

RAPPORT DE PROJET

Module : Traitements d'images

Sujet : Reconnaissance faciale

AUTEURS :

Alioune Harouna Kanoute

Mouhamed Diop Wade

1. Introduction

La détection des visages est une technologie informatique qui identifie les visages humains dans des images fixes ou animées.

Dans ce projet , nous allons développer une application de détection de visage en utilisant OpenCV (Python) . La détection des visages est une technologie tendance presque utilisée dans tous les domaines de nos jours, dans la sécurité, dans la recherche, etc.

Nous utiliserons le module OpenCV, qui est une bibliothèque de vision par ordinateur avec le langage python pour détecter les visages humains.

Nous allons utiliser notre webcam PC pour obtenir le flux vidéo.

La détection des visages est un terme informatique utilisé lorsque le logiciel permet de déterminer l'existence, l'emplacement et la taille d'un visage humain sur une photo ou vidéo particulière. Le logiciel est suffisamment intelligent pour détecter les traits du visage, tout en ignorant d'autres objets.

2. Fonctionnement

Bien que le processus soit quelque peu complexe, les algorithmes de détection de visage commencent souvent par rechercher des yeux humains ou un visage frontal.

Les yeux constituent ce qu'on appelle une région de vallée et sont l'une des caractéristiques les plus faciles à détecter.

Une fois les yeux détectés, l'algorithme pourrait alors tenter de détecter les régions du visage, notamment les sourcils, la bouche, le nez, les narines et l'iris.

Une fois que l'algorithme présume qu'il a détecté une région faciale, il peut alors appliquer des tests supplémentaires pour valider s'il a effectivement détecté un visage.

Projet de traitements d'images

Détection du visage :

- Détecte le visage dans l'image.
- Il recherche le visage humain général comme un segment dans l'image entière. La sortie peut être un ou plusieurs.
- La sortie sera un rectangle ou des rectangles sur les faces de l'image.

Reconnaissance du visage :

- Reconnaître la face d'entrée de la base de données déjà formée avec le score de correspondance le plus élevé.
- Une seule face doit être donnée en entrée et la sortie sera un nom, un nom de classe ou une face inconnue.

3. Outils et technologies utilisés

- **OpenCV** : bibliothèque graphique libre, initialement développée par **Intel**, spécialisée dans le traitement d'images en temps réel.
- **DLib** : librairie contenant des algorithmes d'apprentissage automatique et des outils pour créer des logiciels complexes et résoudre des problèmes du monde réel.
- **Pillow** : bibliothèque qui permet la prise en charge de l'ouverture, de la manipulation et de l'enregistrement de nombreux formats de fichiers images différents.
- **Numpy** : package de traitement à usage générale. Il fournit des tableaux multidimensionnels et des outils pour travailler avec.

4. Mise en œuvre(codage)

Pour la mise en œuvre, on a utilisé PyCharm, qui est un environnement de développement intégré utilisé pour programmer en Python, développé par JetBrains.

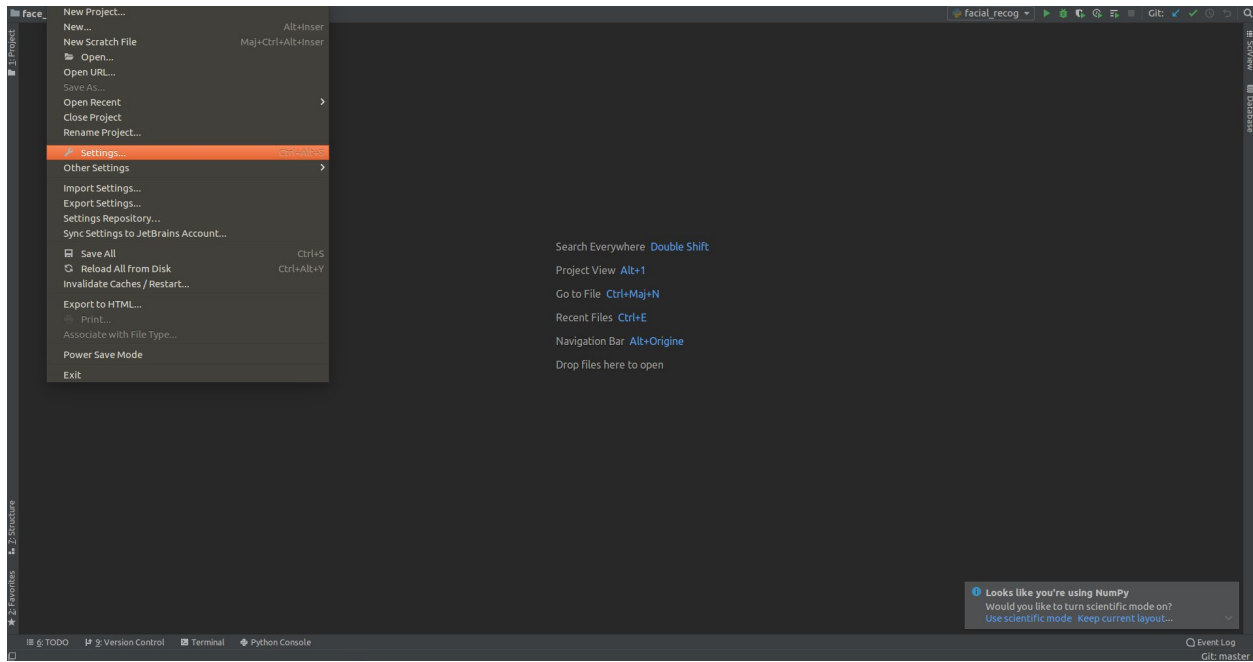
Installation des librairies : on peut procéder de deux manières

- Installer les librairies en ligne de commande
- Installer les livres dans le projet (à l'aide de PyCharm)

Projet de traitements d'images

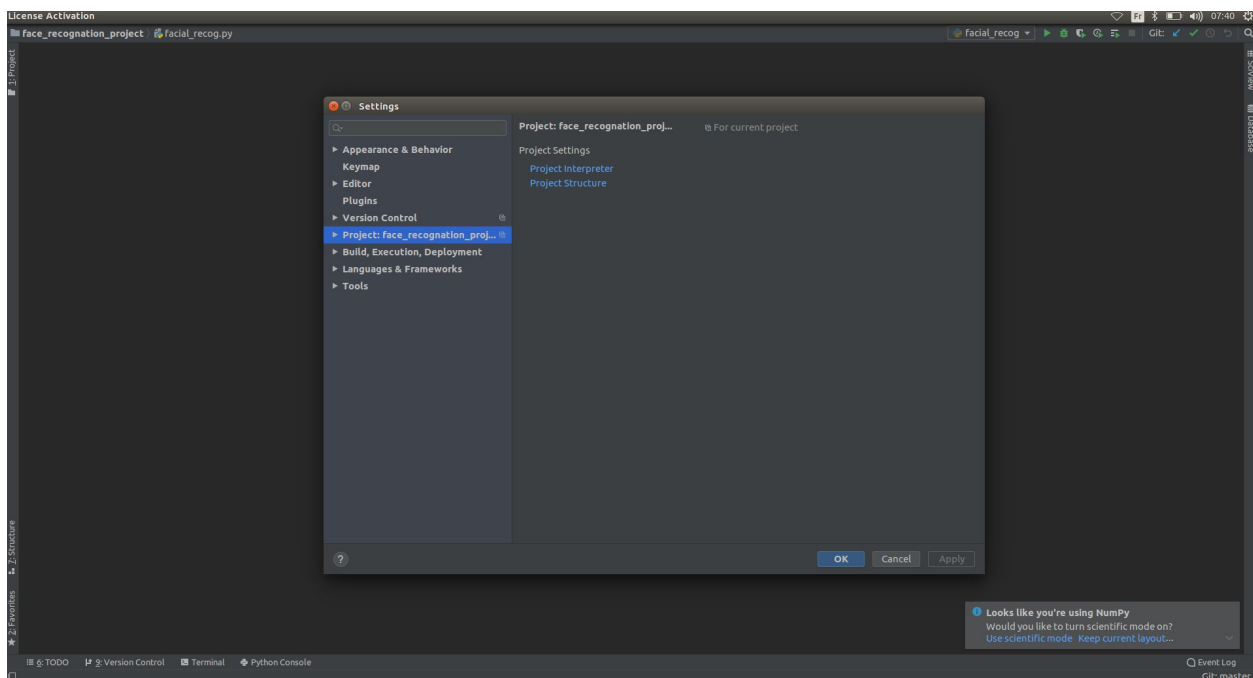
Installation dans le projet avec PyCharm

– *File* —> *Settings*



Étape 1

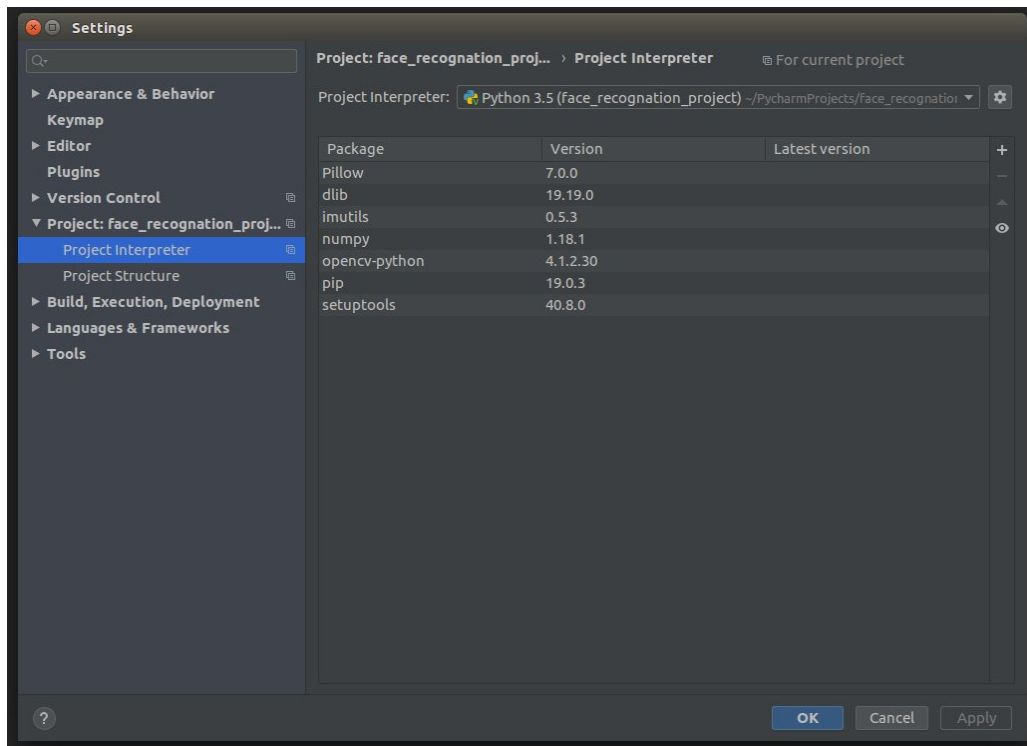
– *Project* —> *Project Interpreter*



Étape 2

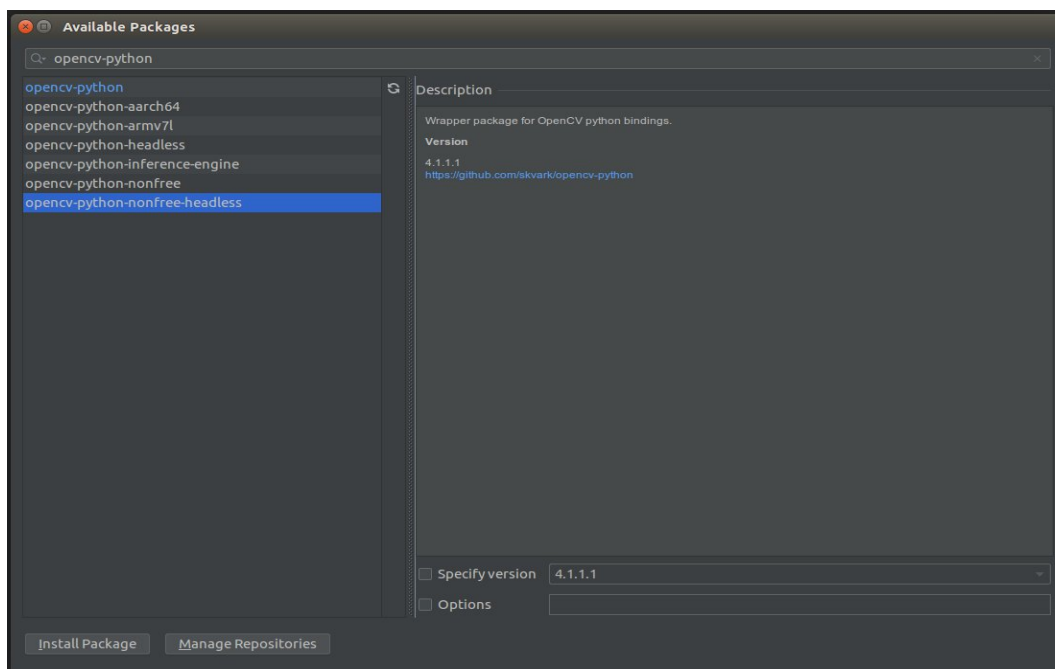
Projet de traitements d'images

- Cliquez sur l'onglet + en haut à droite pour ajouter une bibliothèque



Étape 3

- Tapez le nom de la bibliothèque à installer sur la barre de recherche puis cliquez sur le bouton **Install packages**



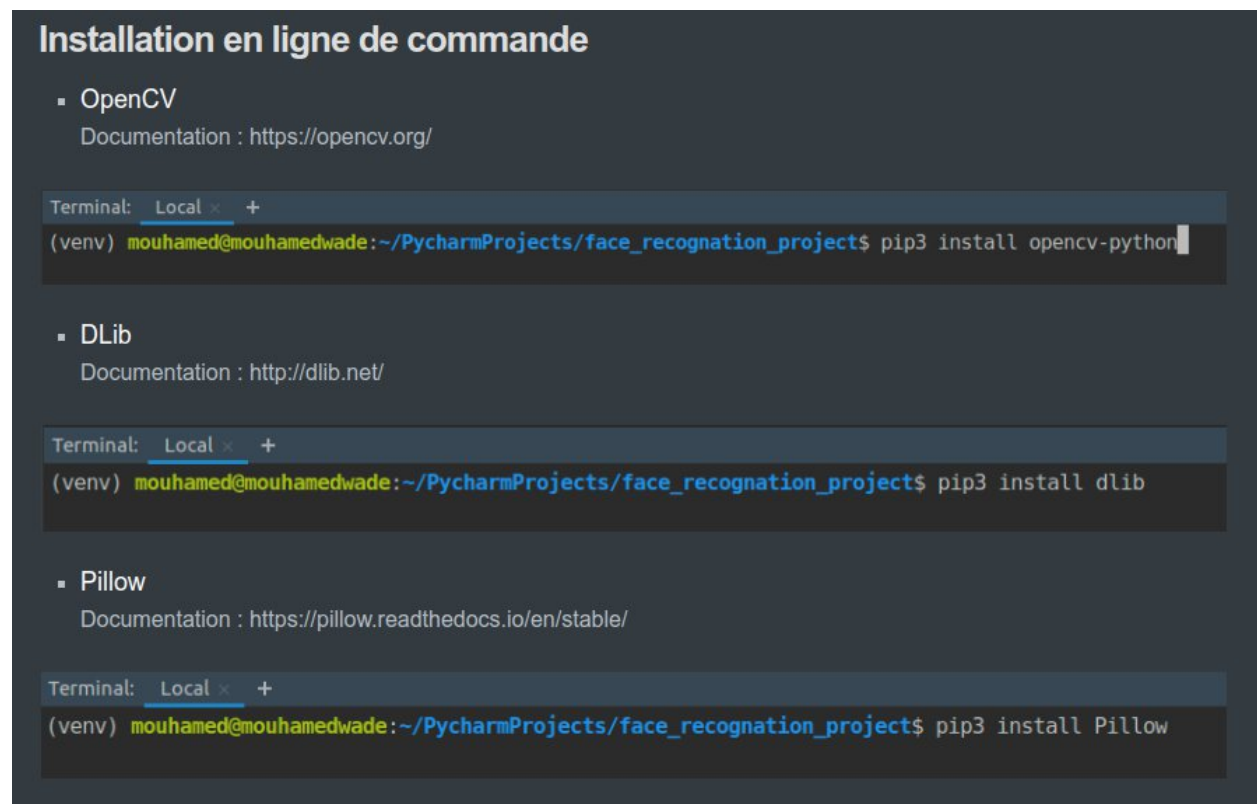
Étape 4

Projet de traitements d'images

On procède de la même façon pour installer les autres librairies.

NB : La librairie Numpy vient avec OpenCV.

Installation en ligne de commande : les commande pour installer chaque bibliothèques en ligne de commande.



The screenshot shows a terminal window with a dark background. It displays the installation of three libraries: OpenCV, DLib, and Pillow. Each section includes a heading, a documentation link, and a terminal command.

- OpenCV**
Documentation : <https://opencv.org/>
Terminal: Local x +
(venv) mouhamed@mouhamedwade:~/PycharmProjects/face_recognition_project\$ pip3 install opencv-python
- DLib**
Documentation : <http://dlib.net/>
Terminal: Local x +
(venv) mouhamed@mouhamedwade:~/PycharmProjects/face_recognition_project\$ pip3 install dlib
- Pillow**
Documentation : <https://pillow.readthedocs.io/en/stable/>
Terminal: Local x +
(venv) mouhamed@mouhamedwade:~/PycharmProjects/face_recognition_project\$ pip3 install Pillow

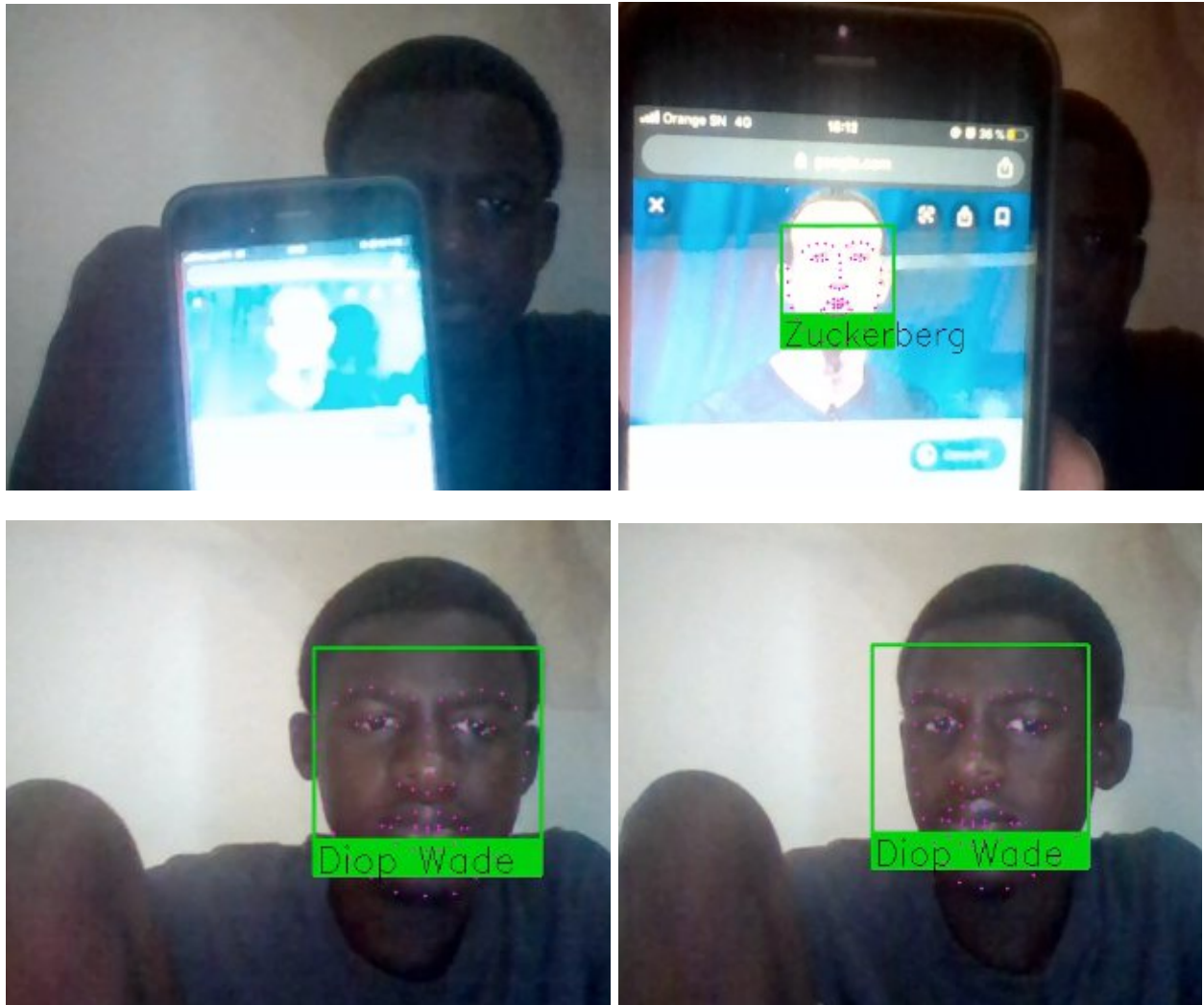
Codage

Nous avons une base d'images de personnes connues et leurs noms.

La première étape est de détecter un visage sur un flux vidéo. S'il y'a détection, le programme recueille le visage détecté, puis place des points sur le visage (pour avoir des coordonnées) et ensuite le transforme en un vecteur de taille 128. La deuxième étape consiste à faire la différence entre le vecteur (128) du visage détecté avec ceux des visages connus. Si cette différence est inférieure à la valeur de tolérance (0.6 pour notre application) on considère que c'est la même personne et par conséquent on affiche le nom sinon on affiche inconnu.

Réalisation

Quelques captures



Démarrer l'application avec la commande

```
> python facial_recog.py --i visages_connus
```

visages_connus état le dossier des visages de notre base.