SIT225 Data Capture Technologies

Distinction Task: Annotate smartphone accelerometer data by capturing activity images

---

Overview

The use of sensors is increasing in our daily lives which produces lots of data. Data becomes an asset and can be consumed by Artificial Intelligence (AI) and Machine Learning (ML) algorithms if data is labeled properly. Data capture protocol often keeps video logs of data collecting session so they can be used later during labeling the data. For example, consider a data collection scenario where electrocardiogram (ECG) data from human subjects are being collected while they are watching images of different emotions (sad, happy, and normal) to study the effect of emotion on cardiac responses. The whole data collection session can be recorded using a video camera to later annotate the cardiac response with corresponding emotion. Accelerometer data can be associated with human activities and if data collection session during activity can be recorded, it will help in annotating the accelerometer data.

Hardware Required

i. A compatible smartphone – Android or iPhone.

Software Required

i. Arduino IoT Cloud
ii. Python 3,
iii. Plotly Python library,
iv. OpenCV Python library to capture image from laptop web camera

Pre-requisites: You must do the following before this task

Week 8 activities in the unit site and 8.1P task.

Task Objective

In 8.1P, you have connected your smartphone to Arduino IoT Cloud and streamed phone's accelerometer data to the cloud dashboard. You have also connected custom Python device where Python script can receive your phone's accelerometer data to visualise N number of samples (readings) in Plotly Dash. For continuous monitoring, once the next N number of accelerometer samples are received, the dashboard graphs are updated and so on.

In this task, you need to capture images using your laptop webcam once fresh data has arrived and you are going to update the dashboard graphs. The graph data and the captured activity

image can be saved in files with matching filenames so the graph data can be annotated with corresponding activity for later analysis.

Steps:

1. Setup your smartphone, Arduino IoT cloud and Python script so phone's accelerometer data can flow through the Arduino IoT could to Python script in your computer.
2. Python script should contain code to capture photos from Laptop webcam, you can use OpenCV Python library.
3. For every 10 seconds of accelerometer data arrival from your phone to Python script, you will need to create a graph using the data points, capture an image using your laptop's webcam and display the graph and image in the dashboard. You can use Python CV2 library for capturing images using webcam.
4. You need to save 10 seconds accelerometer data (x, y and z variables) to a CSV file along with the captured image. Record timestamp from your computer in Python script and append timestamp to the filenames with a sequence number so data can be associated to activity images for annotation later. Filenames can be *1_yyyymmddHHMMss.{csv, jpg}* where a sequence number starting from 1 and increasing is followed by a separating underscore symbol and ending with timestamp. The choice of 10 seconds data is just to make sure enough time can be allocated for an activity; you can have your own choice with proper justification.
5. Repeat steps 3 and 4 for more than 30 minutes and perform at least 2 distinct activities (let's call it activity 1 & 2, you should replace them with real activity name such as waving and shaking) along with no activity in front of your laptop camera and with smartphone at your hand doing activities. Try to have a balanced number of activities of these types – no-activity, activity 1 & 2.
6. Create a CSV annotation file with 2 columns where the first is the filename and the second is the activity label such as 0, 1 or 2 which corresponds to no-activity, activity 1 and activity 2. You can decide an activity type by looking at the captured image.
7. While performing step 6, deciding activity type by viewing image may create confusion and you may find using video would have been more convenient to decide activity type, however, you do not have video log. At this point you need to analyse the data to find patterns which are close to patterns of any other activities. Performing step 8 below would help you perform this step 7.
8. Analyse activities of 3 types mentioned above. Try to find general patterns in the same type of activity (say activity 1) using multiple instances of the same activity. Contrast patterns between i) no-activity and activity 1, ii) no-activity and activity 2, and iii) activity 1 and activity 2.

Submission details

Q1. Describe your data collection protocol and annotation mechanisms justifying their efficacy and shortfalls, if any.

Q2. Describe your data analysis on how activities can be differentiated. Are patterns of each activity unique compared to other activities? Justify your findings by presenting multiple instances of patterns and activity images per activity type.

Q3. Show your annotation file you have created and explain instances where it was difficult to identify an activity from image and a decision was made by analysing the data. Discuss this process in a separate section and be critical.

Q3. Create a video in Panopto/CloudDeakin showing your program execution, Plotly Dash updates with graph and activity image, pattern analysis and share the video link here.

Q4. Create a subdirectory 'week-8.3D' under directory 'SIT225_2024T2' in your drive where you copy the Python script file which contains Plotly Dash functions, Arduino sketch file if any, data file and the generated graphs. Commit and push to changes to GitHub. Include the link to your repository here with a GitHub page screenshot of weekly folder content. A tutor may try to access your GitHub link, if necessary. Give access to your tutor by adding tutor's email address as a collaborator of your private repository.

Instructions

Consolidate outputs following the submission details above into a single PDF file.

Submit your work

When you are ready, login to OnTrack and submit your pdf which consolidates all the items mentioned in the submission detail section above. Remember to save and backup your work.

Complete your work

After your submission, your OnTrack reviewer (tutor) will review your submission and give you feedback in about 5 business days. Your reviewer may further ask you some questions on the weekly topics and/or about your submissions. You are required to address your OnTrack reviewer's questions as a form of task discussions. Please frequently login to OnTrack for the task ***Discuss/Demonstrate*** or ***Resubmit*** equivalent to fix your work (if needed) based on the feedback to get your task signed as ***Complete***.