Automotive Sensor for Object Recognition using RedPitaya and Raspberry Pi

User Guide_SL2-6

Manual settings:

1. Power on Red Pitaya first.
2. Power on Raspberry Pi and connect it to Red pitaya Wifi, password is ***‘redpitaya’***.
3. Now open command line on Raspberry Pi and run command ***‘ssh root@192.168.128.1’*** it asks for a password which is ***‘root’*** after you enter the password you start getting readings from Red pitaya, keep this running pointing to any one of the object i.e. wall, human or car.
4. Placement of Red Pitaya is important, it should be 1 meter away from any object and on around same height.
5. Now copy your SL2-6 folder on desktop of Raspberry Pi, which has python codes and data.
6. Now open another command line, go to the desired folder SL2-6 and run code with command ***‘python3 SignalRepresention.py’*** it shows you signal of data that you are getting from Red pitaya.
7. Like wise in order to save that signal run command ***‘python3 SignalSave.py’*** it starts saving data in .csv file in folder CollectedData. We have already saved multiple samples for wall, human and car in that folder.
8. Now you don’t need Red Pitaya connection for extraction feature since it is using files from folder CollectedData and saving features in FeaturedData, run command ***‘python3 SignalFeatures.py’***, it might ask to install tsfresh library for python so run command ***‘sudo pip3 install tsfresh’*** and run again ***‘python3 SignalFeatures.py’***.
9. Now after doing feature extraction run command ***‘python3 BayesClassifier.py’*** it trains 80% of data and test 20% to share accuracy with you and save this file in BayesClassifierModel.joblib for classification. Accuracy should be around 1, but it will depend the quantity and quality of data.
10. Do same for another classifier by running command ***‘python3 SvmClassifier.py’*** which again repeat above steps but saving file in SvmClassifierModel.joblib.
11. Depending on accuracy for each classifier you must select one with better accuracy, so in our case BayesClassifier.py is working with accuracy 1. So, in Classification.py file we loaded this BayesClassifierModel.joblib, so that It can predict data as per this classifier.
12. Now follow step 1 & 3 in case you switched your Red Pitaya off and run command ***‘python3 Classification.py’*** and point Red pitaya sensor to any of these three objects i.e. Wall, Human & Car. Respective LED should blink as defined in Classification.py code i.e. LED1 for wall, LED2 for human and LED3 for car.

Plug and Play settings:

1. To make whole project plug and play, we have done adjustment in cron job, which is a scheduler in Raspbian and very easy to use. Open command line and type ‘crontab -e’ it will show you 3 option for different editor, select 1 and write below commands and save it. It will execute these commands after every 10 minutes and 5 minutes respectively.

   ```
   */10 * * * * sudo sshpass -p “root” ssh root@192.168.128.1 &
   */5 * * * * sudo python3 /home/pi/Desktop/SL2-6/Classification.py &
   ```

2. Alternatively, you can also make some changes in the script *etc/rc.local* of Raspberry pi, this file is traditionally executed after all the normal system services are started. Open command line and with ‘***sudo chmod +x /etc/rc.local’*** make rc.local script executable and through ***‘sudo vi /etc/rc.local’*** edit this file with given text.

   ```
   sshpass -p “root” ssh root@192.168.128.1 &
   python3 /home/pi/Desktop/SL2-6/Classification.py &
   exit 0
   ```

3. Install and update all required packages e.g. sshpass, tsfresh, sklearn etc before forgetting your internet Wi-Fi and make sure that Raspberry pi is only connected with Red Pitaya Wi-Fi.

Created by Taimour Yousuf 1276574 & Anil Kumar Gupta 1266270