Implémantation d'un système de détection et de reconnaissance faciale

Alexandre Bonhomme
Université Paul Sabatier, Toulouse
Florent Maufras
EFREI, Paris
Timothée Planté
EISTI, Pau

1 Introduction

La reconnaissance faciale est actuellement un domaine en plein essort. Elle rentre petit à petit dans nos vies au travers de nos téléphones mobiles ou de nos ordinateurs portables par exemple. Malgré l'amélioration du taux de détection elle reste l'objet de nombreuses études actuellement.

L'objectif de notre projet sera de mettre en oeuvre un système complet permettant la détection et la recnnaissance de visage issue de plusieurs bases de données. Pour ce faire nous implémenterons différents algorithmes de reconnaissances et nous utiliserons la bibliothèque OpenCV [3] et son implémentation de l'algorithme de Viola & Jones pour la détection de visage.

2 Algorithmes mis en oeuvre

Au cours de notre projet nous avons confronté différents algorithmes afin de tester leurs aptitudes sur plusieurs ensembles de donnée différents.

2.1 K Plus Proches Voisins et Fenêtres de Parzen

L'algorithme des «K plus proches voisins» (KPPV) utilise le voisinage du point de test afin de déterminer ça classe. C'est la classe majoritaire parmis ces K points qui sera attribuée au point de test (1).

Pour cet algorithme il n'y à pas d'apprentissage des paramètres sur l'ensemble d'entrainement, seulement une optimisation des l'hyperparamètres K sur l'ensemble de validation.

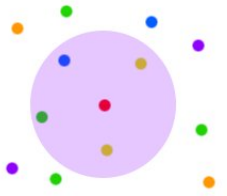

Figure 1 – K Plus Proches Voisins

Dans la Figure 1 le point de test (en rouge) va recevoir comme prédiction la classe jaune car c'est elle la plus présente dans les $K = 4$ voisins les plus proches. Le disque ici n'est que représentatif pour mettre en évidence les voisins. Les points voisins pouvant être à une distance plus ou moins importantes de notre point test.

2.1.1 Extension avec les fenêtres de Parzen

L'application que l'on à fait ici de Parzen nous permet de pondéré les points du voisinage. Plus un point sera distant du point à l'étude plus sont poids sera faible. Cela est calculé à partir d'une gaussienne centrée sur le point à l'étude et d'une largeur fixé par σ, dans notre cas σ sera fixé par l'utilisateur.

2.2 Réseau de Neurones

Le réseau de neurone utilisé dans ce projet est de type Perceptron Multicouche. Ce modèle est entrainé par descente de gradient de bacth sur un ensemble d'entrainement afin d'optimiser les paramètres du réseau en réduisant le risque empirique régularisé. On tâchera ensuite d'optimiser les hyperparamètres de contrôle de capacité sur un ensemble de validation séparé.

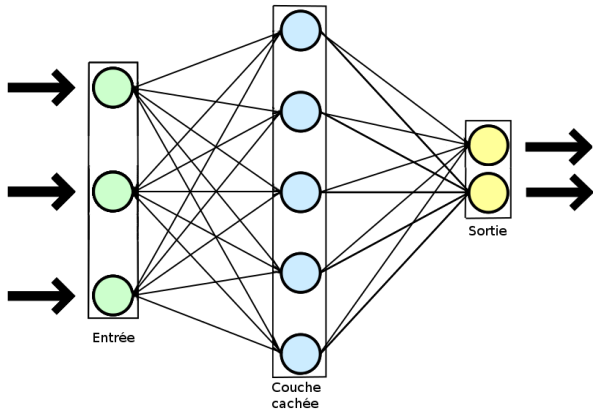

Figure 2 – Réseau de neurones

Dans notre cas nous utiliserons une architechture monocouche (cf. Figure 2) et une pénalité de type «weight decay» (1) pour la régularisation car elle défavorisera les poids les plus lourds lors de l'apprentissage des paramètres.

$$\Omega(\theta) = \|\mathbf{W}^{(1)}\|^2 + \|\mathbf{W}^{(2)}\|^2 \quad (1)$$

Cette pénalité sera pondérée par un hyperparamètre λ.

2.3 Algorithme de Viola et Jones

Afin d'effectué efficassement la détection de visage nous avons utilisé une implémentation de l'algorithme de Viola et Jones [4] proposé par la célèbre bibliothèque de traitement d'image *OpenCV* [3].

Cet algorithme est basé sur le principe de «boosting» qui conciste à assembler plusieurs classifieurs faibles afin d'obtenir un classifieur à forte capacité. Plutôt que de travailler directement sur les pixel de l'image, Viola P. et Jones M., introduise des caractéristique appelées «pseudo-Haar» fortement inspirées des ondelettes de Haar. De plus il utilise une variante d'un algorithme de boosting très célèbre qui est l'«AdaBoost».

(a) Image initiale **(b)** Visage extrait

Figure 3 – Détection d'un visage par l'algorithme de Viola et Jones [4]

Cette technique est particulièrement efficace et offre d'excelents résultats pour la détection de visages (cf. Figure 3) et par extention la détection de motifs (ou d'objets) au sein d'une image.

3 Ensembles de données utilisés

Afin d'entrainer nos algorithme et de mesurer leurs perfomance nous avons essentiellement utilisé deux jeux de données.

3.1 Our Database of Faces (ORL [2])

Le premier ensemble de donnée utilisé est issu des laboratoire de l'entreprise *AT&T*. C'est un ensemble de taille moyenne qui regroupe au total 40 sujets différent avec 10 images par sujets. Les photos on été prise sous différentes postures et durant plusieurs années et on une taille de 92×112 pixels.

Les images sont en niveaux de gris (NdG) et au format PGM.

3.2 Labeled Faces in the Wild Home (LFW [5])

Pour le second jeu de données, il est mis a diposition par l'université du Massachusetts et il s'agit de photographies tirées d'Internet et sur lesquels l'algorithme de Viola et Jones [4] a été appliqué afin d'obtenir une image de taille 250×250 pixels centré sur le visage.

Cet ensemble est relativement grand puis-ce qu'il contient 13233 images de 5749 sujets différents. Les images sont en couleur et au format JPG.

3.3 Prétraitements des données

3.3.1 Cas particulier : l'ensemble LFW

Contrairement à l'ensemble ORL, LFW nécéssite des prétraitement supplémentaires. En effet, nous avons utilisé l'ensemble LFW pour simuler une entrée réel dans noter système. Pour ce faire nous avons appliqué l'algorithme de Viola et Jones afin de détecter les visages au sein des images présentes dans l'ensemble.
Nous avons ainsi pu extraire (après divers traitements de redimentionnement) une image de taille 92×115 pixels en plusieurs NdG.

3.3.2 Cas général

Appartir des images en différents NdG nous construisons une représentation vectoriel de celle-ci (cf. Figure 4).

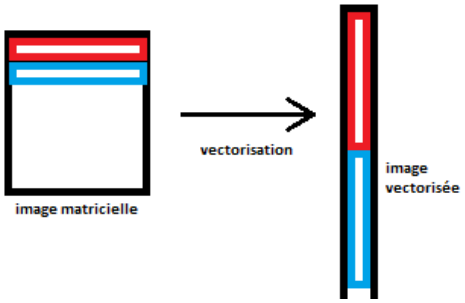

Figure 4 – Vectorisation d'une image

Cependant cette représentation est très lourde car elle contient de nombreuses caractéristiques. Afin de reduire la dimentionailé de nos données d'entraienement nous avons appliqué l'agorithme du *PCA* («Principal Components Analysis»), plus connu sous le nom de «Eigenfaces» [1] lorsqu'il est appliqué à la reconnaissance facial.

3.3.3 Réduction de la dimentionnalité : PCA

Nous avons implémenté le PCA dans ca forme classique.
On normalise l'ensemble de donnée en sustraillant le «visage» moyen :

$$X_N = X - \bar{x} \tag{2}$$

A la suite de quoi nous utilisons une «astuce mathématique» qui consiste à calculer la Décomposition en Valeurs Singulières (ou SVD pour «Singular Value Decomposition») afin de déterminer les vecteurs propres de la matrice de covariance. On obtient alors :

$$X_N = USU^T \tag{3}$$

Ou les colonnes de U contienent les vecteurs propres et S^2 les valeurs propres associées. On ne conserve alors qu'une nombre M de vecteurs propres (ie. M colonnes de la matrice U). Après réduction du nombre de colonne on obtient une matrice U' (aussi appelée, dans notre cas, «Eigenspace») dans laquelle on projetent l'ensemble de donnée X afin de déterminer la matrice de poids W associée :

$$W = \langle (U')^T, X \rangle \tag{4}$$

4 Résultats

5

Un des premiers points spécifiques à notre projet et le fait que ce dernier a été réalisé par une équipe de trois personnes ayant suivi des formations différentes. Nous n'avions pas les même disponibilités non plus pour avancer le projet du fait que nous étions inscris à des cours différents aussi. Cette raison a mené à la mise en place d'une gestionnaire de version avec une centralisation via un site web qui offre ce service : https ://bitbucket.org/ Ceci à donc rajouter une difficulté au projet qui était la gestion de l'équipe et la répartition des tâches au sein de cette dernière. L'usage d'un logiciel tel que Git pour gérer les versions en local et via aussi la passerelle distante.

L'usage du python était aussi une des difficultés auquel nous avons voulu nous confronté étant donné que nous avons tous trois découvert ce langage au cours des derniers mois nous voulions essayé de le pratiquer dans en longueur au cours de ce projet.

Le sujet lui-même représente une difficulté quant à la recherche d'un bon résultat étant donné qu'encore à ce jour des entreprises et chercheurs travaillent encore au développement et à l'amélioration de système de reconnaissance faciale car aucun système n'est encore réellement fiable dans ce domaine.

6 Conclusion

Références

[1] Turk M. and Pentland A., *Eigenfaces for recognition.* J. Cognitive Neuroscience, 1991.

[2] Samaria F. and Harter A., *Parameterisation of a stochastic model for human face identification.* 2nd IEEE Workshop on Applications of Computer Vision, 1994.

[3] Bradski G., *The OpenCV Library.* Dr. Dobb's Journal of Software Tools, 2000.

[4] Viola P. and Jones M., *Rapid object detection using a boosted cascade of simple features.* Computer Vision and Pattern Recognition, 2001.

[5] Huang G. B., Ramesh M., Berg T. and Learned-Miller E., *Labeled Faces in the Wild : A Database for Studying Face Recognition in Unconstrained Environments.* University of Massachusetts, Amherst, 2007.