**Requirements:**

python 3 (3.5 ++ is recommended)

opencv3

numpy

tensorflow

**Structure:**

|-- architecture (architecture for FaceNet)

|-- inception\_resnet\_v1.py

|-- face\_data

|-- facerec\_128D.txt

|-- facerec\_CASIA.txt

|-- facerec\_VGG\_pic.txt

|-- facerec\_VGG.txt

|-- main.py (main program)

|-- models (pre-trained models used in the project)

|--20170512-110547

|--20170512-110547.pb

|--20180402-114759

|--20180402-114759.pb

|--20180408-102900

|--20180408-102900.pb

|-- det1.npy

|-- det2.npy

|-- det3.npy

|-- udf

|-- align\_custom.py

|-- camera

|-- camera.py

|-- face\_model.py

|-- find\_people.py

|-- mtcnn\_detect.py

|-- pic

|-- pic.py

|-- tf\_graph.py

**Instructions:**  
*Use camera mode:* **To do face recognition:**  
 Use pre-trained FaceNet model on VGG dataset to recognize faces:  
 python main.py --mode="VGGcamera"  
 Use pre-trained FaceNet model on CASIA dataset to recognize faces:  
 python main.py –mode="CASIAcamera"  
 Use pre-trained FaceNet model on old 128D dataset to recognize faces:  
 python main.py --mode="FaceNetcamera"  
  
 **To input new user:** Use pre-trained FaceNet model on VGG dataset to input faces:  
 python main.py --mode="VGGinput"  
 Use pre-trained FaceNet model on CASIA dataset to input faces:  
 python main.py --modeCASIAinput"  
 Use pre-trained FaceNet model on old 128D dataset to input faces:  
 python main.py --mode="FaceNetinput"  
*Use picture mode(only implemented on VGG dataset):*  
 **To do face recognition:**  
 python main.py --mode="VGGpicture"  
  
 **To input one new user:** python main.py --mode="VGGpic\_input\_one"  
 To input all new users together:  
 python main.py --mode="VGGpic\_input\_all"