ENSICAEN - 2A Spécialité Informatique
Projet 2A

Reconnaissance faciale : anti-fake

Vimont Ludovic & Kotulski Guillaume

Promo 2016
12 mars 2015

Remerciements

Nous tenons à adresser nos remerciments aux personnes qui ont permis que ce projet se déroule dans les meilleures conditions possibles.

Cherrier Estelle Schwartzmann Jean-Jacques Faye Ndiaga

Sommaire

1 Introduction

1.1 Contexte

Dans le cadre des cours de l'ENSICAEN, nous devions réaliser un projet de deuxième année. Nous avons choisi de prendre le projet reconnaissance faciale anti-fake proposé par M. Schwartzmann Jean-Jacques.

1.2 Objectifs

L'objectif du projet est la réalisation d'un moyen d'authentification par reconnaissance facile grâce à l'utilisation d'une capture vidéo. En effet, actuellement ce genre d'applications sont facilement attaquables. Prenons l'exemple d'un smartphone qui se déverrouille grâce à ce moyen. Il suffit pour un attaquant de disposer de la photo de la victime pour contourner la protection. Ce projet consiste donc à utiliser une capture vidéo et être capable de différencier un humain d'une photo grâce à cette enregistrement. Afin de réaliser cette différenciation, nous devions baser notre travail sur une recherche du MIT. Enfin, nous devons intégrer notre résultat dans une application android de reconnaissance faciale forunie par le GREYC.

1.3 Contraintes

La seule contrainte dont nous disposions était le temps de reconnaissance de la personne, en effet pour l'application android au-dessus de 4 secondes cela était beaucoup trop long, d'un point de vue utilisateur.

2 Développement

2.1 Outils mis en place

— Site web en ligne (CMS Wordpress) : http://www.ecole.ensicaen.fr/~lvimont/
— Gestionnaire de versions (Git)
  • Application Android : https://github.com/F4r3n/Vamp
  • Logiciel de traitement d'images : https://github.com/F4r3n/ImageProject.git
— Gestionnaire de projet (Trello)

2.2 Technologies utilisées

java, c++, qt, android, opencv

2.3 Algorithme mis en place

L'oeil humain est limité, en effet il est incapable de voir les subtils changements temporels. Alors que justement en effectuant des traitements sur une vidéo. On est capable de révéler ces changements comme par exemple, la circulation du sang. On peut même grâce à ça réussir à connaitre le pouls d'une personne.

Notre première idée fut simplement de réaliser une moyenne pour chaque couleur pour un pixel, puis une moyenne de tout les moyennes calculer par pixels. On pensait alors par la suite grâce à cette collection de moyenne arriver a détecter une variation. En effectuant par exemple, une dérivée puis une amplification de la courbe obtenue.

Dans un second temps, en restant basé sur le même principe, nous avons tenté de découper notre image par zone de petits carrés de pixels. Par exemple une zone serait d'une taille de 5*5. On réalise ainsi une moyenne de chaque zone, puis par la suite une moyenne de ces moyennes. Enfin on réapplique les mêmes fonctions qu'au-dessus.

2.4 Solutions mis en place

Durant le projet nous avons développés plusieurs outils pour répondre au besoin. Une application Android qui permettrait d'utiliser le moyen d'authentification désiré et ainsi déverouiller le smarthpone d'une personne. Un logiciel C++ utilisant la librairie QT a également été mis en place, afin de pouvoir tester plus facilement les algorithmes que nous voulions développer. Dans la suite du projet, il a également permis d'utiliser la webcam.

2.4.1 Développement Android

— Détection visage en temps réel
— Rectangle autour visage
— Capture de 15 images par seconde
— Détection couleur pixel (RGB)

— Caméra fonctionnelle
— Compteur sur 5 sec
— Algo mis en place

Lockscreen

2.4.2 Développement C++

— Utilisation de avconv : résultat vidéo divisé en plusieurs images au rythme 15 FPS
— Dessin d'un rect autour des images
— Réalisation de la moyenne
— Fonctions : dérivé, amplification, fft ...
— Filtres passe bas
— Changement espaces de couleurs
— Capture avec une webcam (opencv)
— Ajout du trigger
— Variations

2.5 Problèmes rencontrés

Nombres de FPS capturés avec la caméra frontal de android.

3 Résultats

3.1 Cadre idéal

Nous avons effectuer différents tests durant le projet. En utilisant les vidéos fournis sur le site du MIT. Notre logiciel arrive à obtenir des résultats très correct. En effet, par exemple comme on peut le voir sur cette capture d'écran. Après notre magnification réalisé, on retrouve une fréquence cardiaque d'environ 53 battements par minute (BPM), l'article du MIT lui trouve une fréquence de 54 BPM.

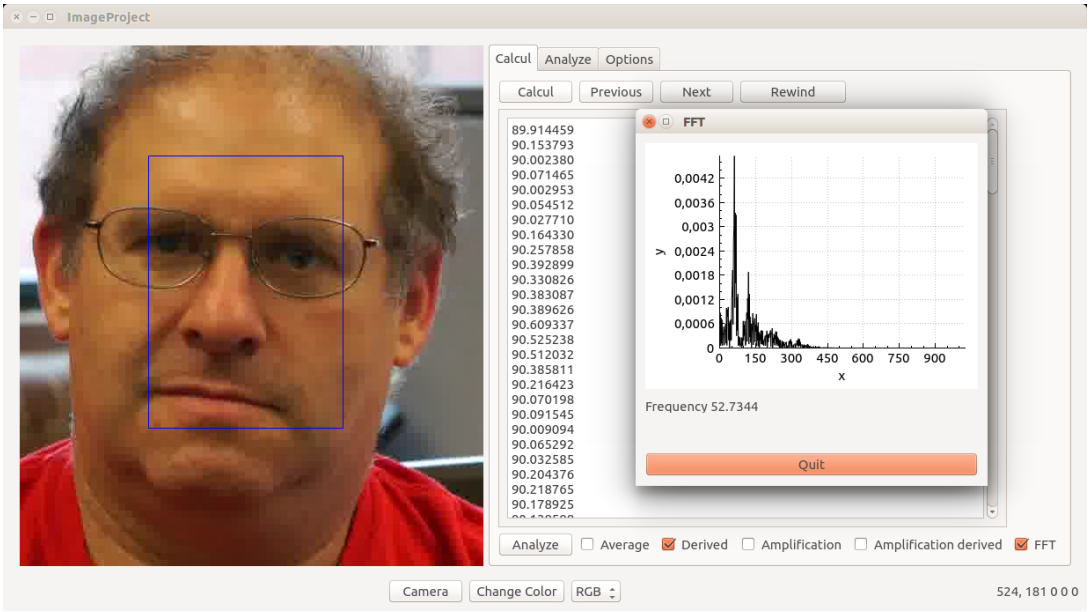

FIGURE 3.1 – Vidéo source du MIT analysé avec notre logiciel.

3.2 Webcam

Avec une webcam, nos résultats sont moins précis, mais sont assez correct pour être utiliser.
capture webcam here

3.3 Android

Notre plus gros problème a bien entendue était lors de nos tests sur Android, la stabilité de la vidéo et le nombre de frames capturés. Actuellement notre application est capable de capturer des frames et appliqué notre algorithme mais nos résultats restent éronnés.

4 Conclusion

4.1 Apports techniques

Github
C++/Qt
Connaissances sur Android

4.2 Apports scientifiques

Espaces de couleurs
Traitement du signal

4.3 Apports humains

Organisation

4.4 Bilan

A Annexes