



NGUYEN Thi-Christine - RICHARD Pierre
24/10/2024

Sommaire

1	Introduction	2
2	Etape 1 : La base de données	2
3	Etape 2 : Reconnaissance des caractéristiques	2
3.1	Haar Cascade	2
4	Etape 3 : Analyse du visage	2
5	Etape 4 : Trouver le vecteur le plus proche	2
6	Conlusion	2

1 Introduction

Suite aux recherches de la semaine précédente, nous avons du faire des choix sur les méthodes à implémenter pour notre application de reconnaissance faciale. Lors de ce compte rendu, nous vous expliquerons les différentes étapes que nous envisageons pour l'implémentation.

2 Etape 1 : La base de données

Nous allons tout d'abord récupérer un maximum d'image de personnes sous différents angles. Ces images serviront à entraîner notre modèle. Nous avons décidé de prendre pour le moment des images de célébrité car elles sont plus accessibles et il en existe de nombreuses.

3 Etape 2 : Reconnaissance des caractéristiques

Nous allons ensuite utiliser des méthodes pour détecter les visages dans les images. Après des multitudes de recherche dans l'optique de faire de la détection en temps réel, nous avons choisi d'utiliser l'algorithme d'Haar Cascade.

En effet, cet algorithme est rapide et va nous permettre de détecter les visages en temps réel.

3.1 Haar Cascade

L'algorithme **Haar Cascade** est une méthode de détection d'objets rapide et efficace, développée par Paul Viola et Michael Jones. Il utilise des caractéristiques visuelles appelées *caractéristiques Haar*, qui représentent des motifs simples de zones claires et foncées. Ces caractéristiques permettent de détecter des motifs typiques du visage humain, tels que les yeux, le nez et la bouche.

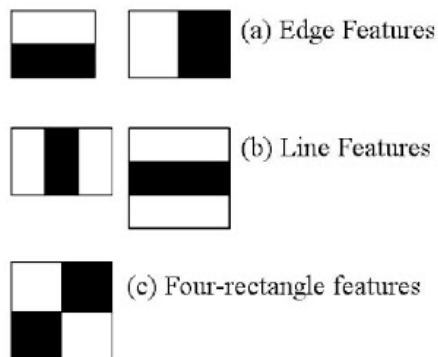


Figure 1: Haar features

Pour optimiser la vitesse de détection, l'algorithme utilise un **classificateur en cascade** qui permet de concentrer les calculs sur les zones les plus probables de contenir un visage. Grâce à cette approche, Haar Cascade est particulièrement bien adapté pour la reconnaissance faciale en temps réel, car il permet d'effectuer la détection de manière rapide et précise, ce qui est essentiel dans notre programme.

4 Etape 3 : Analyse du visage

Pour l'analyse du visage, nous allons utiliser le modèle VGG qui est une architecture de CNN. Le VGG utilise des petites fenêtres de convolution de manière répétée, ce qui permet d'extraire des caractéristiques complexes.

5 Etape 4 : Trouver le vecteur le plus proche

Après avoir mis en place notre modèle VGG pour l'extraction des caractéristiques faciales, nous allons utiliser le modèle pour transformer chaque image de visage en un vecteur de caractéristiques. Une fois ces vecteurs extraits, nous pouvons les utiliser pour la comparaison et l'identification des visages.

Pour identifier ou vérifier un visage, nous allons utiliser un classifieur SVM.

6 Conclusion

En suivant toutes ces étapes, nous allons pouvoir obtenir un système de reconnaissance faciale capable d'identifier les visages. La combinaison de l'algorithme Haar Cascade pour la détection rapide et du modèle VGG pour l'extraction des caractéristiques permet de créer une solution robuste. Grâce au classifieur SVM, nous pouvons associer chaque visage à une identité unique parmi celles présentes dans notre base de données.