Project Title:- Face detection Algorithm on Raspberry Pi using Haar Cascades

Submitted by:-
Vasa Vamsi Krishna,
PG Scholar,
IIT Gandhinagar.

- Running and Killing the code :
  - Initiate test_1.py code on the terminal of the raspbian OS(Make sure all the required codes are present in the same folder). This code will take care of capturing the images every 0.5 sec(approx.) to avoid losing the data while processing it.
  - Now start another terminal and run test_2.py on this terminal.
  - To stop the capturing, manually terminate the code on terminal running test_1.py
  - The test_2.py is coded in a way that it would wait even after processing the images available. Still if need arises to terminate, it can also be terminated manually.
- End to End Pipeline :
  - test_1.py code is responsible for capturing the images of the scene every 0.5(approx.) seconds.
  - This captured images are saved in the Home directory of the Pi in the folder named "Captured_images".
  - The time stamp to this code can be added manually.
  - test_2.py code will intake the images in the sequence they are captured and sent to the opendrive cloud account according to the credentials provided in rpiconfig.py script. Before that it will wait for the capturing code to save atleast 5 images in this directory.
  - For the first image it will run "face_detection" module from the haarcascade_frontalface_default asset taken from the github repository(link for the same is mentioned in the citations)
  - For the next coming images the mean square error between the present and previous image is calculated.
  - If this error calculated is less than certain threshold, we may consider both the images as similar and the image can be excluded there itself and continue to the next image.

- ❖ If it surpasses the error criteria, the image will be processed for the face detection, if no. of faces detected are more than zero, the image will be sent to “send_images” folder in the Home folder.
- ❖ Then the same image is sent to the output folder in the cloud account.
- ❖ After this above mentioned process the previous image is deleted from the local directory (for the sake of memory constraints on Pi).
- ❖ After completing the processing the images, it will wait for the new images. It will wait till five images are saved into the “Captured_images”.

Haar Cascades classifiers are machine learning based technique developed for the various objects detection in the images such as frontal face, eyes etc. these classifiers are trained on various positive and negative images. The reading material for explaining the same is cited.

- ➢ Folder Structure :
    - ❖ Input directory(local) → Home/Captured_images
    - ❖ Output directory(local) → Home/send_images
- ➢ Cloud credentials can be changed as per required.
- ➢ Required Packages:
    - ❖ All the mentioned packages and libraries can be downloaded as pip installs.
    - ❖ Required packages are as follows:
        - Opencv (as per the version of python in use)
        - Numpy
        - Datetime
        - Webdav
        - Picamera

- Citations:
  - https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
  - Link for github repository - https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml