**FaceRecognition**

Webcam face recognition using tensorflow and opencv. The application tries to find faces in the webcam image and match them against images in an id folder using deep neural networks.

**Dependencies**

* OpenCv
* Tensorflow
* Scikit-learn

**Inspiration**

Models, training code and inspriation can be found in the facenet repository. Multi-task Cascaded Convolutional Networks are used for facial and landmark detection while an Inception Resnet is used for ID classification.

Visualisation Hotkeys

* l - toggle facial landmarks
* b - toggle bounding box
* i - toggle id

## Steps of tensorflow

### Creating accurate Machine Learning Models which are capable of identifying and localizing multiple objects in a single image remained a core challenge in computer vision. But, with recent advancements in **Deep Learning, Object Detection**applications are easier to develop than ever before. TensorFlow’s Object Detection API is an open source framework built on top of TensorFlow that makes it easy to construct, train **and** deploy object detection models.

## ****What is TensorFlow?****

Tensorflow is Google’s Open Source Machine Learning Framework for dataflow programming across a range of tasks. Nodes in the graph represent mathematical operations, while the graph edges represent the multi-dimensional data arrays (**tensors**) communicated between them.

Tensors are just multidimensional arrays, an extension of 2-dimensional tables to data with a higher dimension. There are many features of Tensorflow which makes it appropriate for Deep Learning. So, without wasting any time, let’s see how we can implement Object Detection using Tensorflow.

### **Getting Prerequisites**

* Before working on the it, let’s have a look at the prerequisites. We will be needing:
  + Python
  + TensorFlow
  + Tensorboard
  + Protobuf
* Next, we have Protobuf: **Protocol Buffers**(Protobuf)  are Google’s language-neutral, platform-neutral, extensible mechanism for serializing structured data, – think of it like XML, but smaller, faster, and simpler.
* Rename the “models-masters” to just “**models**“.
* Now for simplicity, we are going to keep “models” and “protobuf” under one folder “**Tensorflow**“.
* Next, we need to go inside the Tensorflow folder and then inside research folder and run protobuf from there using this command.
* Next, we don’t need to load the images from the directory and convert it to numPy array as OpenCV will take care of that for us.
* We will not use matplotlib for final image show instead, we will use OpenCV for that as well.