|  |
| --- |
|  |
| The machine |
| Notice |

|  |
| --- |
| Nouidri Sami  12/05/2024 |

# Table des matières

[Table des matières 1](#_Toc166453552)

[Abstract 2](#_Toc166453553)

[Traitement 3](#_Toc166453554)

[Pipeline DeepFace 3](#_Toc166453555)

[Pipeline Edge Detection 4](#_Toc166453556)

[Resultats 5](#_Toc166453557)

[Detection de lunettes 5](#_Toc166453558)

[Conclusion 7](#_Toc166453559)

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

# Abstract

L’objectif de ce projet est de concevoir un système de détéction de certains attributs du visage. Ayant comme entrée une image ou une video d’un visage, le système doit être capable detecter les attributs suivant :

* La présence ou non de lunettes
* L’origine de la personne
* Le sexe de la personne

Inspiré de systèmes de surveillances existant, le programme pourrait servir par exemple a classifier une liste de portraits en fonction de ces attributs.

# Traitement

## Pipeline DeepFace

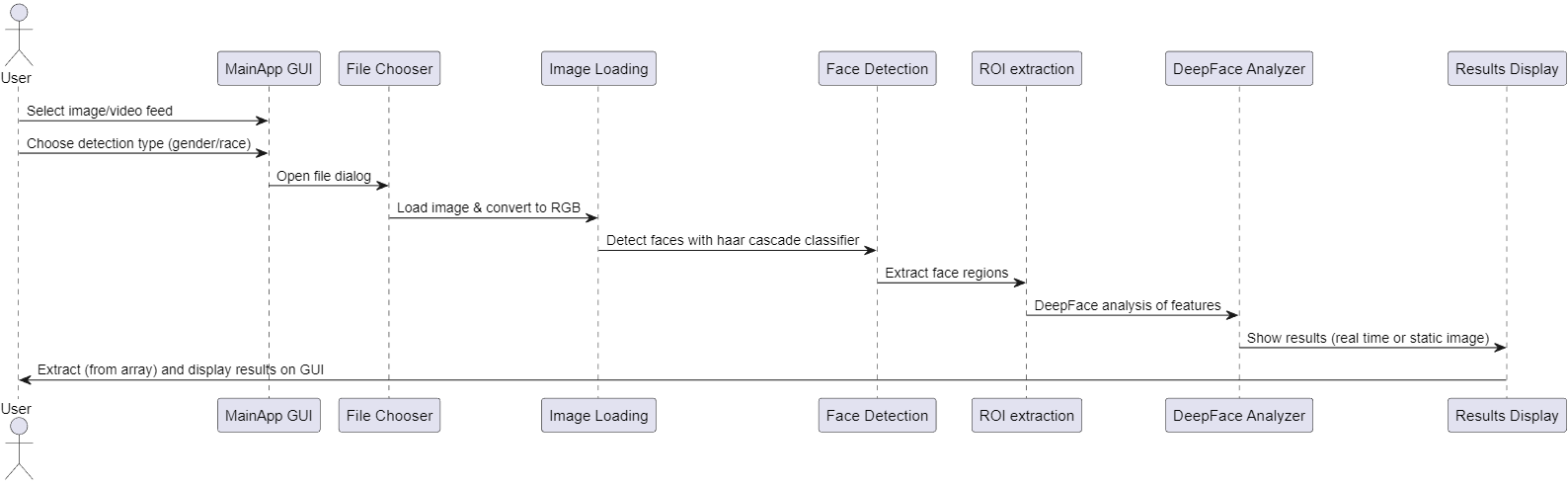


Figure 1 : schéma de la pipeline deepFace simplifiée

Les attributs de sexe et d’origine sont determinée à l’aide du modèle DeepFace de Facebook. Tout d’abord, on commence par detecter la présence de visages à l’aide d’un classificateur Haar specialement prevu 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 a ce moment la 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, dependant 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 [[1]](#footnote-2)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 éthnies dominantes (les trois pourcentages les plus elevées).

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 éthnies 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 detecté à l’aide d’un simple putText().

L’implementation 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 boucle et traite chaque frame de la VideoCapture (plutôt que de traiter qu’une image en entrée).

## Pipeline Edge Detection

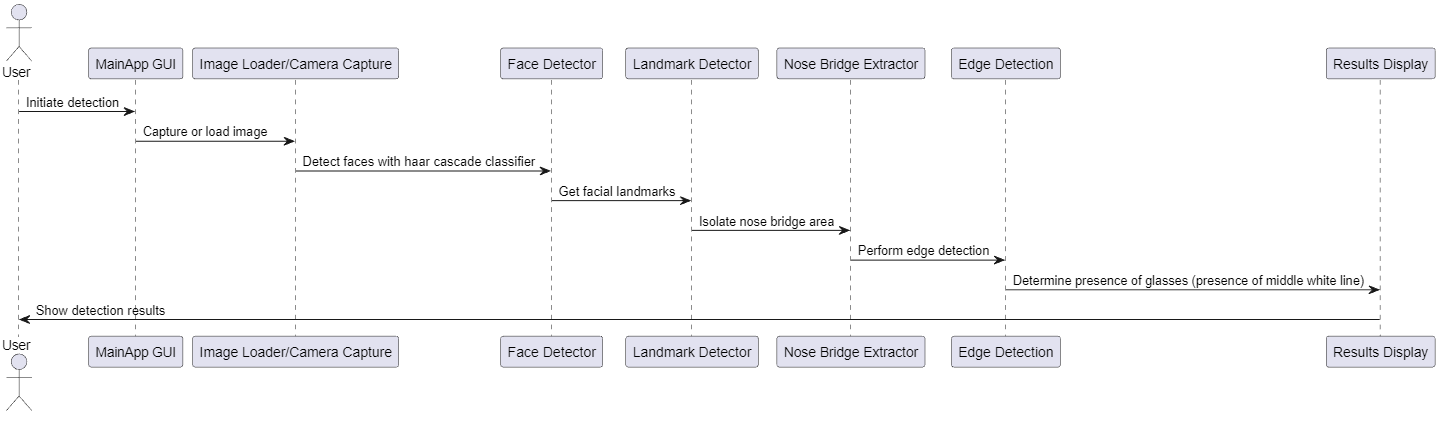


Figure 2 : schéma de la pipeline Edge Detection simplifée

Tout d’abord, tout comme pour DeepFace, on commence par detecter la présence de visages à l’aide d’un classificateur Haar specialement prevu pour (source inclus dans le code). Cependant, on utilise la fonction get\_frontal\_face\_detector() de la librairie de machine learning Dlib, plutôt que les fonctions cascade d’OpenCV.

Une fois le visage detectée avec le classificateur de points de repères, on isole le pont du nez du visage afin de le traiter separemment, 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’interesse 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.

Une image contenant texte, Visage humain, capture d’écran, Logiciel multimédia

Description générée automatiquement

Figure 3 : exemple des edges detecté autour de la region du nez

# Resultats

## Detection de lunettes

Une image contenant texte, verres, Visage humain, capture d’écran

Description générée automatiquement

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 indesirables), et portant un cadre dont le pont est asset épais comme sur l’illustration ci-dessus.

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

Une image contenant texte, capture d’écran, habits, Visage humain

Description générée automatiquement

Figure 5: Detection d'origine

Comme mentionnée auparavant, la detection d’origine affichent des pourcentages plûtot mitigé en general. DeepFace semble beaucoup mieux gerer les differences de luminosité et orientation du visage (grâce a leur technologie de re-construction faciale 3D), ce qui le rend plus fiable dans la plupart des scenario.

Cependant, contrairement a l’illustration, le resultat est plus souvent une estimation des trois éthnies les plus probables, plutôt qu’une réponse definitive.

Une image contenant texte, homme, Visage humain, capture d’écran

Description générée automatiquement

Figure 6 : Detection de genre

Quant a elle, la detection de genre produit des pourcentages plutôt definitif, malgré que le modèle semble parfois confondre le genre si la personne n’a pas d’attributs très distincs (p. ex : une barbe chez l’homme, machoire arrondie / maquillage chez les femmes, etc. )

# Conclusion

Les objectifs mis en place durant la phase de conception du projet ont été atteint avec differents degrée de succès. En effet, plusieurs ameliorations, notamment au niveau de la detection de lunettes peuvent encore être effectuée.

Il serait par exemple envisageable d’essayer de detecter la presence d’un cadre autour des yeux, afin de minimiser les erreurs du a une lumière non-homogène sur le visage.

Finalement, on pourrait aussi envisager d’implementer une analyse contextuel de ces informations, ainsi que d’autres attributs (DeepFace permet notamment d’analyser les emotions du visages), afin d’en tirer des conclusions dans certains scenarios.

1. <https://research.facebook.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/> [↑](#footnote-ref-2)