IMPLEMENTATION OF IMAGE RECOGNITION USING PYTHON IN ARTIFICIAL INTELLIGENCE.

GROUP MEMBERS

KEVIN CHEGE – SCCI/01147/2018

SHADRACK MUTUMA – SCCI/01151/2018

MARTHA AMONDI – SCCI/01143/2018

MARTIN NGIGI WAINAINA - SCCI/01148/2018

MARJORIE MULOMA – SCCI/01142/2018

JAMES OMONDI - SCCI/01167/2018

BETH MAINA – SCCI/01135/2018

AUSTIN OKOLA ODHIAMBO - SCCI/01166/2018

LEAH KERUBO OBWOCHA - SCCI/01136/2018

AMANDA SUZZANNE ADHIAMBO - SCCI/01140/2018

SUMMARY

A group of 10 students were assigned a task of creating a simple image recognition application where We are given a bunch of faces – possibly of your other group members.Make the images of groip members known and if any other image is not that belonging to the group, it comes up as a “visitor”. The task is simple was to identify if this “visitor” is among those present in the group.

INTRODUCTION

The purpose of this report is to explain how we went about with the solution of coming up with the model required to identify the images correctly.

We used python as the preferred programming language due to its availability of vast libraries and human understandable approach. The libraries and/ or the requirement used in the project were:

Python 2.7 or higher, Numpy, dlib, open-cv, face-recognition and cmake.

We used Pycharm as our preferred IDE due to its easier to use interface and ability to configure the libraries without complications. We also used webcam to help identify the group members in real time.

This was as an assignment given by Mr. Eddy Addero at the end of the Artificial Intelligence Course to assess our ability to comprehend whatever he had taken us through during the course.

BODY

Inorder to come up with our images for the system to identify, a couple of members gave out their photos which we used as an image knowledge base for training. Open-cv and Pillow libraries then helped train the images by finding the face encodings and storing the encodings for reference. We also had a test image consisting of members and non members to be used in distinguishing the members.

Once the libraries have gotten the encodings of each face and stored them in an array list, we gave unique names to each of the group members for easier identification, and simply “visitors” for the non members.

The system was given access to the computer’s face cam in order to identify the faces in real time and compare with the already captured in the system.

The facial detection was done using the HOG method i.e., Histogram of Oriented Gradients

The system was then tasked with iterating through the faces in the webcam and in the test image to compare and uniquely identify the group members. Upon finishing, we set it to draw rectangular shapes/ boxes around each face and write the unique name under the box.

Earlier we had a problem where complexity of the skin and the shape of the faces that were almost similar were resulted in the system showing the same output even if the faces were different. This error was reduced by setting a tolerance of 0.5 to each face.

Requirements for running this program:

You will need to install the modules mentioned above using

pip for windows and pip3 for linux based systems

For Linux based Systems

pip3 install numpy open-cv-python dlib face-recognition cmake

For windows based System

pip install numpy opencv-python dlib face-recognition cmake

NB: Your webcam has to be working

CONCLUSION

We managed to implement a system that could uniquely identify an image of a group member from non members using python. The system uniquely identify any face that the webcam picks up as either a group member, identified by name and a visitor.

REFERENCES

<https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/>

<https://youtu.be/AACPaoDsd50>

<https://towardsdatascience.com/building-a-face-recognizer-in-python-7fd6630c6340>

<https://realpython.com/face-recognition-with-python/>