17/11/2016

Commanditaires:

* M. Jonathan FILLON

Ping N°30

* ISN : M. Congyi SHI
* BIG DATA : M. Youri MAHUT
* ISE-VA : Mme Oulimatou THIAM
* ISE-OC : M. Hermann Kevin YANKAM
* ARI : M. Jean-Arnaud NEMI
* ARI : Mme Siqi XIE (Chef de projet)

Système de reconnaissance faciale pour le contrôle de l’assiduité

Cahier de recettes

# Sommaire

[1. Introduction 2](#_Toc467166917)

[2. Cahier des charges 2](#_Toc467166918)

[3. Liste du matériel nécessaire 3](#_Toc467166919)

[a. Matériel Hardware: 3](#_Toc467166920)

[b. Matériel Software : 3](#_Toc467166921)

[4. Liste des tests à effectuer 3](#_Toc467166922)

[a. Acquisition des données et traitement 3](#_Toc467166923)

[b. Reconnaissance faciale 4](#_Toc467166924)

[c. L’application 6](#_Toc467166925)

[5. Remarques et/ou réserves : 9](#_Toc467166926)

# Introduction

Notre groupe est en charge du développement d’un outil de remplissage de fiches de présences dans le cadre du suivi des cours des étudiants de l’ESIGELEC. Cet outil doté d’une caméra doit pouvoir capter les visages des étudiants présents et inscrire les informations recueillies dans une base de données. Chaque photo prise est une photo de groupe. L’outil est en mesure de découper le contour des images afin d’identifier de manière unique l’étudiant. L’identification est rendue possible grâce au prétraitement des photos étudiantes de références. Le suivi des absences est toujours géré par le service en charge à l’heure actuelle. Notre application est capable d’interagir avec les outils déjà existants. Afin de fournir le recours possible à un étudiant en cas de contestation, les photos de groupe sont conservées un certain temps. L’utilisation de notre outil nécessite une déclaration préalable auprès de la CNIL.

# Cahier des charges

L'objectif de ce projet consiste au contrôle de l'assiduité des étudiants par un système de reconnaissance faciale basé sur la vision. Plus précisément, il s'agira de réaliser un système léger équipé d'une webcam wifi qui sera positionné à l'entrée d'une salle de cours et qui sera dans un premier temps déclenché manuellement par l’étudiant. Pour parvenir au développement de ce système, il conviendra de mener en amont une étude de faisabilité à travers la constitution d'un état de l'art. Plus précisément, une application sera reliée au système d’acquisition et devra permettre de mettre en correspondance l'image acquise en live à l'entrée de la salle de cours avec l'image de référence prise initialement lors de l'arrivée de l'élève à l'ESIGELEC. Une fois cette mise en correspondance réalisée pour l'ensemble des élèves entrés dans la salle (i.e. l'étape d'identification réalisée), l'application développée permettra de visualiser, non seulement, la liste des élèves concernés par le cours planifié (liste d'appel), mais aussi, pour chaque élève listé, son image de référence ainsi que son image acquise en live. Durant le procédé d'identification, un indice de certitude sera calculé permettant ainsi d'associer aux images traitées un pourcentage de reconnaissance de l'étudiant. Ce paramètre sera employé pour pré-remplir de façon automatique la liste d'appel. Si le pourcentage de reconnaissance de l'étudiant est en dessous du seuil d'acceptabilité prédéfini, une alerte sera déclenchée et avertira un opérateur qui procédera alors directement au contrôle. De plus, le système sera muni d'un dispositif d'horodatage des images. Ce dispositif sera employé pour calculer et indiquer le retard de l'étudiant de façon automatique ; c'est-à-dire en estimant l'écart de temps entre le début programmé de la séance et l'arrivée de l'étudiant dans la salle de cours. Au final, l'application développée communiquera avec une application tierce ; à savoir "Aurion", i.e., l’ERP de notre établissement. Les informations d’assiduité détectées (absences et retards) seront inscrites dans un fichier respectant le format standard de l'ERP, puis déposé sur le serveur associé.

# Liste du matériel nécessaire

## Matériel Hardware:

* RASPBERRY-PI  RASPBERRYPI3-MODB-1GB  Raspberry Pi 3 modèle B
* RASPBERRY-PI  RPI NOIR CAMERA BOARD.  Raspberry Pi Carte caméra NoIR, Version 2

## Matériel Software :

* OpenCv
* Visual Studio
* Matlab
* Eclipse

# Liste des tests à effectuer

## Acquisition des données et traitement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Etape** | **Tâche** | **Résultat attendu** | **Résultat obtenu** | **A/R** |
| 1 | Appui sur le bouton de prise de photo | Le script est lancé. La LED de la caméra s’allume et la photo est prise |  |  |
| 2 | Une personne est devant la caméra | Son visage est détecté et la photo est enregistrée dans un dossier pour traitement |  |  |
| 3 | Plusieurs personnes sont devant la caméra | Leurs visages sont détectés et la photo est enregistrée dans un dossier pour traitement |  |  |
| 4 | Prise de photo en étant sous certains angles | Un visage est détecté et la photo est enregistrée dans un dossier pour traitement |  |  |
| 5 | Prise de photo en étant de profil | Un visage est détecté et la photo est enregistrée dans un dossier pour traitement |  |  |

## Reconnaissance faciale

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Etape** | **Tâche** : Amélioration de l’image | **Résultat attendu** | **Résultat obtenu** | **A/R** |
| 1 | Conversion de l’image en niveau de gris | Image noir et blanc |  |  |
| 2 | Améliorer le contraste de l’image en noir et blanc avec la fonction ‘’imadjust ‘’ | Image plus claire |  |  |
| 3 | Filtrer l’image pour atténuer au maximum le bruit dans l’image avec la fonction ‘’medfilt2’’ | Image moins floue |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Etape** | **Tâche** : Création d’une base d’images de personnes à reconnaitre, montage & vérifications | **Résultat attendu** | **Résultat obtenu** | **A/R** |
| 1 | Charger la base de données dans le workspace de MATLAB avec la fonction ‘’*imageSet*’’ | Apparition dans la zone de travail |  |  |
| 2 | Affichage du montage d’un visage au choix de la base de données avec la fonction ‘’*montage*‘’ | Montage des différentes images de visage |  |  |
| 3 | Affichage du montage des visages de la BDD avec un visage de cette BDD cote à côte avec les fonctions ‘’*montage*‘’,’’read’’, ‘’size’’,’’imshow’,  ’’montage’’ | Montage des différentes images de visage BB & visage individuel côte à côte |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Etape** | **Tâche** : Extraction de caractéristiques unique pour chaque visage & apprentissage | **Résultat attendu** | **Résultat obtenu** | **A/R** |
| 1 | Dislocation de la BDD en 2 catégories : visage d’apprentissage – visage de test  Avec la fonction ‘’partition’’ | Catégories générées |  |  |
| 2 | Extraction et affichage pour un seul visage sélectionné avec la fonction ‘’extractHOGFeatures ‘’ | Affichage du visage de la personne choisi , et de la méthode d’extraction |  |  |
| 3 | Extraction des caractéristiques pour les visages d’apprentissage | Matrice de caractéristiques 320\*4680 ; matice d’index 1\*320 |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Etape** | **Tâche** : Modélisation, classification | **Résultat attendu** | **Résultat obtenu** | **A/R** |
| 1 | Création d’un classifier en utilisant la méthode ’’fitcecoc’’. | Apparition du classifier dans la zone de travail MATLAB |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Etape** | **Tâche** : Test d’identification | **Résultat attendu** | **Résultat obtenu** | **A/R** |
| 1 | Extraire les caractéristiques de l’image d’entrée ‘’ExtractHOGFeatures ‘’ | Matrice des caractéristiques |  |  |
| 2 | Comparer ces caractéristiques avec celles du classifier des autres images a travers les fonctions ‘’find’’,’’strcmp’’ | Montage des différentes images de visage |  |  |
| 3 | Afficher les visages appariés | Affichage côte à côte du visage de test et de celui apparié |  |  |
|  |  |  |  |  |

## L’application

|  |  |  |  |
| --- | --- | --- | --- |
| **Tâche** : Manipulation de la base de données | **Résultat attendu** | **Résultat obtenu** | **A/R** |
| Ajout élément | Ajout d'un élément |  |  |
| Ajout élément | Au besoin : Vérification de l'unicité de l'élément ajouté |  |  |
| Ajout élément doublon | Message prévenant l'utilisateur |  |  |
| Récupération d'un élément | Vérification de la validité de l'élément et récupération |  |  |
| Récupération d'un élément | Création de l'élément BDD en tant qu'objet Java |  |  |
| Suppression d'un élément | Vérification de la validité de l'élément et suppression |  |  |
| Suppression d'un élément | Vérification de la validité de l'incidence de la suppression |  |  |
| Suppression d'un élément | Application des actions inhérentes à la suppression de l'élément sur les autres éléments de la bdd |  |  |
| Update d'un élément | Vérification de la validité de l'élément et update |  |  |
| Update d'un élément | Vérification de la validité de l'incidence de la update |  |  |
| Update d'un élément | Application des actions inhérentes à la MAJ de l'élément sur les autres éléments de la bdd |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Étape** | **Taches :** Manipulation des maquettes | **Résultat attendu** | **Résultat obtenu** | **A/R** |
| 1 | Login sans email, mot de passe | Message : manque email et nécessaire |  |  |
| 2 | Login sans mot de passe | Message : manque mot de passe |  |  |
| 3 | Login avec mot de passe/email faux | Message : erreur de mot de passe ou email |  |  |
| 4 | Login correct (professeur) | Message login succès ! Tourner à page choix de cours pour professeur |  |  |
| 4 | Login correct(admin) | Message login succès ! Tourner à page admin |  |  |
| 5(professeur) | Choix du cours et cliquer bouton « entre » | Tourner à page prendre photo |  |  |
| 6(professeur) | Cliquer bouton cliquer photo | Prendre et afficher les photos |  |  |
| **Etape** | **Taches :** Manipulation des maquettes | **Résultat attendu** | **Résultat obtenu** | **A/R** |
| 7(professeur) | Comparaison | Tourner à la page liste d’absence |  |  |
| 8(professeur) | Modifier les textes dans la liste | Modifier et sauvegarder les texte |  |  |
| 9(professeur) | Cliquer bouton enseigner | Enseigner la liste d’absence |  |  |
| 5(admin) | Cliquer bouton d’enseignants/ salles/cours/étudiants/écoles | Page d’enseignants/ salles/cours/étudiants/écoles |  |  |
| 6(admin) | Boutons de Modifier/créer/supprimer professeur/ salles/ écoles/ étudiants/ cours | Modifier/créer/supprimer professeur/ salles/ écoles/ étudiants/ cours |  |  |
| 5(admin) | Cliquer bouton modifier liste d’absence | Tourner a page modifier les listes d’absences |  |  |
| 6(admin) | Boutons de Modifier/supprimer liste d’absence | Modifier/supprimer liste d’absence |  |  |

# Remarques et/ou réserves :