en programmation C++, contenant plusieurs fonctions mathématique utile pour notre travaille
3. TBB (Threading Building Blocks) : est une bibliothèque logicielle développée par la société Intel, Cet outil développé en C++ permet d'abstraire au maximum les détails complexes de la programmation sur microprocesseur multi cœur. Ainsi un développeur n'a plus à se soucier d'écrire son code pour les threads (processus léger) POSIX ou pour les threads Windows car c'est TBB qui s'occupe de tous les détails spécifiques. La bibliothèque TBB fonctionne sur différents compilateurs (Intel, Microsoft et GCC) et se veut donc parfaitement indépendante par rapport à l'architecture sous-jacente, comme par rapport à l'environnement logiciel.
4. OpenCV (Open Computer Vision) : est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel, met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques.
NB: OpenCV doit ne s'installe qu'avec un logiciel Cmake version 2.8.2.

Avant commencer, il faut télécharger les bibliothèques cité en dessous et les mettre dans un dossier "Libraries" dans le disque locale C.

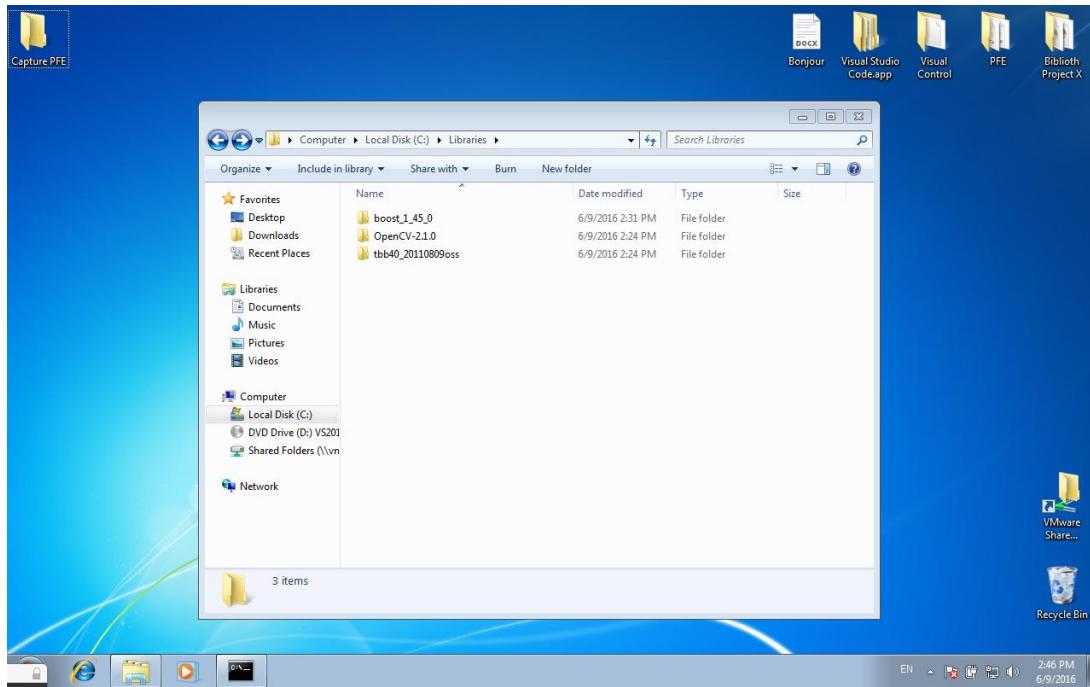


Figure 14 : Listes des bibliothèques

I / installation de Boost :

On démarre le " Visual Studio command prompt " pour commencer la configuration du Boost d'ailleurs on utilise la version 1.45.0, On va au dossier contenant les fichiers du bibliothèque " C:\Libraries\boost_1_45_0 "

On exécute les commandes suivantes dans l'ordre :

1. bootstrap (permet la creation du fichier executable bjam.exe)
2. bjam variant=debug,release threading=multi link=static

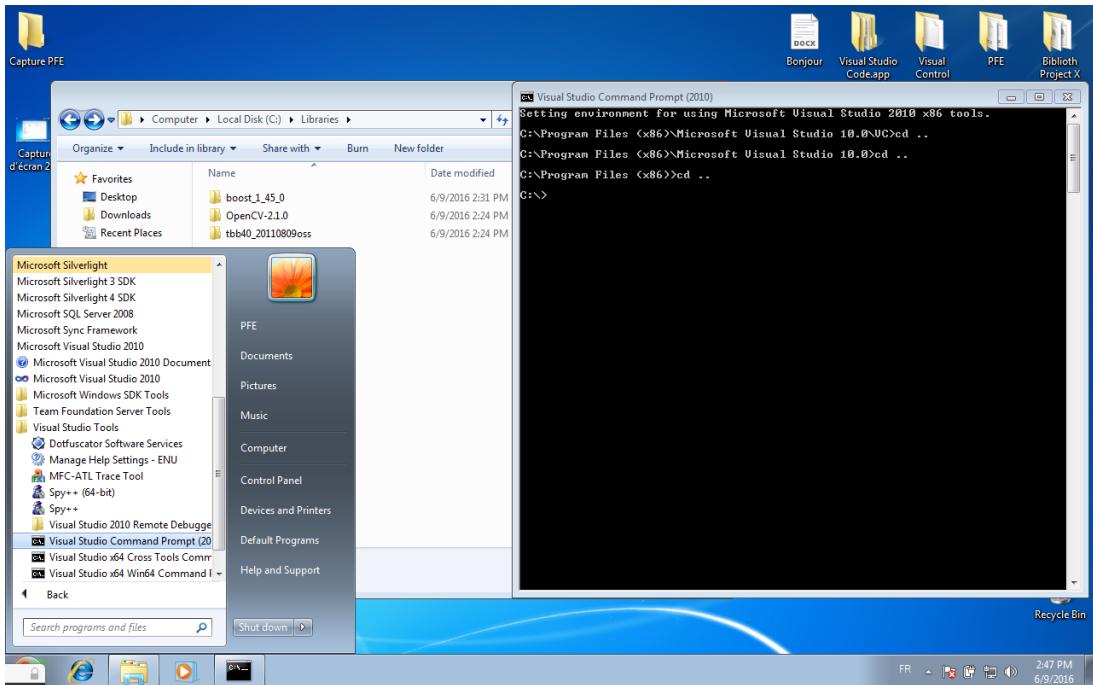


Figure 15 : Exécution des commandes

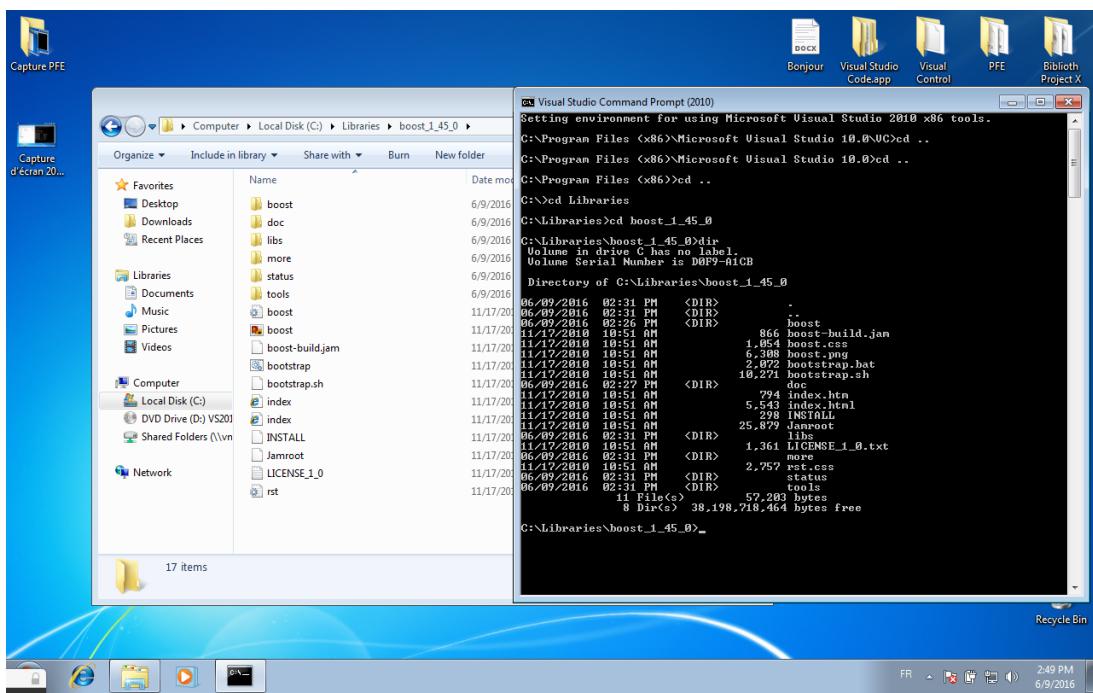


Figure 16 : Exécution des commandes 2

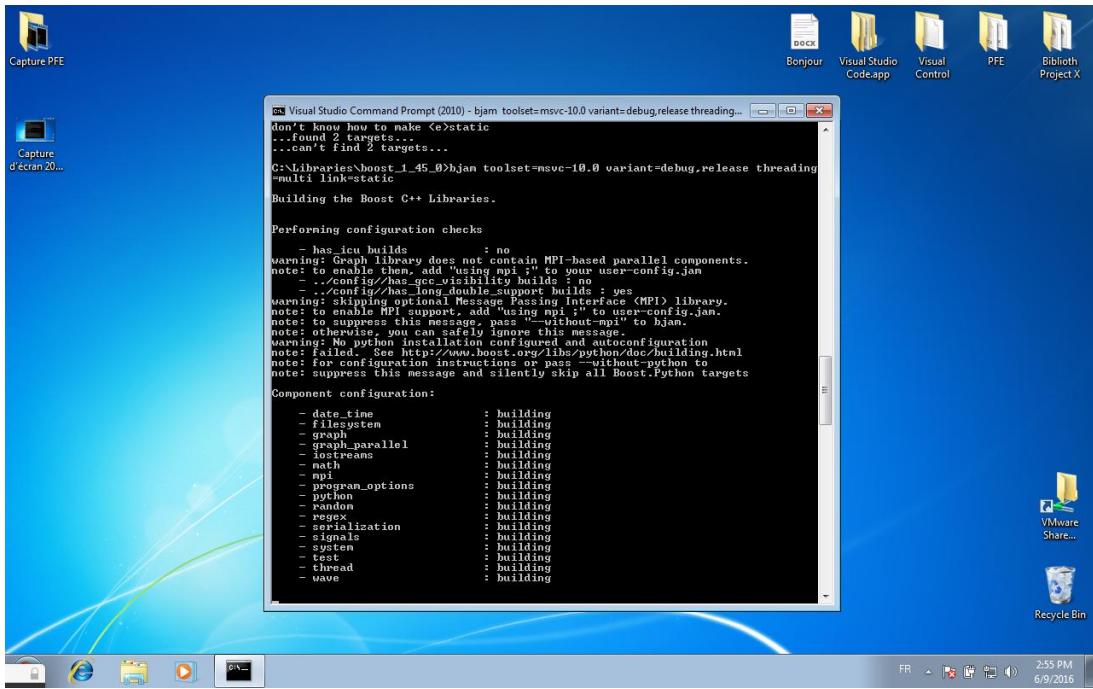


Figure 17 : Exécution des commandes 3

Après quelques minutes, la configuration est faite et les fichiers sont posé aux dossier qui convenable.

Il faut aller aux réglages du Visual Studio pour lui permettre de détecter ces fichiers dans les pages des propriétés

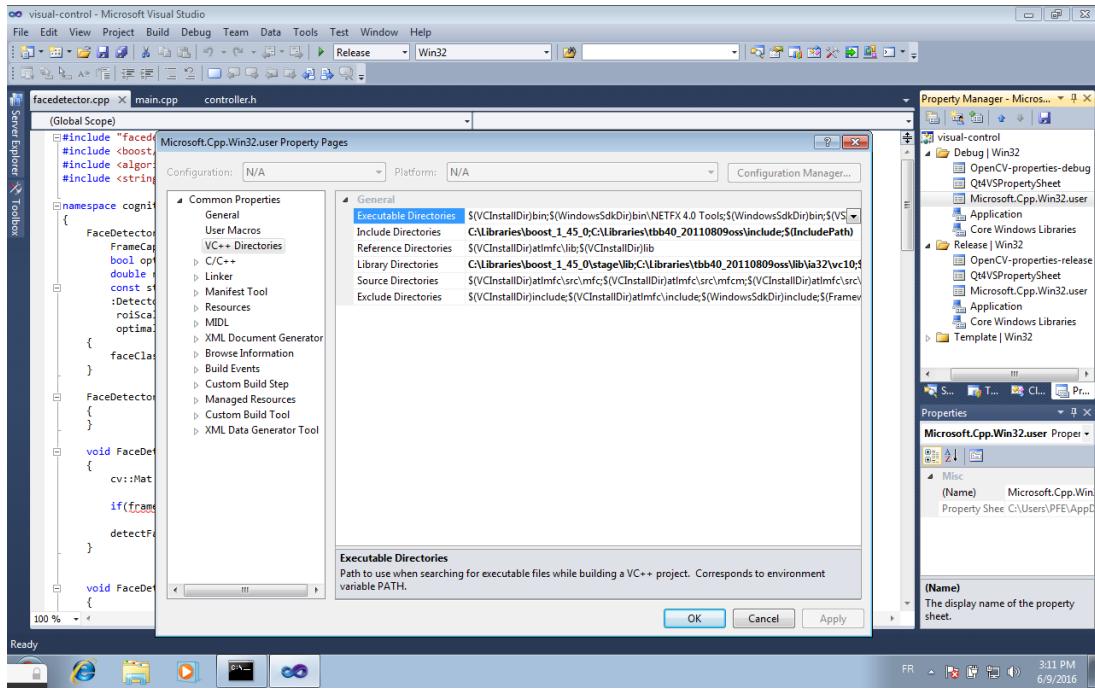


Figure 18 : Installation des bibliothèques sur Visual Studio

La bibliothèque Boost est prête, on télécharge le Cmake 2.8.2 qui permet d'installer la bibliothèque OpenCV dans l'emplacement "C:/Libraries/OpenCV-2.1.0/build" et le Threading Building Blocks (TBB 4.0) au niveau du système de Visual Studio, le Cmake indique que l'installation est faite sans TBB, alors il faut l'inclure dans la prochaine étape

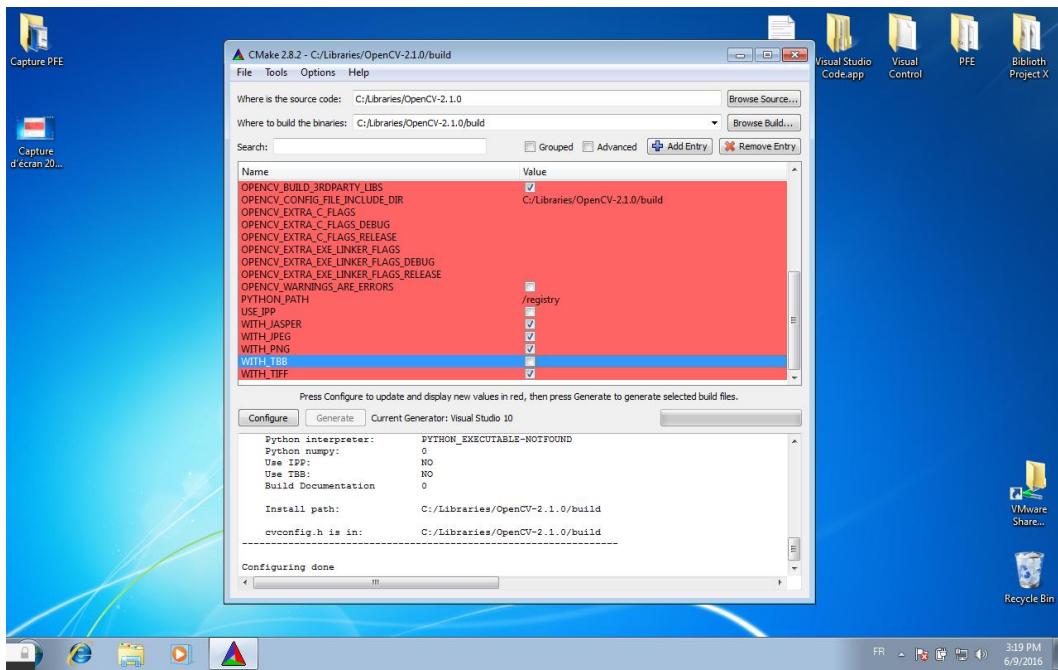


Figure 19 : Installation de Open CV et TBB avec Cmake

Après avoir indiqué l'emplacement du TBB dans notre disque locale le Cmake mentionne " Use TBB : YES "

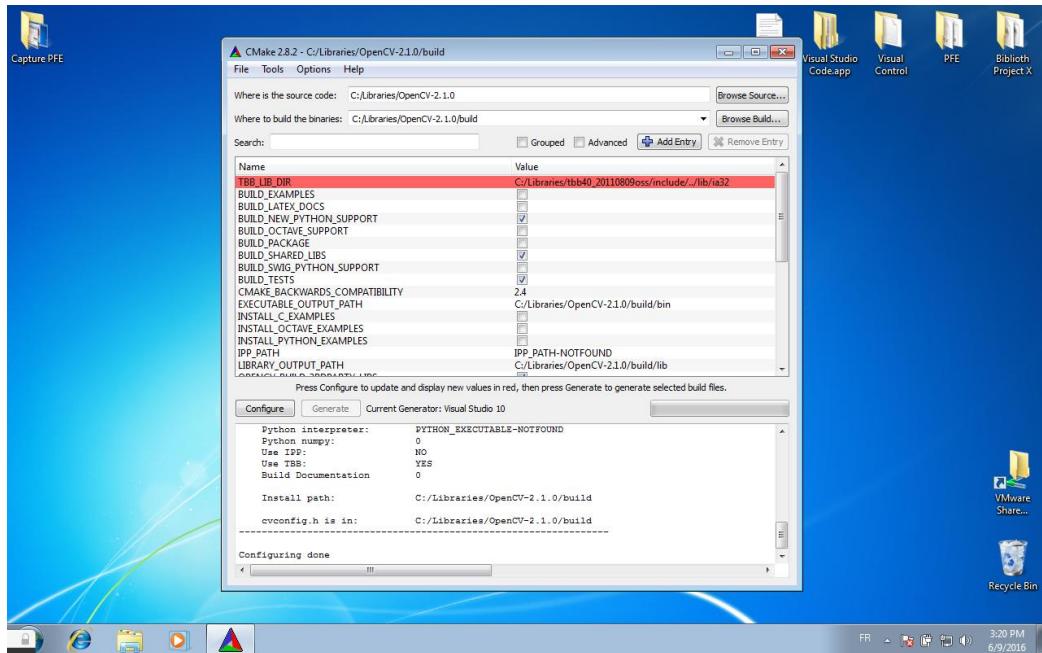


Figure 20 : Installation de TBB

Dans l'emplacement " C:/Libraries/OpenCV-2.1.0/build " on trouve un fichier sln (solution) exécutable par le Visual Studio, il permet de mettre la configuration finale du OpenCV.

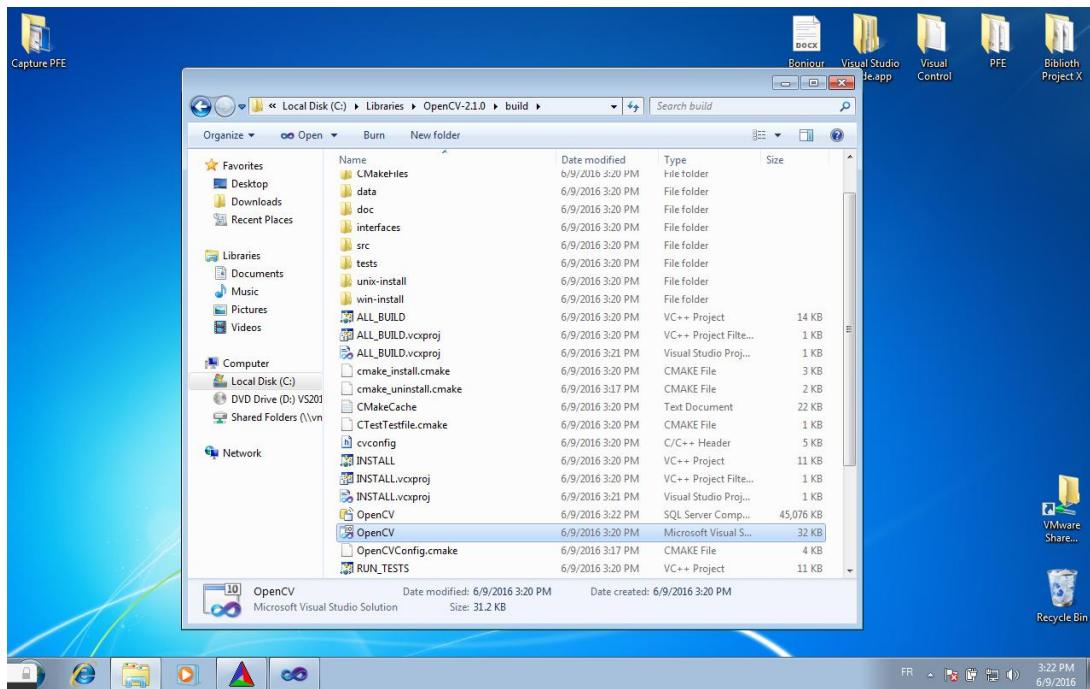


Figure 21 : Installation de OpenCV sur Visual Studio

Après ouverture ca donne comme ça

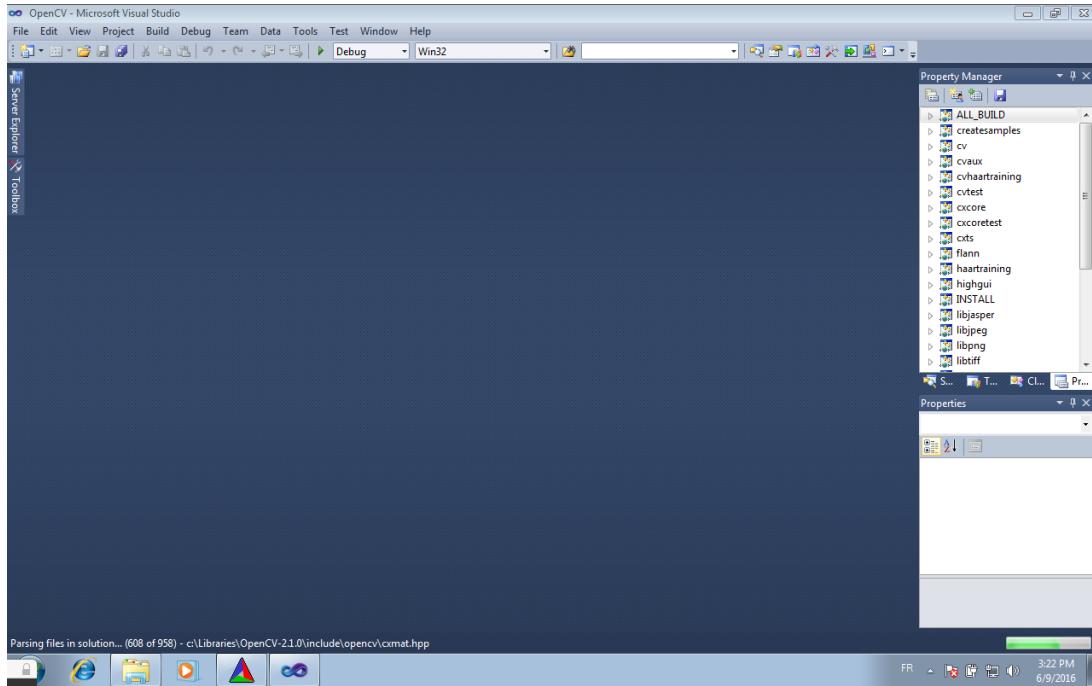


Figure 22 : Installation de OpenCV

et puis, un "Build Solution"

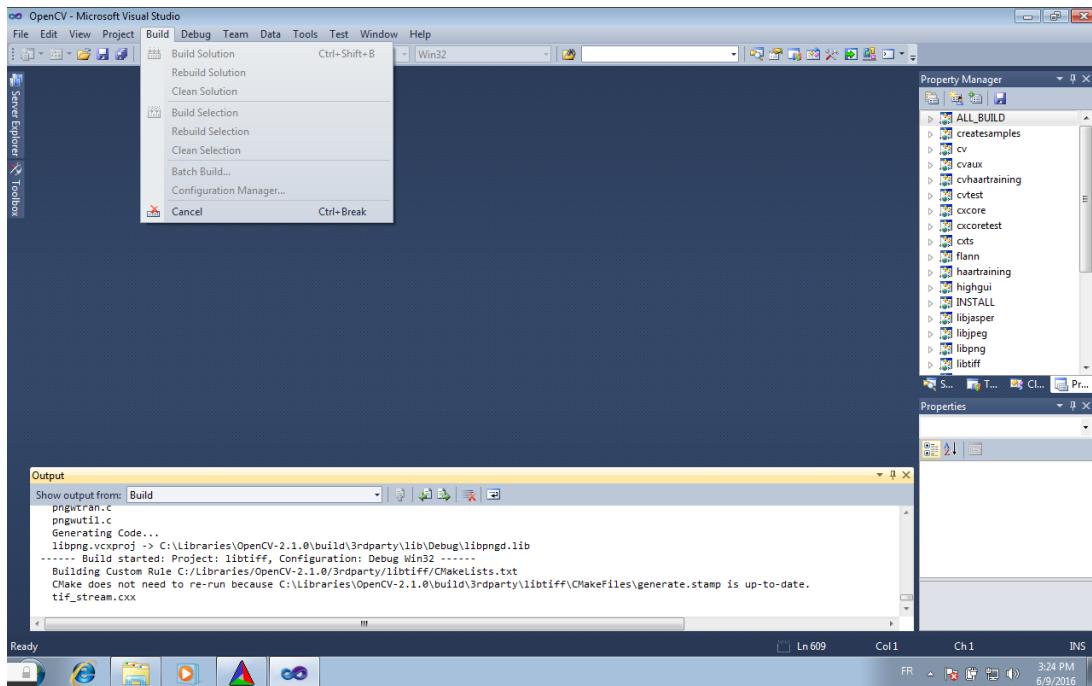


Figure 23 : Installation de OpenCV

Ca prend pas mal de temps, et puis ca marche la bibliothèque est reconnu par Visual Studio mais ce dernier ne détecte pas l'emplacement des fichiers "include", pour en remédier on ajoute une nouvelle propriété au tableau des propriétés nommé " config OpenCV "

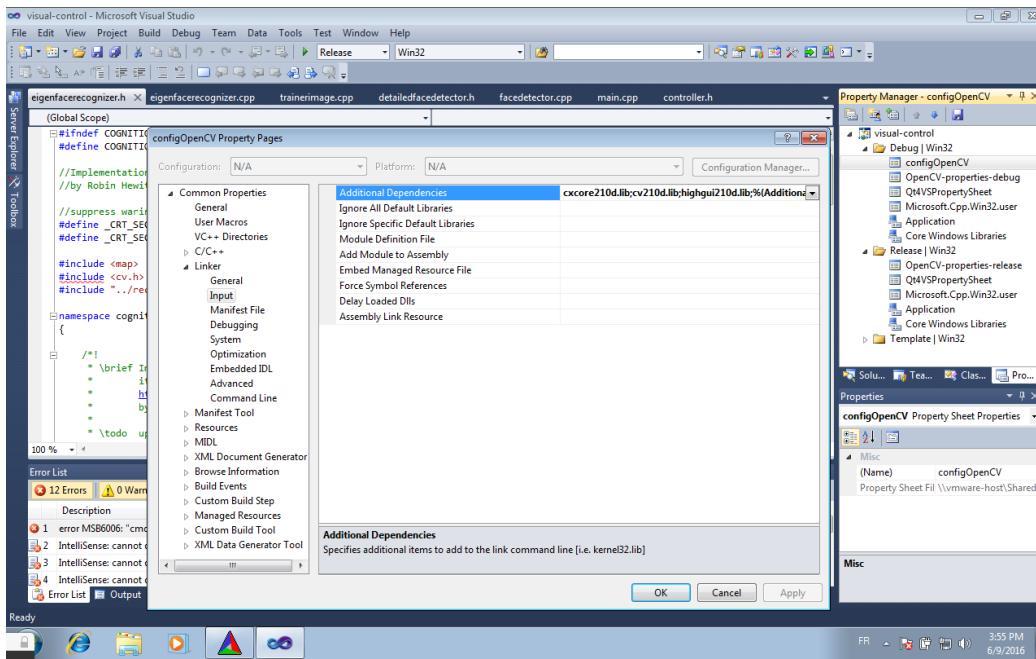


Figure 24 : Installation de OpenCV sur Visual Studio

L'image suivante indique que la bibliothèque est reconnu, l'erreur de l'inclusion du bibliothèque est disparu de la ligne du header #include<cv.h>

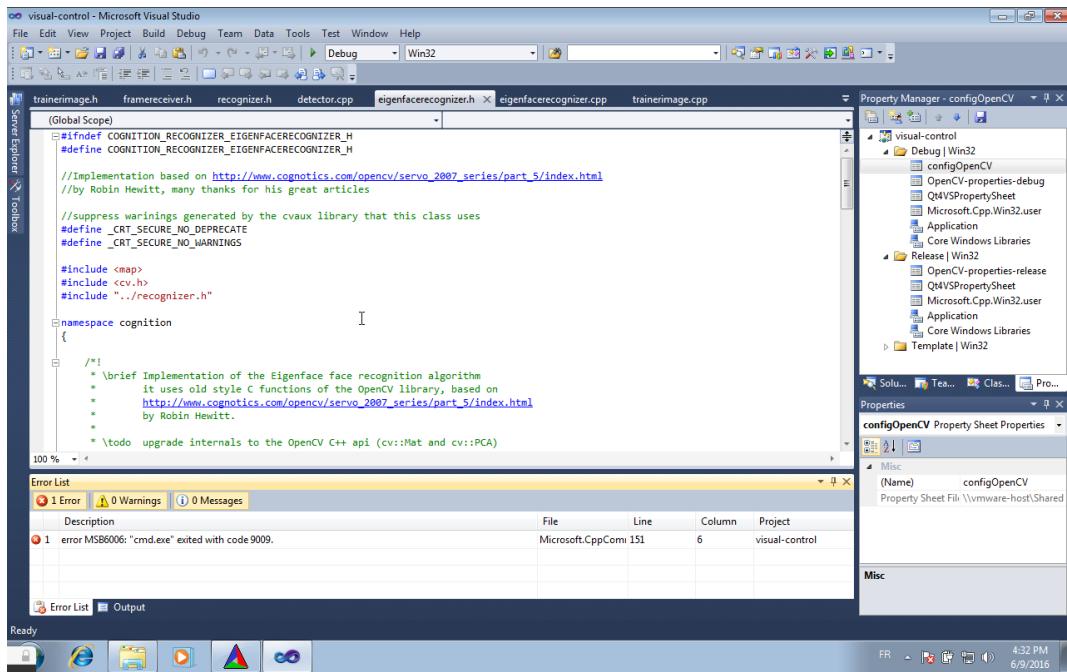


Figure 25 : La disparition des erreurs d'inclusion

2. 2. 2. Teste de l'application

L'interface graphique de l'application est claire, d'abord il faut s'exposer devant le camera pour être détecté dans le carreau rouge pour le visage, les carreaux bleus pour les yeux et le jaune pour la bouche.

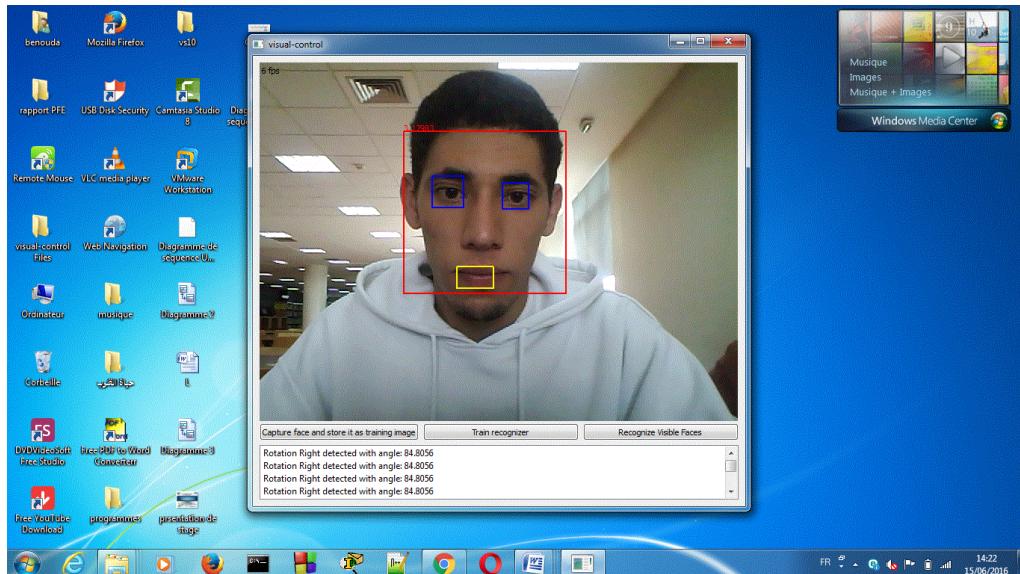


Figure 26 : Test d'application 1

Ensuite, il faut appuyer sur le bouton ‘ Capture image and store it as training image ‘ qui permet d’enregistrer le visage de l’utilisateur.

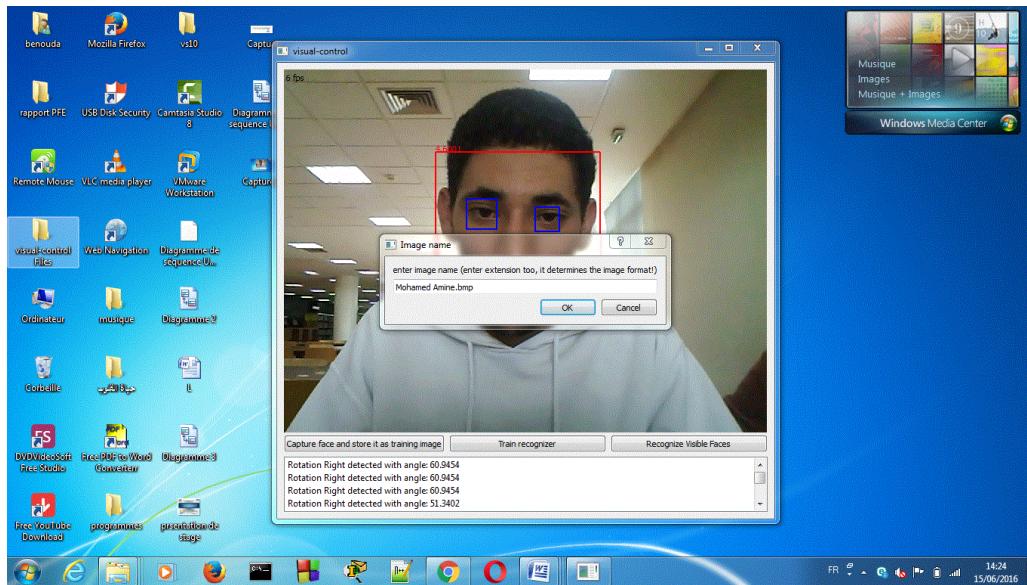


Figure 27 : Test d’application 2

Les captures des visages prises, sont sauvegardées dans la base de reconnaissance du logiciel.

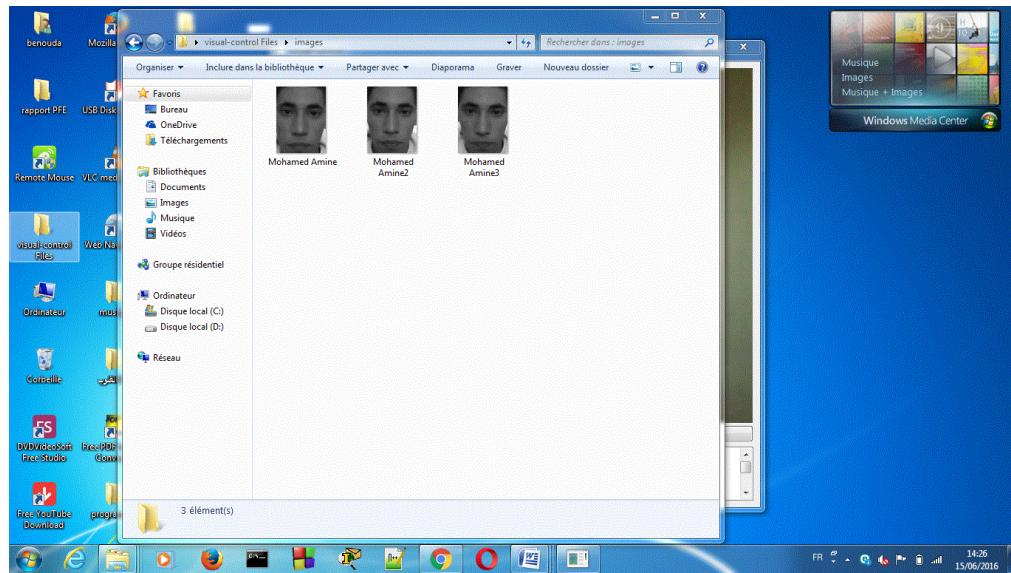


Figure 28 : Test d’application 3

Ensuite, faut entraîner le système en utilisant le bouton ‘ Train Recognizer ‘ . comme on a déjà expliquer dans l’algorithme Eigenface.

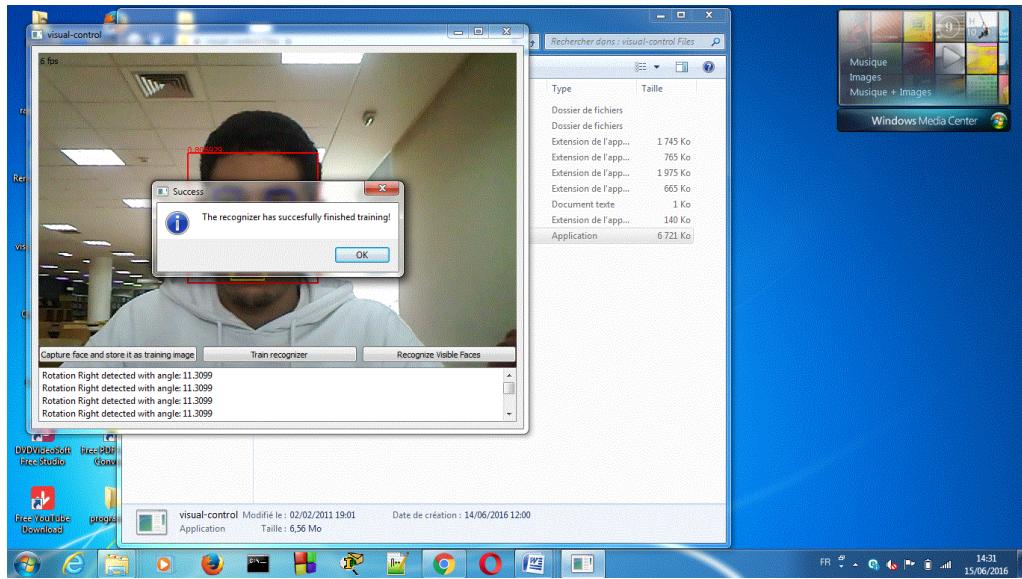


Figure 29 : Test d'application 4

Le bouton ‘Recognize visible Faces’ permet d'afficher l'identité de la personne détecté dans le carreau rouge.

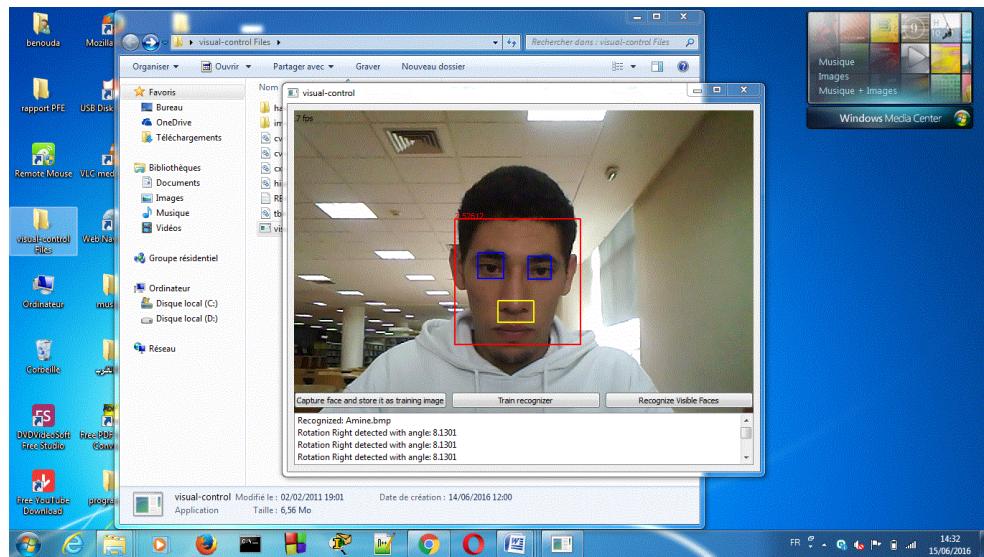


Figure 30 : Test d'application 5

CONCLUSION

Dans ce qui précède, l'ensemble des travaux réalisés nous a permis de mettre en place une application pour répondre aux besoins du problématique posée et de utiliser ce système de manière efficace.

CONCLUSION GENERALE

Le sujet de notre projet était l'implémentation d'un logiciel de reconnaissance d'individus en se basant sur les visages.

Malgré notre motivation portée au choix de ce sujet, nous nous sommes heurtés à quelques difficultés :

Satisfaire les besoins de notre projet du point de vue mémoire du disque, mémoire RAM et processeurs était un problème consistant.

En plus de ce qui suit, la plus part de documentations trouvées sur internet étaient en langue anglaise, ce qui nous a semblé difficile dans notre recherche.

Ce projet nous a permis de découvrir plus profondément plusieurs aspects du développement d'un logiciel complexe. Il nous a fallu d'abord nous renseigner sur le côté algorithmique de la reconnaissance de visage, et plus généralement de la vision par ordinateur, qui est un domaine en pleine expansion. Il nous a fallu résoudre plusieurs problèmes ayant plus ou moins de rapport avec la programmation et utilisations du langage C++.

Ainsi, la réalisation de ce projet nous a permis de comprendre plus en profondeur le développement d'un logiciel, de la conception à la réalisation en passant par la réflexion logique et algorithmique. Si nous avions plus de temps, nous aurions aimé développer un peu plus l'interface graphique afin de compléter son rôle (il ne se serait toutefois agit principalement que de réutiliser des concepts déjà présents dans ce travail). Il aurait été également intéressant d'affiner le moteur de reconnaissance, en trouvant un moyen de prendre des images correctement cadrées et en déterminant les seuils de façon rigoureuse afin d'améliorer le fonctionnement. Si d'autres projets se font sur ce thème, il serait sans doute intéressant de leur fournir l'interface graphique afin de tester une utilisation par des utilisateurs externes.

REFERENCES

Face Recognition Using Principal Components Analysis (PCA)

http://ml.cecs.ucf.edu/meli/wiki/index.php/Face_Recognition_Using_Principal_Components_Analysis_%28PCA%29

Face Recognition using Principal Component Analysis

Kyungnam Kim

Department of Computer Science

University of Maryland, College Park

MD 20742, USA

http://www.cmpe.boun.edu.tr/courses/cmpe360/spring2007/files/PCA/project_spec.pdf

A tutorial on Principal Components Analysis

http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

Analyzing PCA-based Face Recognition Algorithms: Eigenvector Selection and Distance Measures

Wendy S. Yambor Bruce A. Draper J. Ross Beveridge

Computer Science Department

Colorado State University

Fort Collins, CO, U.S.A 80523

July 1, 2000

Similarity Measurement Between Images

Chaur-Chin Chen * and Hsueh-Ting Chu

Department of Computer Science

National Tsing Hua University

Hsinchu 300, Taiwan

E-mail: cchen@cs.nthu.edu.tw

Texture analysis of images using Principal Component Analysis

Manish H. Bharati, John F. MacGregor*

Dept. of Chem. Eng., McMaster University, Hamilton, Ont., Canada, L8S 4L7

PCA : Face Recognition

Rahul Garg 2003CS10183 Varun Gulshan 2003CS10191

October 29, 2005