



Master 2 Imagine
Projet Image: Reconnaissance faciale

Compte Rendu 4

DEROUBAIX Renaud, SERVA Benjamin

10 novembre 2024

Table des matières

1. Rectification de certaines conclusion du rapport précédent	2
2. Version Amélioré	3
2.0.1 Explication	3
2.0.2 Tests	3
3. Choix de la méthode CNN	4
3.1 Pourquoi celle-là ?	4
4. Création de la base de donnée	5
5. Objectif pour la semaine #6 - CR5 :	6

1. Rectification de certaines conclusion du rapport précédent

On avait tiré la conclusion que lorsque notre coefficient de correspondance était positive alors on pouvait estimer que la reconnaissance faciale était valide. De plus on avait dit que sur tous les tests de correspondance effectués que tous ont été bons, seulement on s'est trompé.

Pour les 80 tests effectués il y en a un qui a échoué et c'est le suivant :



FIGURE 1 – mauvaise correspondance

Pourtant le pourcentage de correspondance calculé est supérieur au minimum qu'on avait obtenu sur tous les tests. De plus dans l'état de l'art on avait noté les différents avantages/inconvénients des différentes méthodes existantes et pour la méthode LBPH-FaceRecognizer on y avait noté sa sensibilité aux changements de poses. Dans la BDD, pour cette personne, il n'y a effectivement aucune photo avec cette pose. Or le fait que l'autre photo a une position/angle de vue fortement similaire, notre programme a calculé une meilleure correspondance.

En changeant cette photo test avec une de la BDD concernant cette personne pour qu'on ait dans le modèle toutes les poses différentes alors cette fois-ci on obtient bien 100% de bonne correspondance.

Mais on tire quand même en conclusion que le seuil de bonne correspondance qu'on avait fixé n'est pas bon on pourrait maintenant émettre l'hypothèse que si celui-ci est supérieur à 10% alors il y a une correspondance. Mais il reste quand même plus de test à faire pour l'affiner.

Malheureusement il va être dur malgré les tests de déterminer un seuil de correspondance auquel on peut se fier donc il serait plus intéressant de changer la méthode que l'on utilise actuellement.

2. Version Amélioré

2.0.1 Explication

Comme précédemment on va crée le modèle avec LBPHFaceRecognizer sauf que maintenant on va simplement utiliser le modèle sans utiliser la fonction predict.

Pour chaque label du modèle il y a certains nombres d'histogrammes/vecteurs caractéristiques (un pour chaque image associé à une personne, 8 dans le cas de la BDD que l'on utilise) et on va les extraire et faire un vecteurs moyens pour chaque label.

Enfin on fera une comparaison de distance euclidienne entre le vecteur caractéristiques de l'image que l'on teste et les vecteurs moyen.

2.0.2 Tests

En faisant les même tests que pour la première méthode et en étudiant les statistiques on obtient ces valeurs :

- Moyenne : 1.21
- Max : 1.64
- Min : 0.83
- Médiane : 1.22

Quand on test avec des images de personnes non présentes dans la bdd alors on obtient des valeurs aberrantes aux alentours de 5.

Avec cette méthode contrairement à la précédente où on pouvait obtenir des pourcentages négatifs il est beaucoup plus simple de déterminer si la personne testé est présente dans la BDD, on n'a pas besoin de faire une multitude de test pour affiner le seuil de correspondance.

3. Choix de la méthode CNN

Lors de l'état de l'art, nous avons étudié plusieurs méthodes CNN envisageables, principalement ArcFace, DeepFace, et FaceNet. Après réflexion, nous avons décidé d'utiliser **InsightFace** (qui utilise ArcFace pour l'apprentissage des embeddings) en combinaison avec une architecture *MobileFaceNet*, dans l'optique de développer une application mobile. InsightFace offre une précision élevée et est bien adapté aux contraintes de calcul et de mémoire des appareils mobiles.

De plus, on souhaite entraîner notre propre modèle (extraction d'embedding/caractéristique) plutôt qu'utiliser un modèle pretrained. L'avantage d'utiliser un modèle pretrained est qu'il est plus simple d'utilisation, étant donné qu'il utilise de grands datasets avec des personnes fortement différentes, la précision est très élevée et l'extraction de caractéristique est robuste. Mais en entraînant notre propre modèle sur notre propre base d'image, on obtiendra des résultats personnalisés et très adaptés à nos besoins, on aura plus de contrôle sur notre architecture. Le potentiel gros problème pourrait venir du manque de diversité venant de notre base de données.

3.1 Pourquoi celle-là ?

Le choix d'InsightFace dans l'optique d'une application mobile de reconnaissance faciale a les avantages suivants :

- **Performance et précision** : InsightFace utilise des techniques modernes d'embedding telles que *ArcFace* et *CosFace*. Ces techniques permettent une représentation des visages robuste, essentielle pour garantir une bonne reconnaissance faciale, dans n'importe quelles conditions d'éclairage ou d'angle.
- **Modèles pour mobile** : InsightFace propose des architectures CNN optimisées, comme *MobileFaceNet*, qui sont spécifiquement conçues pour fonctionner sur des appareils avec des capacités de calcul limitées. MobileFaceNet permet de réduire la taille du modèle et la charge de calcul tout en maintenant une précision élevée.
- **Entraînement personnalisé** : InsightFace permet d'entraîner des modèles sur des datasets personnalisés, ce qui nous permettrait d'entraîner un modèle sur notre propre banque d'image afin d'avoir de meilleurs résultats sur les visages souhaités.
- **Optimisation pour le quasi temps réel** : Les modèles proposés par InsightFace sont bien adaptés pour de la reconnaissance en temps quasi réel.

InsightFace est le choix le plus adapté pour notre application mobile de reconnaissance faciale en raison de sa précision, de son efficacité sur des dispositifs mobiles et de sa flexibilité en termes d'entraînement et de déploiement.

4. Création de la base de donnée

Pour le projet, on aimerait faire un modèle entraîné sur notre propre banque d'image composé principalement d'étudiant du master volontaire, connaissance personnelle et peut-être des célébrités avec des images libres de droits sur Internet pour élargir le modèle (genre/ethnie)

Pour cela, il faut mettre en place notre propre base de données avec des images traitées et labellisées.

Après analyse des datasets utilisés pour des modèles pré-entraîné, on a décidé d'avoir des images de visage de taille 112*112 pixels.

Utiliser des photos personnelles pourrait poser un problème de qualité dû aux variations dans les images, ce qui pourrait diminuer la précision des résultats. C'est pourquoi, si nécessaire, nous pourrions entraîner le modèle à l'aide de **Sub-center ArcFace**, qui est plus robuste face aux variations intra-classe (comme l'angle, l'éclairage et la résolution des images). Ensuite, à l'aide de la **Knowledge Distillation**, nous pourrions simplifier ce modèle vers un modèle MobileFaceNet optimisé pour une application mobile, en conservant les caractéristiques apprises par Sub-center ArcFace pour une meilleure performance.

Quelques exemples d'images de notre base de données (étudiants du master), parmi la collection d'environ 90 images récoltées :



FIGURE 2 – Extrait d'imagette de la base de donnée de visage

En fonction des résultats observés lors de nos premiers essais de reconnaissance faciale, différents pré-traitements pourraient être envisagés, tels que le redimensionnement (différent de l'actuel), l'alignement des visages, la normalisation de l'éclairage et la conversion en niveaux de gris, etc.

5. Objectif pour la semaine #6 - CR5 :

Nos objectifs d'ici le prochain compte rendu sont :

- Continuer ce qu'on a commencé sur la version CNN.

Sources

3. [InsigthFace](#) code source

Github

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