

Compte Rendu Semaine 3

# Accès sécurisé par reconnaissance faciale

Josua Philippot - Félix Yriarte  
Master 2 IMAGINE

Octobre 2021 - Décembre 2021



UNIVERSITÉ  
DE MONTPELLIER

# Table des matières

<b>1</b>	<b>Travail effectué</b>	<b>2</b>
1.1	Réflexion quant à la comparaison de résultats . . . . .	2
1.2	Amélioration à apporter aux LBPH . . . . .	2
1.3	Eigenfaces . . . . .	3
<b>2</b>	<b>Perspectives</b>	<b>3</b>
<b>3</b>	<b>Nos sources</b>	<b>3</b>

# 1 Travail effectué

## 1.1 Réflexion quant à la comparaison de résultats

Quelle que soit la méthode "traditionnelle" (n'utilisant pas de réseaux de neurones convolutifs) utilisée pour la reconnaissance faciale, il existe des paramètres à fixer, afin d'obtenir les meilleurs résultats possibles. Par exemple, pour la méthode par LBPH, nous calculons la divergence entre deux histogrammes concaténés afin de déduire une mesure de la ressemblance entre deux visages. Il reste nécessaire de fixer un seuil à partir duquel nous déclarerons les visages comme provenant de la même personne.

Nous avons pu expérimenter notre algorithme développé l'an dernier sur la base d'images de visages *AT&T* [1], qui est constituée de 400 images en niveaux de gris de visages, venant de 40 personnes différentes.

Nous avons déjà pu remarquer une différence sensible entre la divergence de deux images de visages (différentes) d'une même personne, et la divergence de deux images de visages provenant de personnes différentes. Même si ces résultats sont encourageants, il sera nécessaire d'améliorer notre méthode de comparaison d'histogrammes, puis dans un second temps, de tracer des courbes ROC, afin de définir au mieux une valeur de seuil permettant de discriminer les matchs censés être positifs et négatifs.

Cette dernière méthode d'affinage des paramètres par courbes ROC sera applicable à la méthode par eigenfaces, développée pendant ce projet, comme on pourra le voir dans la suite de ce rapport.

## 1.2 Amélioration à apporter aux LBPH

L'année dernière nous avons implémenté un système de reconnaissance faciale utilisant les LBPH. Celui-ci nous permettait de comparer deux images et d'obtenir un score de reconnaissance qui nous permettait de quantifier à quel point nos deux images étaient susceptibles d'être des photos du même visage (plus le score était bas, moins il y avait de différence entre les deux images). En reprenant cette base, il y avait quelques points que l'on pouvait améliorer.

Tout d'abord, notre système prenait en compte l'intégralité de l'image, ce qui le rendait très sensible à deux choses : Le fond de l'image (et plus généralement tout ce qui ne faisait pas parti du visage), mais aussi la résolution de l'image. Pour pallier ce problème, il nous fallait dans un premier temps mettre les images à comparer à la même résolution, et éliminer le bruit autour du visage, soit faire de la détection de visage.

Nous avons lu dans la littérature[2, 3], que dans les systèmes qui utilisent les LBPH, il y a d'abord une phase de détection de visage qui est mise en place pour récupérer uniquement la zone du visage et ainsi limiter le bruit. Pour faire cette acquisition, la méthode du classificateur en cascade de Haar est souvent utilisée car elle est considérée comme étant fiable et rapide à mettre en place. Ainsi nous avons décidé d'utiliser l'implémentation du classificateur en cascade proposée par *OpenCV* [4] pour faire notre phase de détection de visage.

D'autre part, il nous est possible d'optimiser la reconnaissance avec LBPH en comparant les images d'une certaine manière : quand on utilise LBPH on obtient une matrice d'histogrammes

qui représente les informations de l'image divisée par blocs, en comparant les parties les plus pertinentes de l'images (par exemple l'emplacement ou devrait se trouver les yeux, le nez, et la bouche) il nous serait potentiellement possible de comparer uniquement les caractéristiques "importantes" d'un visage. Aussi, nous avons trouvé dans la littérature [3] une variante de la méthode classique de LBP nommée MLBP, dans laquelle la valeur considérée pour appliquer la LBP est la médiane du voisinage considéré, et non pas la valeur du pixel courant.

### 1.3 Eigenfaces

Nous avons commencé à développer une seconde méthode de reconnaissance faciale "traditionnelle" (n'utilisant pas de réseaux de neurones convolutifs) par Eigenfaces [5, 6, 7]. Dans cette méthode, des caractéristiques représentatives d'un ensemble d'images de visages sont déduites par récupération des vecteurs propres de la matrice de covariance d'un ensemble d'images.

Nous avons pu débiter l'implémentation de cette méthode en utilisant la librairie C++ d'opérations matricielles Eigen [8]. Il nous est encore nécessaire de développer la récupération des vecteurs propres de la matrice d'images, afin de pouvoir discriminer des images de visages par leurs caractéristiques dans le domaine des visages (*face space*).

## 2 Perspectives

Nous pouvons donc avancer sur différents points :

- Mise en place de courbe ROC (expérimentations approfondies sur la base [1] intégrale)
- Mise en place d'une détection de visage à l'aide d'un classificateur en cascade de Haar.
- Mise en place et comparaison de notre méthode par LBP et de celle par MLBP.
- Mise en place de la méthode de reconnaissance par Eigenfaces.

## 3 Nos sources

L'intégralité du travail que nous allons produire se trouve sur ce dépôt Github :  
<https://github.com/JPhilippot/FaceRecognition>

## Références

- [1] Kasikrit Damkliang. *AT&T Database of Faces*. kaggle.com <https://www.kaggle.com/kasikrit/att-database-of-faces>, 2021.
- [2] Aftab Ahmed, Jiandong Guo, Fayaz Ali, Farha Deebe, and Awais Ahmed. Lbph based improved face recognition at low resolution. In *2018 international conference on Artificial Intelligence and big data (ICAIBD)*, pages 144–147. IEEE, 2018.
- [3] XueMei Zhao and ChengBing Wei. A real-time face recognition system based on the improved lbph algorithm. In *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*, pages 72–76, 2017.
- [4] OpenCV. *Cascade Classifier*. [https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html), 2021.
- [5] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, pages 586–587. IEEE Computer Society, 1991.
- [6] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1) :71–86, 1991.
- [7] Sandipan Dey. *EigenFaces and A Simple Face Detector with PCA/SVD in Python*. JANUARY 6, 2018, <https://sandipanweb.wordpress.com/2018/01/06/eigenfaces-and-a-simple-face-detector-with-pca-svd-in-python/>, 2021.
- [8] INRIA Benoît Jacob. *Eigen Library*. [https://eigen.tuxfamily.org/index.php?title=Main\\_Page](https://eigen.tuxfamily.org/index.php?title=Main_Page), 2021.