

# THE MACHINE

Notice

Nouidri Sami

12/05/2024

## Table des matières

Table des matières.....	1
Abstract.....	2
Traitement.....	3
Pipeline DeepFace .....	3
Pipeline Edge Detection .....	4
Resultats.....	5
Detection de lunettes.....	5
Conclusion .....	7

**Tous les termes utilisés dans ce rapport doivent être interprétés au sens épïcène**

## Abstract

L'objectif de ce projet est de concevoir un système de détection de certains attributs du visage. Ayant comme entrée une image ou une vidéo d'un visage, le système doit être capable de détecter les attributs suivants :

- La présence ou non de lunettes
- L'origine de la personne
- Le sexe de la personne

Inspiré de systèmes de surveillances existants, le programme pourrait servir par exemple à classer une liste de portraits en fonction de ces attributs.

# Traitement

## Pipeline DeepFace

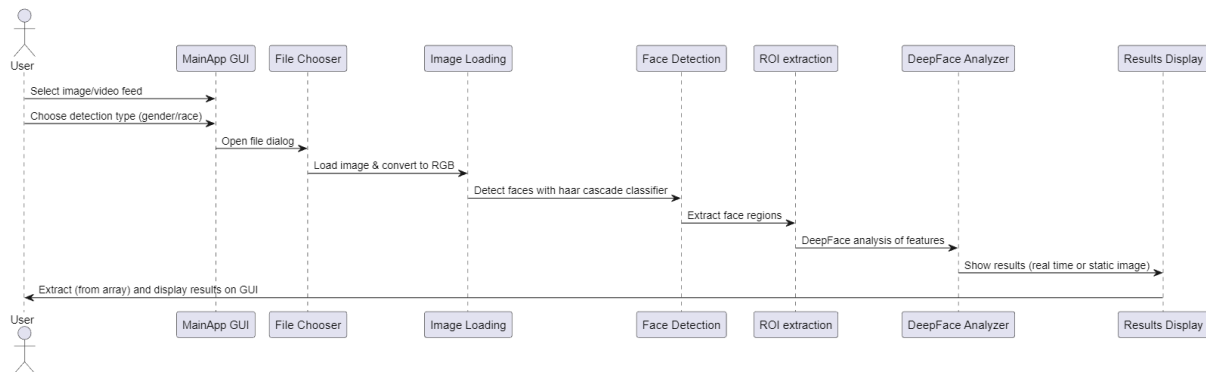


Figure 1 : schéma de la pipeline deepFace simplifiée

Les attributs de sexe et d'origine sont déterminée à l'aide du modèle DeepFace de Facebook. Tout d'abord, on commence par détecter la présence de visages à l'aide d'un classificateur Haar spécialement prévu pour (source inclus dans le code).

Une fois les coordonnées externes du visages extrait à l'aide de la fonction detectMultiscale(), on effectue du slicing pour extraire le visage lui-même (plutôt que le carré externe qui l'englobe).

Si l'on travaille avec une image statique, c'est à ce moment là que l'on doit la convertir en RGB pour que DeepFace puisse la traiter (la VideoCapture de la webcam n'a pas besoin d'être convertit).

DeepFace va ensuite analyser l'image et essayer de prédire l'origine ou le sexe de la personne, dépendant des paramètres d'entrées du programme. En interne, le modèle construit un modèle 3D du visage avec un modèle approximatif et se base dessus pour la prédiction ( voir la recherche <sup>1</sup>de Facebook pour plus d'informations).

Une fois les resultats obtenues (sous forme d'un dictionnaire clé – valeur), Le programme va simplement extraire le sexe dominant (pourcentage le plus haut) ou les trois ethnies dominantes (les trois pourcentages les plus élevées).

En general, DeepFace est plutôt confiant sur le sexe (même quand il se trompe) et attribut des pourcentages presque binaire (94 – 99% pour le dominant), tandis que les ethnies ont des pourcentages distribuées avec une plus petites variances (33%, 28% et 12% par exemple).

Finalement, on affiche ces resultats sur les bords du visage détecté à l'aide d'un simple putText().

L'implémentation temps réel de cet algorithme (GenderDetection.py) comporte la même logique, ainsi que des routines de multi-threading et de traitement de frame. En sommes, le programme boude et traite chaque frame de la VideoCapture (plutôt que de traiter qu'une image en entrée).

<sup>1</sup> <https://research.facebook.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/>

## Pipeline Edge Detection

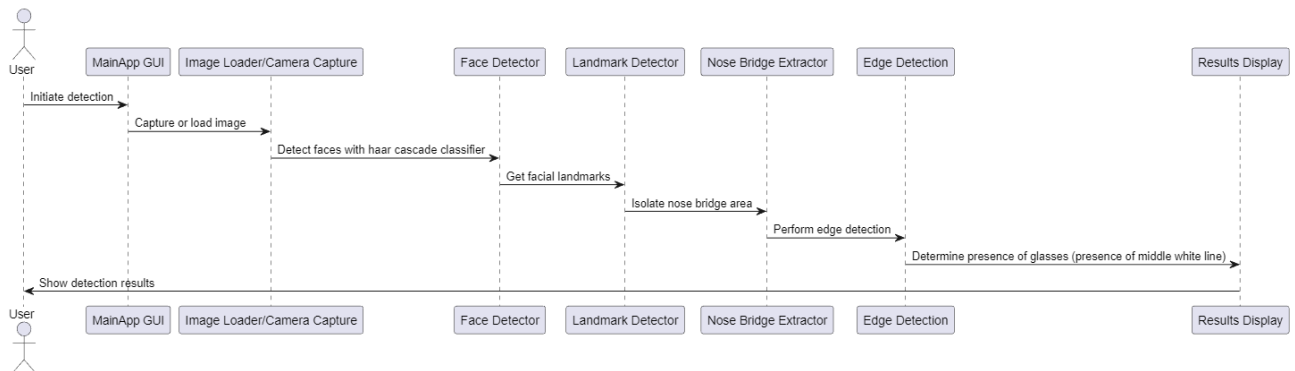


Figure 2 : schéma de la pipeline Edge Detection simplifiée

Tout d'abord, tout comme pour DeepFace, on commence par détecter la présence de visages à l'aide d'un classificateur Haar spécialement prévu pour (source inclus dans le code). Cependant, on utilise la fonction `get_frontal_face_detector()` de la bibliothèque de machine learning Dlib, plutôt que les fonctions cascade d'OpenCV.

Une fois le visage détecté avec le classificateur de points de repères, on isole le pont du nez du visage afin de le traiter séparément, en effectuant une slice des cases 27 – 36 du tableaux « landmarks » (points de repères).

Une fois le pont isolé, on produit une image binaire de cette région. Ceci est fait en appliquant un Gaussian blur pour obtenir des edges plus grossières, ensuite la transformation en binaire avec la fonction Canny.

Finalement, on s'intéresse au centre de l'image binaire. Si ce dernier contient du blanc (255), on estime qu'il y a de forte probabilité que la personne porte des lunettes.

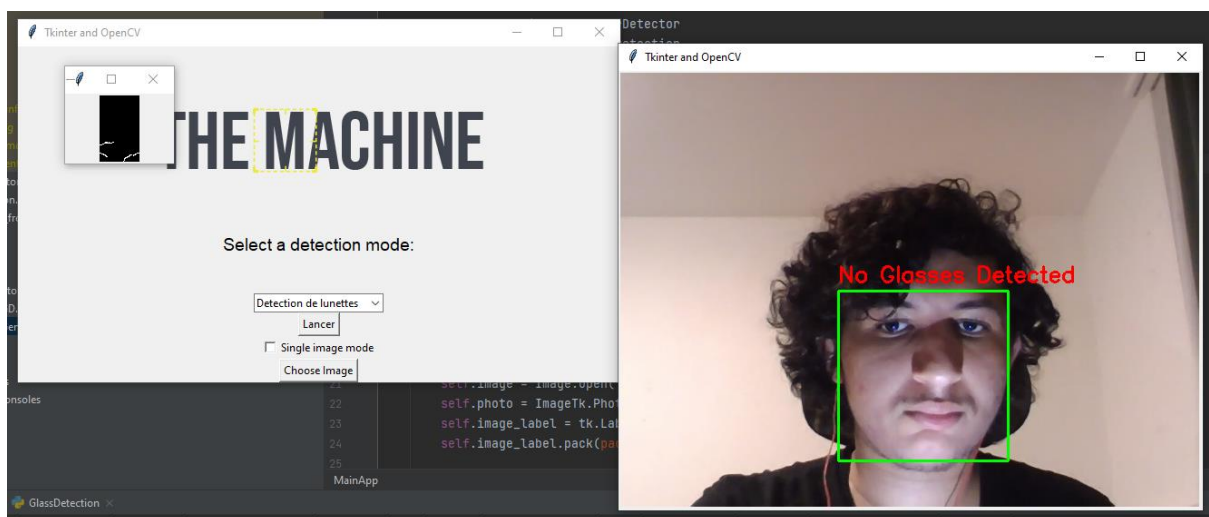


Figure 3 : exemple des edges détecté autour de la région du nez

## Resultats

### Detection de lunettes

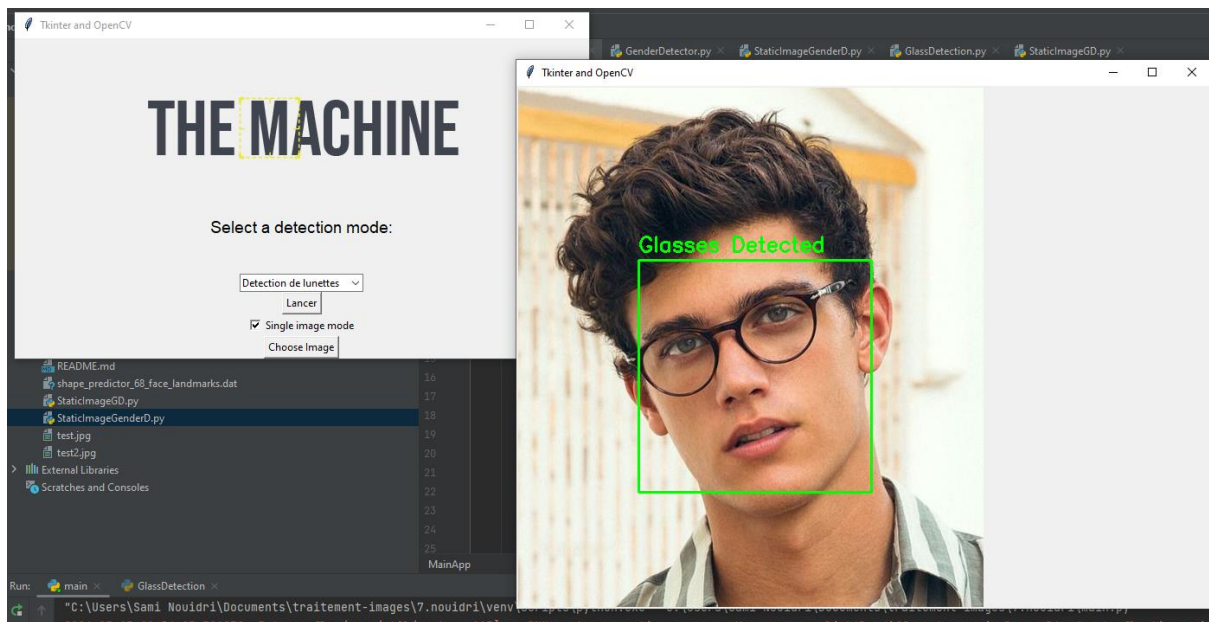


Figure 4 : Detection de lunettes sur une image statique

La detection de lunettes est celle qui produit le resultat le moins fiable. Due a la méthode de detection, le sujet doit-être directement en face de la caméra, illuminé de manière uniforme (trop d'ombre d'un coté ou de l'autre produit des edges indésirables), et portant un cadre dont le pont est assez épais comme sur l'illustration ci-dessus.

En effet, si un de ces facteurs n'est pas idéal, la detection devient moins fiable.

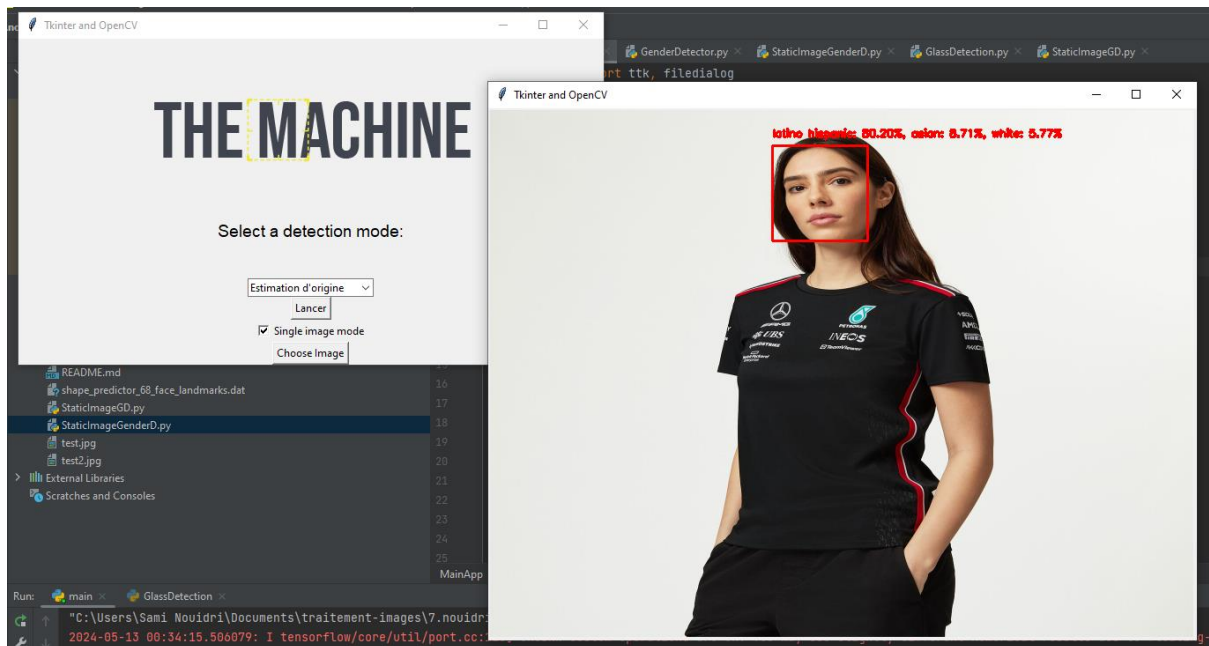


Figure 5: Detection d'origine

Comme mentionnée auparavant, la détection d'origine affiche des pourcentages plutôt mitigés en général. DeepFace semble beaucoup mieux gérer les différences de luminosité et d'orientation du visage (grâce à leur technologie de re-construction faciale 3D), ce qui le rend plus fiable dans la plupart des scénarios.

Cependant, contrairement à l'illustration, le résultat est plus souvent une estimation des trois ethnies les plus probables, plutôt qu'une réponse définitive.

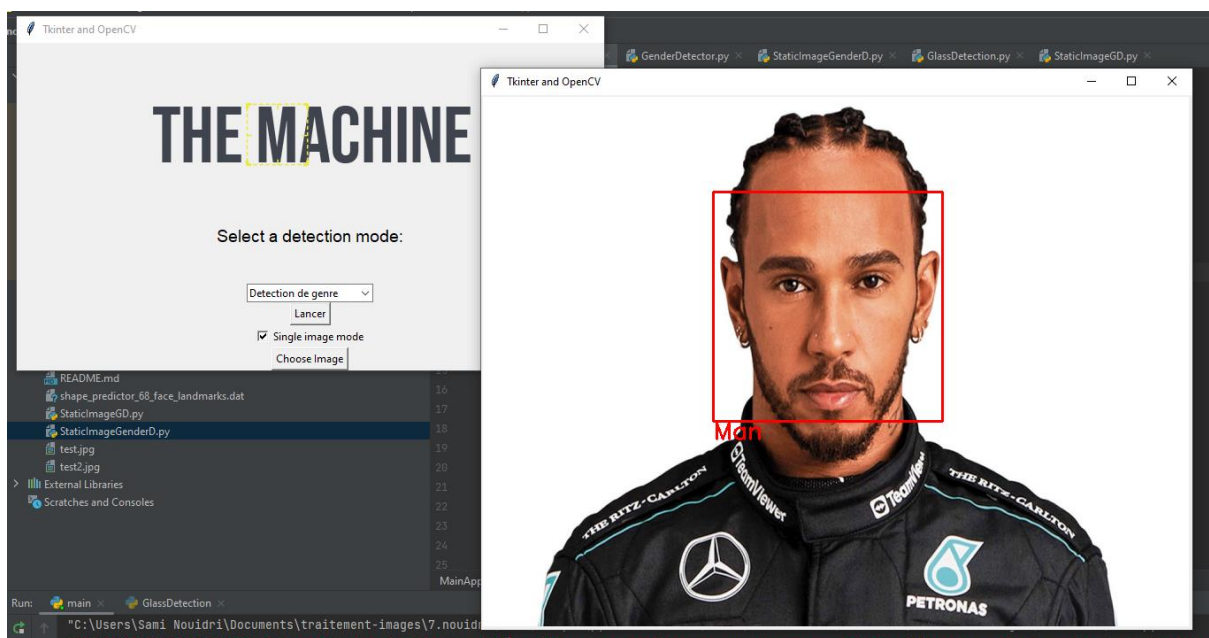


Figure 6 : Detection de genre

Quant à elle, la détection de genre produit des pourcentages plutôt définitifs, malgré que le modèle semble parfois confondre le genre si la personne n'a pas d'attributs très distincts (p. ex. : une barbe chez l'homme, mâchoire arrondie / maquillage chez les femmes, etc.)

## Conclusion

Les objectifs mis en place durant la phase de conception du projet ont été atteints avec différents degrés de succès. En effet, plusieurs améliorations, notamment au niveau de la détection de lunettes, peuvent encore être effectuées.

Il serait par exemple envisageable d'essayer de détecter la présence d'un cadre autour des yeux, afin de minimiser les erreurs dues à une lumière non-homogène sur le visage.

Finalement, on pourrait aussi envisager d'implémenter une analyse contextuelle de ces informations, ainsi que d'autres attributs (DeepFace permet notamment d'analyser les émotions des visages), afin d'en tirer des conclusions dans certains scénarios.