face_rec package documentation

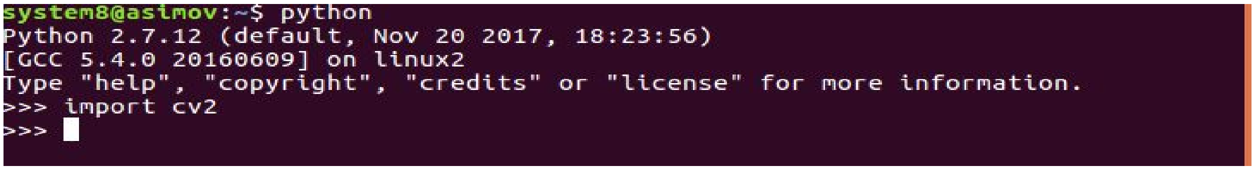
What the package does ?

The package monitors webcam stream for faces. If a face is detected for more than 20 frames then the closest face is determined (in case of multiple faces in frame) and runs face recognition on that face. If the face is of a person we have trained for then it outputs the label of the person, if the face is not trained for then it outputs **Unknown**. Also if no face is present in frame it outputs **None**.

If the face moves out of field of view of camera, then corresponding rotation command is sent to the neck motor.

Pre-requisites:

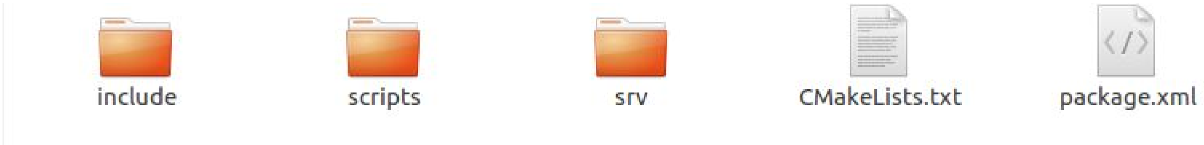
Make sure opencv is installed with python support. You can check by firing python in terminal and importing cv2



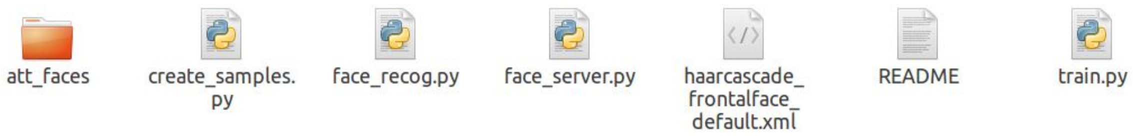
```
system8@asimov:~$ python
Python 2.7.12 (default, Nov 20 2017, 18:23:56)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> 
```

Package contents:

Copy the face_rec package into you catkin workspace and ensure the following directories are present inside face_rec package.



The code is present inside the scripts folder. The srv folder consists of service file definition. The directories inside scripts folder looks like below :



face_recog.py :
Consists of the python code for training and face_recognition. Also publishes command to the servo.

create_samples.py:
This script creates training samples from images. It detects the faces in images, converts them to appropriate image dimensions and stores the training samples inside att_faces directory.

Usage : python create_samples.py /path/to/images

face_server.py:
Consists of code the defines the server which subscribes to the topic /face_rec, in which the label of the person identified by face_recog.py. It publishes None if no face is detected, Unknown if face is of unrecognized person and if the face is of a person whose is present inside att_faces directory then that label is published.

train.py:
This code creates training images for face_recog.py to train. To call the code use : **python train.py name_of_person**. It collects 50 sample images from the webcam and close automatically. When the code is running, move the head around but ensure that eyes can be seen throughout .A directory will be created in att_faces with name_of_person
Which contains the training samples. There should be fifty images. Open the directory and check that all 50 images contain the face of the person.

<u>Performance Benchmarks:</u>

The code initially trained with 20 training samples. In order to increase the accuracy of prediction we increased the number of training samples to 50 images. This has improved the accuracy of face recognition however face detection still suffers from false positives when trying to identify/detect faces. Equalizing histogram did reduce the number of false positive yet it is not 100 percent perfect.

Overall the efficiency of code is limited by noise in the background and lighting conditions. In order to produce best results it is recommended to create the training images under same conditions in which we aim to reproduce the results. Also as mentioned above, after running train.py check the quality of training samples present inside the att_faces/name_of_person directory.