



Master 2 Imagine
Projet Image: Reconnaissance faciale

Compte Rendu 2

DEROUBAIX Renaud, SERVA Benjamin

27 octobre 2024

Table des matières

1. État de l'art :	3
1.1 Définition : Extraction de Caractéristiques	3
1.2 Méthodes dites classiques (sans l'utilisation de CNN)	3
1.2.1 Haar Cascades	3
1.2.2 Histogram of Oriented Gradients (HOG)	3
1.2.3 LBPHFaceRecognizer	4
1.2.4 EigenFace Recognizer	4
1.3 Techniques de comparaisons	5
1.3.1 Distances Euclésiennes	5
1.3.2 Cosinus	5
1.3.3 Comparaison	5
1.4 Méthodes utilisant des CNN	6
1.4.1 DeepFace	6
1.4.2 FaceNet	7
1.4.3 ArcFace	7

2. Comparaison de méthode → performance :	9
3. Objectif pour la semaine #3 - CR3 :	10

1. État de l'art :

1.1 Définition : Extraction de Caractéristiques

La reconnaissance par extraction de caractéristiques consiste à identifier des points clés ou des motifs distinctifs dans une image de visage, puis à utiliser ces informations pour reconnaître ou différencier les individus. Les algorithmes d'extraction de caractéristiques comparent les vecteurs de caractéristiques pour effectuer des correspondances. Cette approche dépend de la qualité des caractéristiques extraites.

1.2 Méthodes dites classiques (sans l'utilisation de CNN)

1.2.1 Haar Cascades

1. Détection de motifs de base : Chaque caractéristique Haar analyse une région en comparant les pixels sombres et clairs, identifiant des motifs comme les contours des yeux, du nez, ou de la bouche.
2. Sélection de caractéristiques : Puisqu'il existe un très grand nombre de caractéristiques Haar possibles, un algorithme de sélection (comme AdaBoost) est utilisé pour choisir les plus pertinentes. AdaBoost renforce les caractéristiques qui différencient le mieux les visages des autres objets, optimisant ainsi la précision tout en réduisant le nombre de calculs.
3. Cascade de caractéristiques : La détection se fait avec une cascade de classifieurs, ce qui signifie qu'une image passe par plusieurs étapes où seules les régions contenant des motifs caractéristiques avancent dans la cascade. Cela permet un traitement rapide, car les régions sans caractéristiques de visage sont écartées dès les premières étapes.

1.2.2 Histogram of Oriented Gradients (HOG)

1. Prétraitement : L'image est d'abord redimensionnée pour homogénéiser les tailles et les proportions.
2. Calcul des gradients : Le gradient est calculé pour chaque pixel en mesurant les changements d'intensité horizontaux et verticaux. Cela permet de détecter les contours, essentiels pour représenter les formes d'un visage.
3. Orientation et magnitude des gradients : Pour chaque pixel, on calcule l'angle (orientation) et la magnitude du gradient. Cela permet d'identifier les bords marqués comme les yeux, le nez et la bouche.
4. Bloc de cellules et histogrammes : L'image est divisée en petites régions appelées cellules. Dans chaque cellule, on crée un histogramme des orientations des gradients. Chaque histogramme regroupe les orientations similaires dans des "buckets", et leur contribution est pondérée par leur magnitude.

5. Normalisation : Pour obtenir des caractéristiques indépendantes de l'éclairage et du contraste, chaque bloc de cellules est normalisé. Cela réduit les variations dues aux changements de luminosité et améliore la robustesse de la détection.
6. Création de la signature HOG : Les histogrammes normalisés des blocs sont concaténés pour créer un vecteur de caractéristiques HOG, représentant l'image de manière simplifiée et stable.

1.2.3 LBPHFaceRecognizer

1. Division en Pixels : L'image est convertie en niveaux de gris, et chaque pixel est comparé à ses voisins dans un rayon donné.
2. Calcul des Local Binary Patterns (LBP) : Pour chaque pixel central, un motif binaire est créé : si le pixel voisin est supérieur ou égal au pixel central, il reçoit une valeur de 1, sinon, il reçoit 0. Ces valeurs binaires forment un nombre binaire unique pour chaque pixel.
3. Création de l'Histogramme : L'image est divisée en petites cellules, et un histogramme de motifs LBP est calculé pour chaque cellule, ce qui permet de capturer les variations de texture locales.
4. Création du Modèle LBPH : L'ensemble des histogrammes pour chaque cellule est concaténé pour former un vecteur qui représente le visage entier.

1.2.4 EigenFace Recognizer

1. Préparation des Images : Les images de visages sont normalisées (généralement en niveaux de gris) pour garantir que chaque image est de même dimension et que l'éclairage, l'échelle et la position sont similaires.
2. Construction de la Matrice des Vecteurs de Visages : Chaque image est aplatie en un vecteur 1D, et ces vecteurs sont combinés pour former une grande matrice où chaque colonne représente un visage de l'ensemble d'entraînement.
3. Application de l'Analyse en Composantes Principales (PCA) : PCA est appliqué à la matrice des visages pour déterminer les vecteurs propres et valeurs propres. Les vecteurs propres qui capturent le mieux les variations entre les visages deviennent les eigenfaces. Ces eigenfaces sont les composantes principales qui définissent les caractéristiques essentielles des visages. En gardant uniquement les eigenfaces avec les valeurs propres les plus élevées, on réduit la dimension des données tout en conservant les informations les plus distinctives.
4. Projection des Images : Chaque visage de l'ensemble d'entraînement est projeté dans cet espace de faible dimension, formé par les eigenfaces, et est représenté par un "vecteur des poids". Ce vecteur des poids capture les contributions de chaque eigenface pour reconstruire l'image d'origine.

1.3 Techniques de comparaisons

Ces techniques vont devoir être utilisées dans un second temps, c'est à dire après avoir extrait des caractéristiques des images pour pouvoir comparer ceux ci entre eux.

1.3.1 Distances Euclésiennes

Elle mesure la "distance directe" entre deux points dans un espace multidimensionnel. Pour deux vecteurs A et B, elle est calculée comme la racine carrée de la somme des carrés des différences de chaque dimension :

$$d(A, B) = \sqrt{\sum (A_i - B_i)^2}$$

1.3.2 Cosinus

Elle mesure l'angle entre deux vecteurs, ce qui indique leur similarité en termes de direction plutôt qu'en termes de magnitude. Elle est définie comme :

$$\text{sim}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

Si le cosinus est proche de 1 alors les vecteurs sont similaires en direction.

1.3.3 Comparaison

La distance euclidienne est plus sensible aux différences de longueur, tandis que la distance cosinus se concentre sur la similarité de forme, souvent plus robuste pour des vecteurs d'encodage de visages.

Malgré tous il serait intéressant de combiner ces deux distances et de chercher à les minimiser.

1.4 Méthodes utilisant des CNN

1.4.1 DeepFace

DeepFace se base sur une classification softmax pour obtenir des embeddings. Son objectif est de projeter les visages dans un espace d'embedding R^d où les images de la même personne se regroupent et celles de personnes différentes se séparent, afin d'obtenir la vérification et l'identification des visages.

Pour chaque visage, DeepFace applique une transformation géométrique avant de l'envoyer dans le réseau de neurone. Cet alignement se fait en 3 étapes :

1. **2D Alignement** → 6 points à l'intérieur de la zone de détection centrés sur les yeux, le bout du nez et la bouche.
2. **3D Alignement** → 67 points correspondant à une référence 3D.
3. **Frontalization** → compense les variations de pose et d'angle, garantissant que chaque visage entre dans le réseau dans une position standardisée.

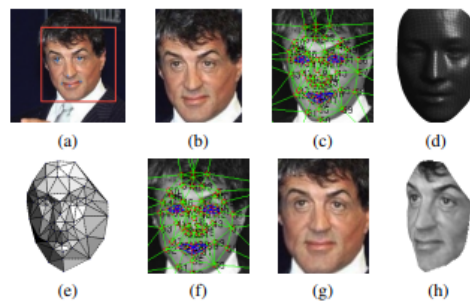


FIGURE 1 – Alignement de visage

Ce processus d'alignement rend le modèle plus robuste aux variations d'angle et de pose.

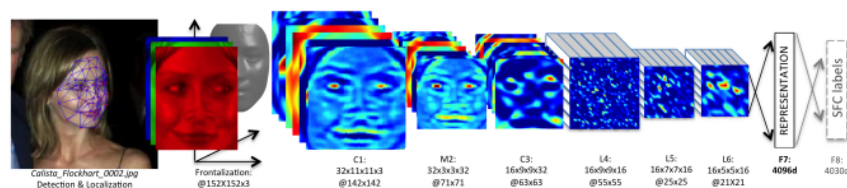


FIGURE 2 – Architecture DeepFace

1.4.2 FaceNet

FaceNet est un modèle de reconnaissance faciale utilisant le 'triplet loss' afin d'obtenir de la vérification, reconnaissance et regroupement de visage.

On cherche à avoir un embedding $f(x)$ d'une image x dans un espace R^d afin d'avoir une distance minimale pour les visages similaires quelle que soit l'image et à l'opposé, une distance élevée pour des visages de personne différente.

le **Triplet Loss** est la fonction qui permet au modèle d'apprendre cet embedding. Il fonctionne de la manière suivante :

On cherche à ce que toute images x_i^a (anchor) d'une personne spécifique soit plus proche de toute les image x_i^p (positive) de la même personne que n'importe quelle image x_i^n (negative) de n'importe quelle personne.

On veut $L = \sum_i^N \max(\|f(x_i^a) - f(x_i^p)\|^2 - \|f(x_i^a) - f(x_i^n)\|^2 + \alpha, 0)$ avec α une marge pour éviter une distance trop petite entre x_i^a et x_i^p & f le réseau de convolution.



FIGURE 3 – Triplet Loss

De nombreux triplets sont facilement satisfaits, ils n'apporteront donc rien à l'entraînement du modèle et ralentiront la convergence des résultats. C'est pourquoi, il est nécessaire de choisir des 'hard triplet' afin d'améliorer le résultat du modèle.

La **sélection de triplet** se fait en prenant des 'hard positive' (x_i^p éloigné de l'anchor x_i^a) ainsi que des 'hard negative' (x_i^n proche de l'anchor x_i^a). Il est impossible de parcourir l'ensemble des images pour trouver les 'hard positive' et 'hard negative', c'est pourquoi la solution est de chercher le argmin et argmax dans un sous-ensemble de l'ensemble de données d'entraînement ou encore un mini-batch des données d'entraînement.

1.4.3 ArcFace

ArcFace est un modèle de reconnaissance faciale qui utilise une fonction de perte angulaire, appelée Additive Angular Margin Loss (AAM). Son objectif est de positionner chaque visage dans un espace d'embedding R^d de façon à maximiser la distance angulaire entre les personnes différentes, tout en minimisant celle entre les images d'une même personne.

Dans ArcFace, on cherche à ce que chaque embedding soit séparé de façon optimale en termes d'angle, en forçant une marge fixe entre les différentes classes dans l'espace angulaire. Cela permet au modèle de mieux généraliser pour des variations

d'éclairage, d'angle et d'expression dans les images faciales.

Le reste de l'architecture est similaire à une fonction softmax classique. La fonction de perte angulaire s'appuie sur une version modifiée de la softmax avec l'ajout de cette marge angulaire. Cet ajout fait en sorte que la séparation des classes dans l'espace d'embedding se base sur une distance angulaire plutôt qu'euclidienne, renforçant ainsi la discrétisation des classes tout en conservant l'efficacité et la simplicité de la softmax dans le processus de classification.

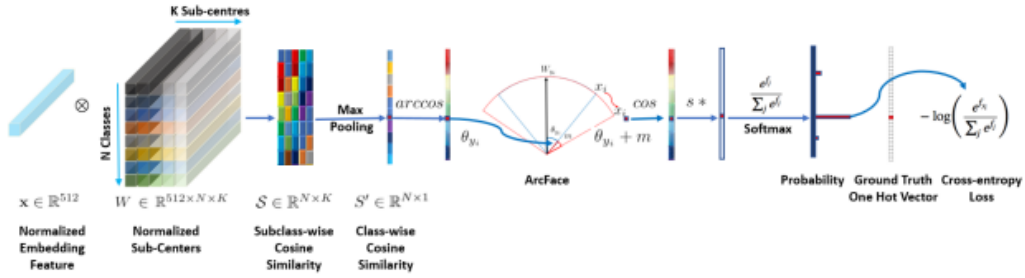


FIGURE 4 – Architecture ArcFace

2. Comparaison de méthode → performance :

Pour cette comparaison, nous allons étudier seulement les solutions open-source.
Les méthodes mélange de la détection de visage et de la reconnaissance faciale.
Les précisions présentées proviennent des tests et expérimentations effectués par les auteurs des méthodes





Méthode	Type d'ap-proche	Précision	Base de données de test	Avantage	Inconvénient	Lien
FaceNet	CNN	99,05% / 99,65% (± 0.00252)	CASIA-WebFace / VGGFace2	Haute précision, même en cas de variations d'expressions d'angle.	/	
OpenFace	CNN	92%	LFW	Optimisé pour les applications en temps réel.	Précision réduite en cas de changements importants de lumière	
ArcFace	CNN	99.7+%	LFW, VGGFace2	Excellente précision pour les variations d'angles grâce à l'alignement angulaire.	/	
InsightFace	CNN	97,3%	IJB-C, Glint360K	Bonne robustesse aux variations d'angles. Implémente ArcFace ainsi que d'autre méthode.	/	

TABLE 1 – Comparaison des méthodes de reconnaissance faciale open-source



Méthode	Type d'ap-proche	Précision	Base de données de test	Avantage	Inconvénient	Lien
Ageitgey/face_reco-gnition	Basé sur FaceNet, dlib	99.38%	Labeled Faces in the Wild (LFW)	Facilité d'utilisation et bonne précision sur les données d'entraîne-ment standard.	/	
Deepface	Multi-backend CNN (VGG-Face, FaceNet, OpenFace, Deepface, etc)	97.35% ± 0.25%	LFW, VGGFace2	/	/	
EigenFace Recognizer	Analyse en Composantes Principales (OpenCV)	/	/	Simple et rapide, adapté aux petites bases de données.	Faible précision avec des variations de pose et de lumière importantes	Open CV
LBPHFace Recognizer	Caractéristiques locales (OpenCV)	/	/	Résistant aux variations d'éclairage, performances en temps réel possibles.	Précision limitée, sensible aux changements de pose	open CV

TABLE 2 – Comparaison des méthodes de reconnaissance faciale open-source

3. Objectif pour la semaine #3 - CR3 :

Nos objectifs d'ici le prochain compte rendu sont :

- Choisir une des méthodes dites classique (sans CNN) présentées dans l'état de l'art et l'implémenter.
- Mettre en place une BDD pour les tests.
- Tester la méthode implémentée sur notre BDD

— Analyser les résultats et conclure sur l'efficacité de la méthode

Sources

1.2.3 Face Recognition: Understanding LBPH Algorithm
1.2.4 OpenCV Eigenfaces for Face Recognition
1.4.1 DeepFace: Closing the Gap to Human-Level Performance in Face Verification
1.4.2 FaceNet: A Unified Embedding for Face Recognition and Clustering
1.4.3 ArcFace: Additive Angular Margin Loss for Deep Face Recognition
2. Face Detection Explained: State-of-the-Art Methods and Best Tools

Github

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