

Compte Rendu Semaine 6

Accès sécurisé par reconnaissance faciale

Josua Philippot - Félix Yriarte
Master 2 IMAGINE

Octobre 2021 - Décembre 2021



Table des matières

1	Contexte	2
2	Méthode par CNN	2
3	Seuil pour nos méthodes déjà développées	2
4	Application avec GTK+	3
5	Perspectives	3
6	Nos sources	3

1 Contexte

Nous avons axé notre travail cette semaine sur trois points :

- le développement de notre méthode par CNN
- la recherche de "bonnes valeurs" de seuil pour nos méthodes (courbes ROC)
- la mise en place d'une interface graphique pour présenter nos méthodes et résultats

En effet, même s'il nous semble plus judicieux de mettre en priorité en place un réseau CNN adapté à la reconnaissance faciale, cette méthode n'est pas utilisable sur les ordinateurs de la faculté, nous avons donc décidé d'avancer sur ces autres points lors de nos sessions en présentiel.

2 Méthode par CNN

Pour mettre en place cette méthode, il est courant d'utiliser de l'accélération GPU, c'est à dire d'effectuer une partie des calculs sur GPU. Il est pour cela nécessaire d'installer certains programmes (CUDA, cuDNN), ce qui n'est pas faisable sur les ordinateurs de la faculté.

Nous pensons, comme nous avons pu le voir dans la littérature, utiliser un réseau déjà entraîné sur une importante base de données (ImageNet), puis le spécifier à notre tâche, en ne ré-entraînant qu'une faible partie des couches finales du réseau (aussi appelé *fine-tuning*). N'ayant qu'un faible volume d'images pour entraîner le réseau, il ne nous semble pas envisageable d'entraîner intégralement un CNN.

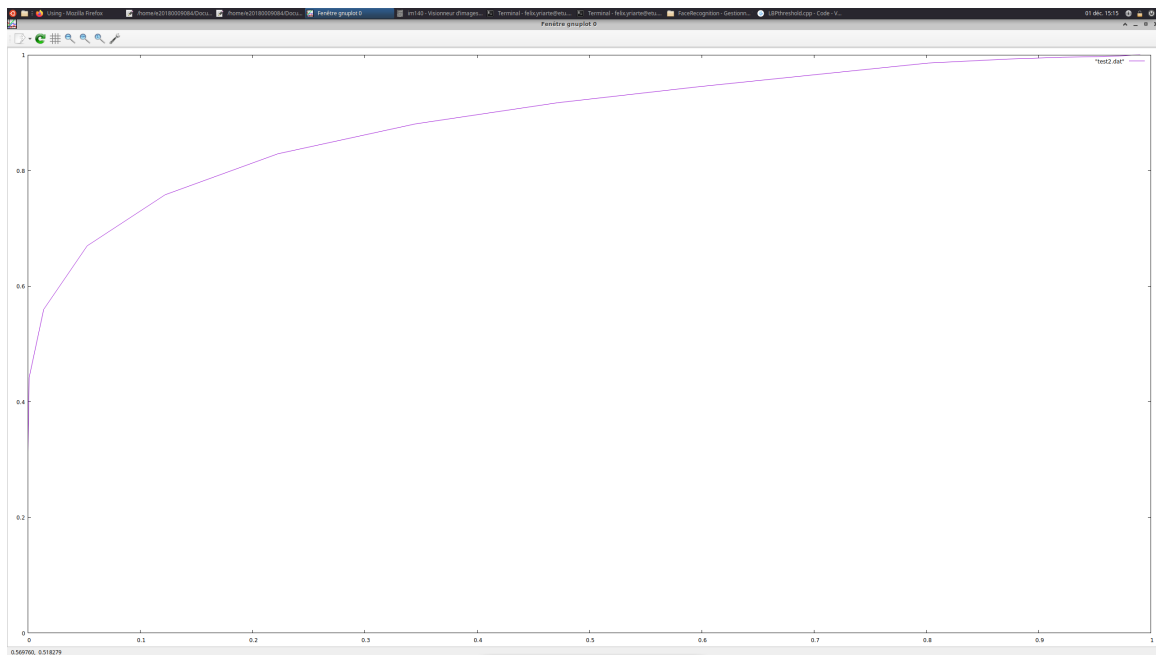
Dans l'optique de nous familiariser avec les CNNs, nous mettons en place dans un premier temps des réseaux simples, afin de mieux connaître les outils que nous utiliserons pour en place le réseau pour la reconnaissance faciale.

3 Seuil pour nos méthodes déjà développées

Afin de déterminer si une image de visage de requête correspond bien à une identité stockée dans notre base, il est nécessaire de déterminer une distance maximale séparant deux identités constituant une bonne mise en correspondance.

Pour trouver ce seuil optimal, nous avons tracé une courbe ROC, représentant la précision du modèle, en fonction de son anti-spécificité. On cherche alors la valeur de seuil offrant des valeurs de précision et d'anti-spécificité les plus proches du point (0,1) sur la courbe.

Ci-dessous, la courbe obtenue sur la base d'images AT&T, avec notre méthode par LBPH.



La courbe obtenue indique que la méthode utilisée est meilleure qu'une classification aléatoire. Nous obtenons effectivement une précision de 75% et une spécificité de 88%. Ces résultats sont obtenus pour notre méthode sans utilisation de détection de visage par cascades de Haar, nous nous attendons en effet à avoir de meilleurs résultats avec cette dernière.

4 Application avec GTK+

En vue de mettre en place une application depuis laquelle on pourrait lancer une reconnaissance utilisant nos différentes méthodes, nous avons cherché une librairie nous permettant de faire une interface graphique, si possible mutli-plateforme. Notre choix c'est porté sur GTK+, la librairie en C open-source développée par *GNOME Foundation* et utilisée par de nombreuses applications de nos jours. Durant cette semaine nous souhaitons apprendre le fonctionnement particulier de cette librairie et produire ladite application.

5 Perspectives

Nous pouvons donc avancer sur différents points :

- Création d'un poster A1 faisant état de nos méthodes implémentées
- Documentation CNNs, implémentation utilisant docker et Tensorflow
- Comparaisons différentes méthodes

6 Nos sources

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

Références