

10/15/2018

Younsi Abderrazek

Facial Expression Detection

Krifi Fatma

Sedghiani Thabet

# Deeplearning

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It’s achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.



# Realisation

Facial Expression or Facial Emotion Detector can be used to know whether a person is sad, happy, angry and so on only through his/her face. This Repository can be used to carry out such a task. It uses your WebCamera and then identifies your expression in Real Time.

This is a three step process. In the first, we load the XML file for detecting the presence of faces and then we retrain our network with our image on five different categories. After that, we import the label\_image.py program and set up everything in Realtime.

We need to install tensorflow and opencv for python by tapping this commands:

pip install tensorflow

pip install opencv-python

## STEP 1 - Implementation of OpenCV HAAR CASCADES

I'm using the "Frontal Face Alt" Classifier for detecting the presence of Face in the WebCam. This file is included with this repository.

Next, we have the task to load this file, which can be found in the [**label.py**](https://github.com/MauryaRitesh/Facial-Expression-Detection/blob/master/label.py) program. E.g.:

# We load the xml file

classifier = cv2.CascadeClassifier('haarcascade\_frontalface\_alt.xml')

Now everything can be set with the Label.py Program. So, let's move to the next Step.

## STEP 2 - ReTraining the Network - Tensorflow Image Classifier

We are going to create an Image classifier that identifies whether a person is sad, happy and so on and then show this text on the OpenCV Window. This step will consist of several sub steps:

* We need to first create a directory named images. In this directory, create five or six sub directories with names like Happy, Sad, Angry, Calm and Neutral. You can add more than this.
* Now fill these directories with respective images by downloading them from the Internet. E.g., In "Happy" directory, fill only those iages of person who are happy.
* Now run the "face-crop.py" program.
* Once you have only cleaned images, you are ready to retrain the network. For this purpose, I'm using Mobilenet Model which is quite fast and accurate. To run the training, hit the got to the parent folder and open CMD/Terminal here and hit the following:
* python retrain.py --output\_graph=retrained\_graph.pb --output\_labels=retrained\_labels.txt --architecture=MobileNet\_1.0\_224 --image\_dir=images

That's it for this Step.

## STEP 3 - Importing the ReTrained Model and Setting Everything Up

Finally, I've put everything under the "label\_image.py" file from where you can get everything. Now run the "label.py" program by typing the following in CMD/Terminal:

python label.py

It'll open a new window of OpenCV and then identifies your Facial Expression.





