



Master 2 Imagine
Projet Image: Reconnaissance faciale

Compte Rendu 3

DEROUBAIX Renaud, SERVA Benjamin

3 novembre 2024

Table des matières

1. BDD	3
1.1 Extrait de la base de donnée d'un individu	3
2. Méthode choisie : LBPHFaceRecognizer	3
2.1 Pourquoi celle-ci ?	3
2.2 Comment l'avons nous implémenté ?	4
2.2.1 Prétraitement	4
2.2.2 Création du modèle	4
2.2.3 Reconnaissance de visage	4
2.2.4 Affichage des résultats	4
3. Résultat des tests de la méthode implémentée	5
3.1 Reconnaissance d'un visage de la BDD qui a servi à l'entraînement du modèle	5

3.2	Reconnaissance d'un visage qui fait partie de la BDD mais qui n'a pas servi pour l'entraînement du modèle	6
3.2.1	Analyse des Statistiques	6
3.3	Reconnaissance d'un visage qui ne fait pas partie de la BDD	7
4.	Conclusion sur cette méthode	8
5.	Objectif pour la semaine #3 - CR4 :	8

1. BDD

Pour la base de données, nous utilisons actuellement une base de données trouvées sur internet qui contient 40 individus différents et pour chacun individus il y a 10 photos de visages qui varient légèrement (expression du visage et angle de prise de vue).

1.1 Extrait de la base de donnée d'un individu



(a) échantillon 1



(b) échantillon 2



(a) échantillon 3



(b) échantillon 4

FIGURE 2 – exemple d'images de la BDD

On peut voir ici sur ces échantillons les différentes variations avec notamment plusieurs angles de photos, différentes expressions de visages et aussi l'ajout/suppression de lunettes.

2. Méthode choisie : LBPHFaceRecognizer

2.1 Pourquoi celle-ci ?

De toutes les méthodes que l'on a présentées dans l'état de l'art c'est celle qui avait le meilleur compromis entre facilité d'implémentation et efficacité pour la reconnaissance faciale.

2.2 Comment l'avons nous implémenté ?

On l'a implémenter en **Python** avec la bibliothèque **opencv**.

2.2.1 Prétraitement

C'est des choses que l'on n'a pas eu à faire mais il faut les prendre en compte pour ne pas fausser les résultats :

1. S'assurer que nos images sont de même tailles car dans le programme il y a une redimension automatique mais il se peut que ça déforme complètement l'image et que ça fausse les résultats.
2. Dans notre cas notre BDD ne contient que des visages et pour l'instant on ne teste qu'avec des visages aussi, mais l'objectif est de la tester sur des images quelconques donc il faudra faire un prétraitement pour détecter le visage dans l'image.

2.2.2 Création du modèle

Tout d'abord pour générer le modèle on utilisé 8 images sur 10 pour chaque individu. On a créé notre modèle grâce à la fonction **LBPHFaceRecognizer_create()** et on a ensuite ajouté toutes nos images. Enfin on a fait entraîner le modèles c'est à dire pour chaque image il a calculé un vecteur caractéristique (grâce à la méthode expliquée dans le compte rendu 2).

Pour finir on a enregistré notre modèle sous le nom **model.yml** pour pouvoir l'utiliser.

2.2.3 Reconnaissance de visage

On charge tous simplement notre modèle ainsi que l'image que l'on souhaite identifier, et on utilise la fonction **predict** qui renvoie deux valeurs qui sont les suivantes :

- étiquette (label) : pour faire simple, c'est l'identifiant de l'image associé à la base de données (celle avec le plus de correspondance)
- confiance (confidance) : c'est un indicateur sur la correspondance quand elle vaut 0 c'est que l'image est exactement la même et plus elle est élevé moins ça correspond. Pour faire plus compréhensible on a ramené cette valeur à un pourcentage.

2.2.4 Affichage des résultats

Pour avoir quelque chose de visuel on affiche l'image mise en argument du programme (image de gauche) mais aussi l'image qui lui correspond le plus dans la BDD (image de droite), enfin la correspondance en %.



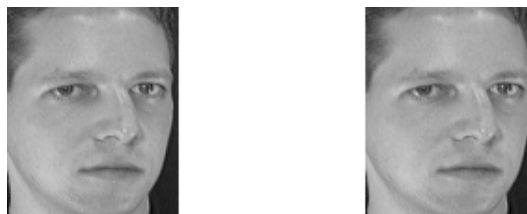
Correspondance: 28.75%

FIGURE 3 – exemple de résultat

3. Résultat des tests de la méthode implémentée

3.1 Reconnaissance d'un visage de la BDD qui a servi à l'entraînement du modèle

Logiquement on obtient une correspondance de 100% et c'est la même image qui est trouvée en correspondance.



Correspondance: 100.00%

FIGURE 4 – exemple de résultat

Bien entendu ce cas n'arrivera jamais dans un système de reconnaissance faciale et c'était seulement pour voir si l'implémentation était correcte.

3.2 Reconnaissance d'un visage qui fait partie de la BDD mais qui n'a pas servi pour l'entraînement du modèle

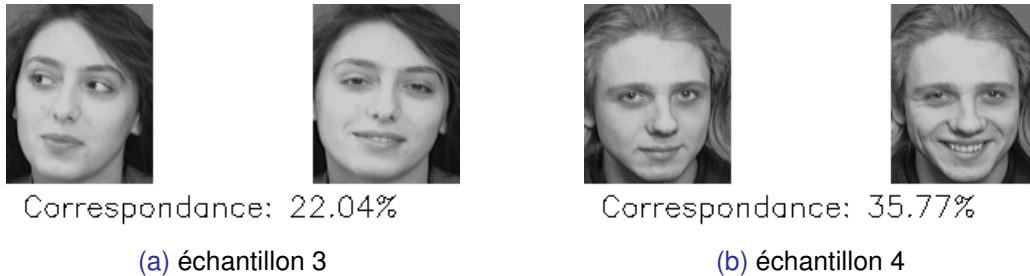


FIGURE 5 – exemple d'images de la BDD

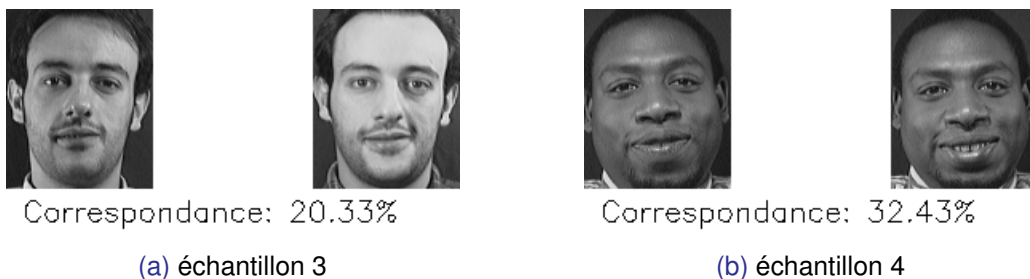


FIGURE 6 – exemple d'images de la BDD

3.2.1 Analyse des Statistiques

Pour tous ces tests on a mesuré les statistiques sur la correspondance qui sont les suivants :

- Moyenne : 30.04%
- Maximum : 42.71%
- Minimum : 7.70%
- Médiane : 32.44%

On a vu précédemment que quand on faisait le test sur un visage de la BDD qui a servi à l'entraînement du modèle, on obtient 100% de correspondance, mais pour ces tests on obtient une moyenne de 30.04% et un minimum de 7.70%. On pourrait ce dire que par exemple pour la correspondance minimum le visage testé n'appartient à la BDD, pourtant pour tous les tests le visage de correspondance trouvé appartient bien à la même personne donc on pourrait émettre l'hypothèse que quand la correspondance est positive alors le visage testé est bien dans la BDD.



FIGURE 7 – cas de la correspondance très faible

On voit que la légère inclinaison de la tête vers l'arrière influe très fortement sur la détection de correspondance.



FIGURE 8 – cas de la correspondance maximum

Pour ce cas, on aurait tendance à se dire que les différences doivent être flagrante mais pourtant elles sont à peine visible.

3.3 Reconnaissance d'un visage qui ne fait pas partie de la BDD



FIGURE 9 – exemple de résultat

On a utilisé une deuxième BDD pour faire ce test, et il s'avère qu'elle n'était pas de la même taille et qu'il a fallu les rogner et redimensionner car avec la redimension automatique le résultat change.



FIGURE 10 – exemple de résultat pour une mauvaise redimension

On voit que le résultat de correspondance est pire qu'avants mais aussi que l'image la plus proche dans la BDD est différente malgré tous notre méthode détecte que cette image ne fait pas partie de la BDD.

4. Conclusion sur cette méthode

D'après les tests que l'on a effectués on voit que cette méthode est efficace malgré tous il serait intéressant d'effectuer des tests plus poussés pour mieux identifier ses limites notamment sur les variations d'angles de vues. Avec ce qu'on a observés on se doute que pour un système de détection automatique où la caméra se trouve en hauteur, cette méthode sera inapte.

5. Objectif pour la semaine #3 - CR4 :

Nos objectifs d'ici le prochain compte rendu sont :

- Faire des tests plus approfondis de la méthode implémenté.
- Développer une base de données personnels (avec des personnes du master ou autres).
- Commencer à se pencher sur la méthode avec CNN.

Sources

1. [BDD de visages](#)

3.3 [Deuxième BDD de visages](#)

Github

[Lien Github](#) avec tout le code et compte rendu du projet.