

ENSICAEN - 2A SPÉCIALITÉ INFORMATIQUE
PROJET 2A



Reconnaissance faciale : anti-fake
VIMONT Ludovic & KOTULSKI Guillaume

PROMO 2016
12 mars 2015

1 Remerciements

FAYE Ndiaga SCHWARTZMANN Jean-Jacques

2 Introduction

2.1 Contexte

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

2.2 Objectifs

L'objectif du projet est la réalisation d'un moyen d'authentification par reconnaissance faciale 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 devons baser notre travail sur une [recherche](#) du MIT. Enfin, nous devons intégrer notre résultat dans une application android de reconnaissance faciale fournie par le [GREYC](#).

2.3 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.

3 Développement

3.1 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)

3.2 Technologies utilisées

java, c++, qt, android, opencv

3.3 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 connaître 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 à 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.

3.4 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 déverrouiller le smartphone 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.

3.4.1 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

3.4.2 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

3.5 Problèmes rencontrés

4 Résultats

Cadre idéal

5 Conclusion