Face Detection and Recognition

Students: Daniel SAVA

Antonia GAVRUTA

Instructor: Claudiu GROZA

Date: May 2019

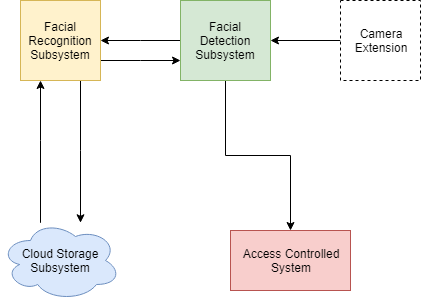
Politehnica University of Timisoara, AC CTI-EN Year 3

1. **Repository**

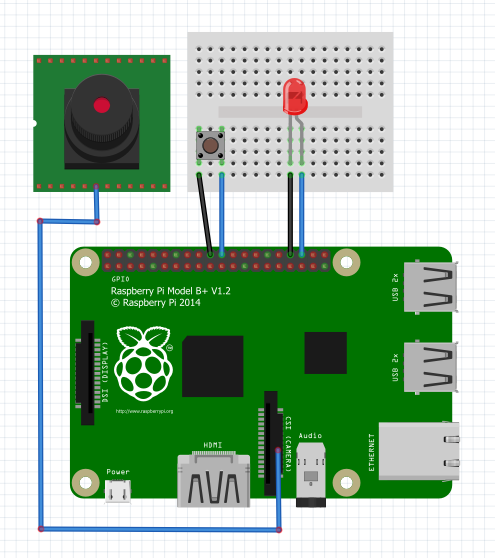
The schematics, diagrams and code of the project are contained under the following git repository:

[**https://github.com/SavaDaniel/msproject**](https://github.com/SavaDaniel/msproject?fbclid=IwAR3stUmIwz1n7D_G1Md92FZmccRjeGH5f2UhtK0-RW_h9qTwPmdB1gdDA50)

1. **User requirements**
2. The system must detect the face of any user that comes close to the camera.
3. If the detected face is recognized, the access to the system is granted, otherwise it is not.
4. The system may provide access to all the pictures saved by the camera.
5. The system must provide the necessary information with a maximum latency of 10 seconds.
6. The system must run in an environment that will provide 24/7 access.
7. The system should be open for extension (eg. Adding an LCD display).
8. The system may provide a module for data interpretation.
9. **System overview**

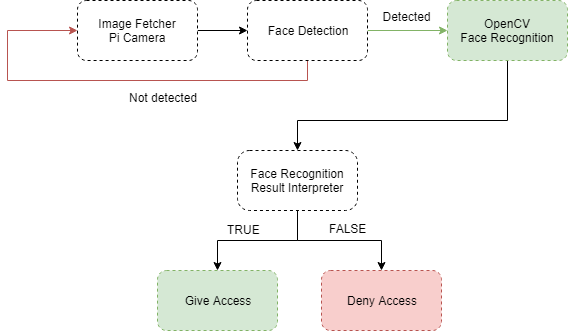


1. **Hardware overview**



Hardware components used:

1. Raspberry Pi 3 Model B+ - With a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE and faster Ethernet.
2. Raspberry Pi Camera V2 - This 8mp camera module is capable of 1080p video and still images. It connects to the Raspberry Pi Board via a Camera Serial Interface (CSI) ribbon cable.
3. LED - Which will light in case a user will be granted access.
4. Button - Images will be fetched when the button is pushed.
5. **Software overview**



OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license.

OpenCV supports the deep learning frameworks TensorFlow, Torch/PyTorch and Caffe.

Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on artificial neural networks. Learning can be supervised, semi-supervised or unsupervised.

Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts.

In our case we have used the ” NumPy” library, which is the fundamental package for scientific computing with Python. It contains among other things:

* a powerful N-dimensional array object
* sophisticated (broadcasting) functions
* tools for integrating C/C++ and Fortran code
* useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy is licensed under the [BSD license](https://www.numpy.org/license.html#license), enabling reuse with few restrictions.

1. **Results and further work**

The current version of the project supports the following functionalities:

• Reliable recognition of a person or multiple persons in front of the camera

• Storing the pushed photos

The following list of extensions and improvements was identiﬁed to be supported in the future:

• More advanced detection and recognition features

• Reliable recognition of multiple objects

**7. References**

1. Draw IO https://www.draw.io

2. <https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python/?fbclid=IwAR2VlarBajz3y8oOg3fIrwE3V0ef-GyHTydvqnP66KA4axXMEYmCM2A-AZg>

3. <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc.html?fbclid=IwAR0kKCjco4Mbd1Inta70anDKsL_2GlcqVSA5Yyq4_53mp5bydRSD0zu__ks>

4. <https://stackoverflow.com/?fbclid=IwAR3lvlAGnpa30UwsxNG8oTVZ9UkNtJ4M6oYKvrb-9UtsEqboiXYyvxoz3C0>

5. <https://www.youtube.com/?fbclid=IwAR1B0Etr-Kw_6TIYN1Qz_gryJaZ_m3G5v3T-bYr8ncCUAaXvdYRq4U0-TPI>

6. <https://www.pyimagesearch.com/2016/01/04/unifying-picamera-and-cv2-videocapture-into-a-single-class-with-opencv/?fbclid=IwAR38jXOLvJJDimCfjm_xjpZ7IYvEVgJBbSvFZJpqK9PkHMJFTVGrIRwy6WM>