

# **Advice individual project**

**Shihaab Rouine  
Semester six**

## Introduction

This is an advice document that will help continue the project. It contains a schematic explanation of the application, the concept, which libraries to use and what the quality criteria is.

## Concept

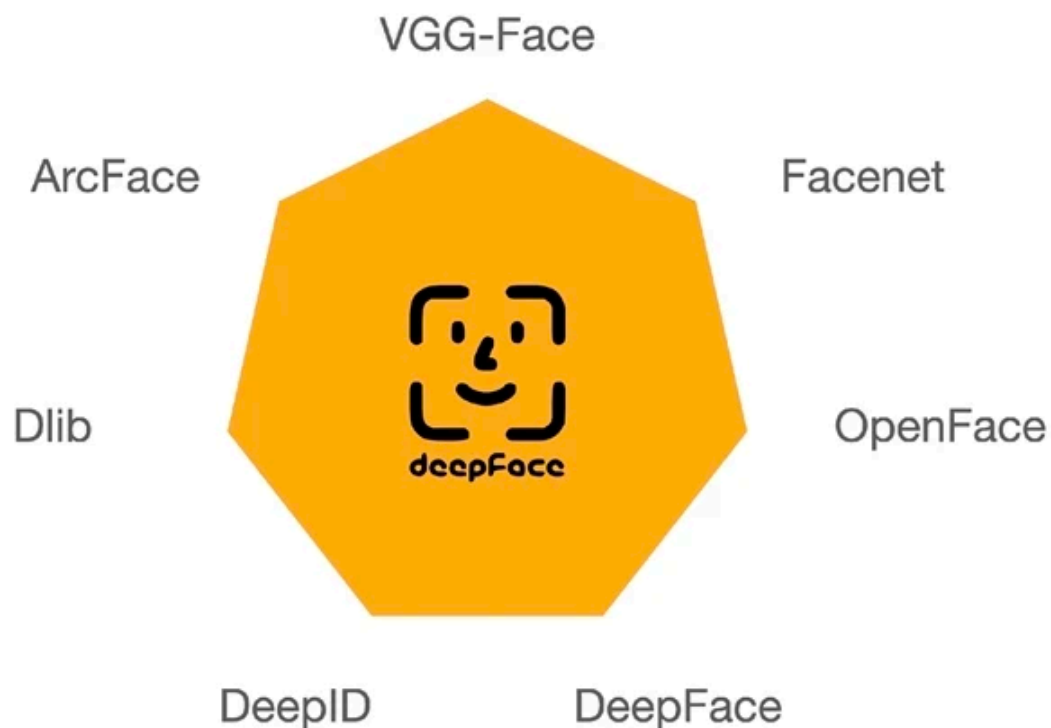
This is also explained in the project plan with extra clarification.

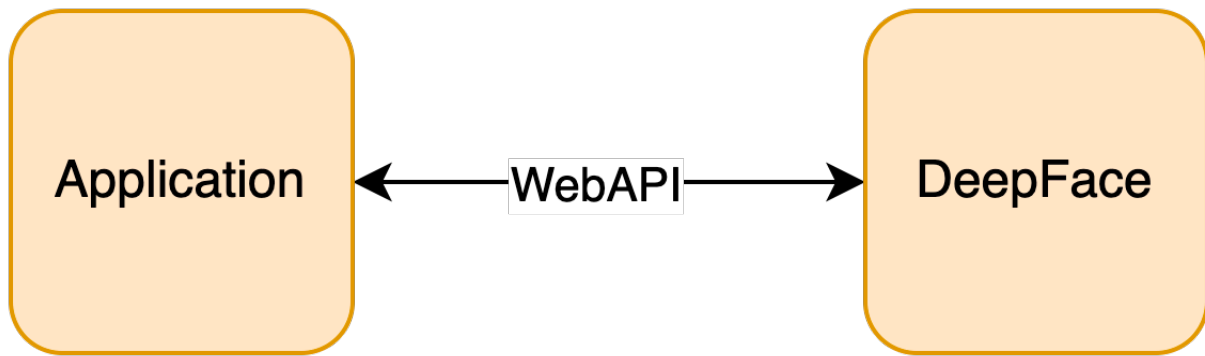
A panel that scans your face for emotions prior to entering a room, the emotions needs to meet a certain level of happiness. If the level is not met, the person attempting to enter needs to show a laugh or a smile to the camera to set foot in the room.

The idea is that forcing a person to smile in a bad mood will, hopefully, spark some sort of up cheering moment. Some meetings seem too serious for what they meant to be, using the panel at the entrance may lighten the mood. This may also help people express their opinions earlier during meetings, as the lightened mood makes it less daunting.

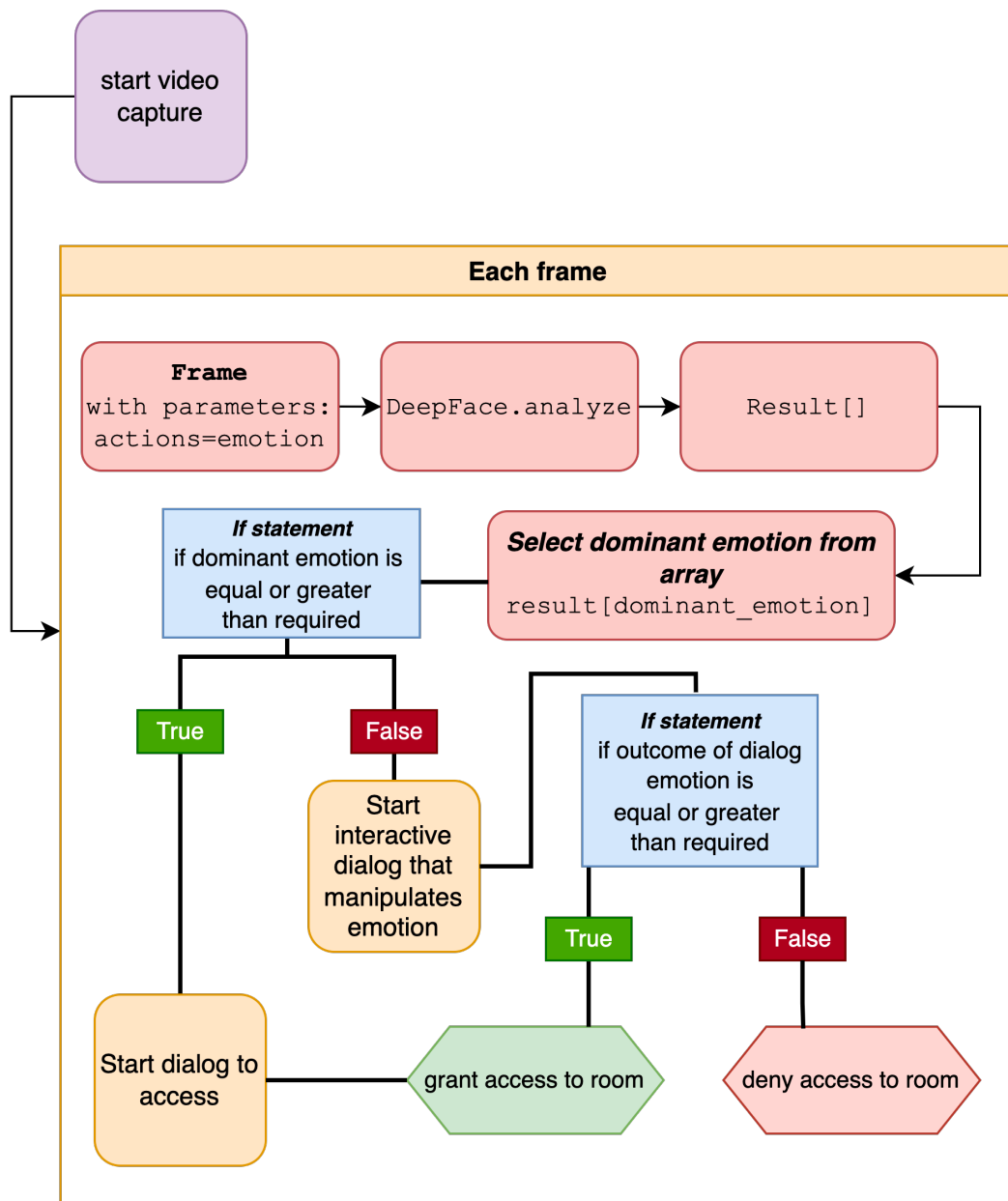
## Schematics

The Application makes use of the powerful framework *DeepFace*. *DeepFace* consists of multiple face recognition models, like VGG-Face, Facenet, OpenFace, ArcFace and Dlib.





DeepFace also has an API; you may clone the API files from DeepFace and provide them as an input to a Python program. This will activate a rest service. In this way, you can call DeepFace from an external system.



Above is the schematic that visualises how the system should work. The “Each frame” container loops every frame of the live video feed to analyse each frame for emotion. Then triggers one of the two dialogs. The content of these dialogs should be tested and researched.

During the “interactive dialog that manipulates emotion“ the user will need to

## **Libraries**

For the emotion recognition application there are only two libraries really needed for the program to run properly. OpenCV and DeepFace. OpenCV will be automatically installed with DeepFace.

OpenCV also known as cv2, is a library of programming functions primarily geared towards real-time computer vision. In this application cv2 is used to quickly analyse the live video feed.

DeepFace has many more dependencies, but luckily these libraries are included when you install the DeepFace framework, so you don't have to install them separately.

## **Quality criteria**

There is some criteria for the concept to work properly.

### **Scalability**

The concept can be broadened and scaled to any size, it is the dialogical interaction between user and system that is key. When the system is applied to small scale it will work just the same when applied to a larger scale. Its function and result will not differ.

### **Safety**

As far as this concept is concerned, there is no major safety threat. Although the use of information of the users may be a privacy breach.

### **Software hardware compatibility**

The application need to be running from a system that is compatible with OpenCV and DeepFace. The architecture of the processor needs to be *x64* or *x86* to be able to install the libraries. Via the *cmd* or *terminal* this can be easily checked with the ‘*arch*’ command.

Each frame needs to be analysed by the program, this takes a bunch of processing power. A powerful computer is a must. This can be a laptop or a powerful tablet, but also a Raspberry Pi. A Raspberry Pi is a small single board computer, useful due to its size and it easy to develop with.