Department of Electrical, Computer, and Software Engineering

Part IV Research Project

Project Compendium Index

Project Number: 129

Computer Vision for Assistive Technology

Dennis Cheung

Peter Joe

Muhammad Nadeem

Xuyun Zhang

23/10/2019

1. Project Summary

We developed a posture recognition to assist the elderly throughout the year and this section presents the project journey. More details can be referred to the actual report.

1.1. Project abstract

A large portion of our elderly population in New Zealand live independently. Unfortunately, we also found out that almost 1 in 3 elderly falls annually causing minor to serious injury. In addition, we discovered that while seeking caregiving services is a popular choice, not every family is financially capable of hiring a caregiver. Thus, we developed a computer vision model that can recognize four different common postures. Images are first captured by a webcam that is programmed by a raspberry pi. The images are then sent to the cloud server with our recognition model equipped with it. The recognition model is developed using a python library called ImageAI. The overall recognition is divided in two parts, 1) Human detection and 2) Posture recognition. We used a pre-trained model provided by ImageAI to perform human detection. They have used RCNN to extract the area of interest so after executing it, only the human body part will be left on the image. Then we used a model trained by ourselves to perform posture recognition. The training of our model was done by feeding many of our pre-labelled images for each posture. Finally, information of our results is displayed on a phone application.

1.2. Division of Work

My task was to complete the smartphone application which is to display all information processed and collected by the recognition server in a human-friendly format. Peter's task was mainly focused on machine learning section including finding datasets and training the actual recognition model using python.

1.3. Project Journey

Our literature review was heavily focused on papers that attempted to use some forms of assistive technology to perform activity recognition to assist the elderly. Papers we found usually had innovative ideas and satisfying results but lacked supports which are not very helpful for the elderly. Hence after we finished our literature review, we decided that we would implement a mobile application to display real-time information and provide adequate advices when needed to the elderly's family members.

As mentioned in Division of Work section, I mainly focused on developing the mobile application where as Peter implemented the critical machine learning section. While our machine learning algorithms were still under developed, I began looking for a suitable platform to develop our phone application. Initially, Android Studio attracted my attention the most as it is widely known and is proven to be working platform. However, I as a programmer who had developed a website using React before which is a JavaScript framework for programmers to make websites with ease, I had adopted a similar framework called React Native which is targeted to phone app development to create the phone app for our project. A platform called Expo was then selected to be used as our development platform as it simplifies things that other programmers struggle with.

During the implementation phase, I faced a few troublesome problems. First, I had to constantly search for functionalities/APIs that are provided by the framework since it was my first time using it. For example, I wanted to create a pie-chart to display the number of recognitions of different postures. Unfortunately, the package I was using was not capable of performing certain aspects while it claimed it could and took me several hours before realizing the issue. I learned a lesson from this as to not fully trust the documentation provided and should only treat it as a guidance. Second, as Peter used his home desktop as the hosting server, whenever he decided to not turn on the server to do his personal activities, I would be unable to test the connections between my phone app and his model. The complicated things between us as I am a night-owl person. I eventually had to correct my sleeping schedule back to normal to synchronize our working hours. Finally, we attempted to complete remote control for the user to turn on/off the camera the day before our exhibition day which caused us a hassle, but we ultimately overcame it with patience.

1.4. Results

Ultimately, I created four different screens, 1) Login screen (Fig.1), 2) Detail screen (Fig.2), 3) Log screen (Fig. 3) and 4) Setting pop-up screen (Fig.4).

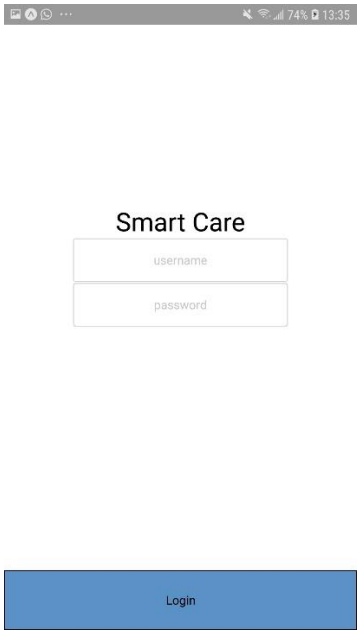

Fig. 1. Login screen

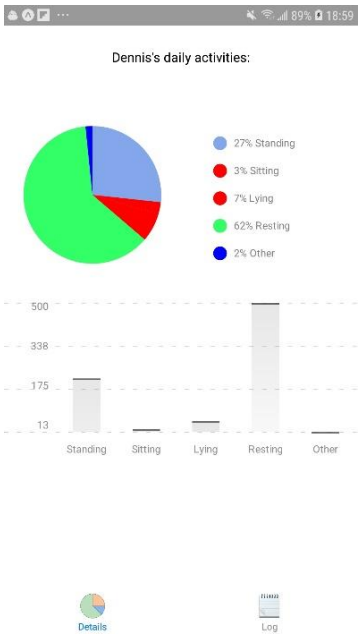

Fig. 2. Detail screen

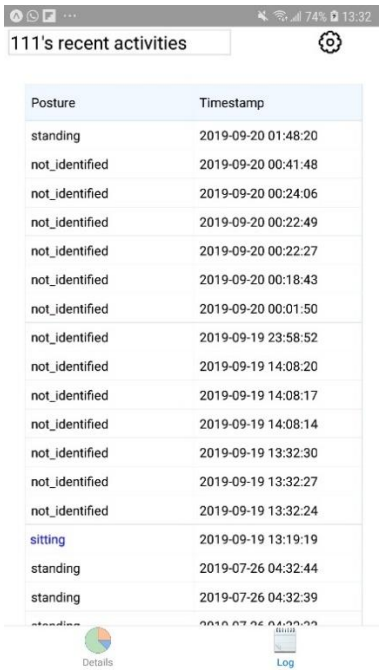

Fig. 3. Log screen

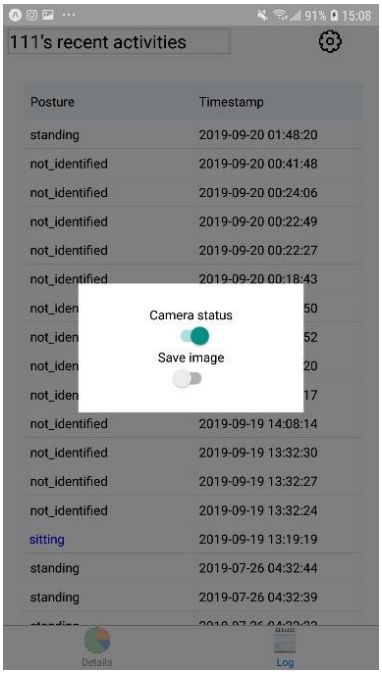

Fig. 4. Setting pop-up screen

Login screen (Fig.1) as the name suggests requires the user to first login before being able to access any information. The reason we have a login screen is because when we have multiple cameras monitoring different elderlies, we would want to ensure privacy for our users and the elderlies. Detail screen (Fig.2)

displays the information we collected in the database in a graphical format. The information is regularly updated from Peter's side when a new posture is recognized. Log screen (Fig.3) lists all posture recognized in the past and provides a setting button on the top right. Clicking the setting button will create a pop-up screen shown in Fig.4 which allows the user to determine whether they want to turn on/off the camera remotely as well as to record real-time images of the elderly who is being monitored.

1.5. Conclusion and Future Direction

The existence of the phone app is significant because as I mentioned in Project Journey, we discovered that many papers did not consider providing supports for the family members of the elderly. In addition, we provided remote control to the users to further protect privacy of the elderly. Finally, the phone app still has high potential for future development such as making it looks nicer or displays information in a more effective way. Therefore, our phone achieved what other literatures lacked and made our project more appealing.

2. List of Files

This section provides an in-depth guidance of how to execute/change the code.

2.1. Requirements

There are several items that are essential to setup the project.

1. Node.js
2. A phone charging cable
3. A smartphone

2.2. Steps to run

All components can be found inside the folder “P4-Phone-App”

1. If you don't already have Expo cli, open terminal and paste: npm i expo-cli
2. npm run
3. Connect your smartphone to your computer and ensure debugging mode is turned on (Each smartphone has a different way of enabling debugging mode)
4. When expo cli is finished initializing, you will see a huge QR code. Press a on your keyboard
5. A connection should now be starting up on your cellphone.
6. Guest account: 111, password: 222
7. If "server is not available" shown up, you may be required to change a small piece of code to bypass the login screen
8. stence of the phone app is significant because as I mentioned in Project Journey, we discovered that many papers did not consider providing supports for the family members of the elderly. In addition, we provided remote control to the users to further protect privacy of the elderly. Finally, the phone app still has high potenti

2.3. How to bypass the login server when it is down

When the server is down, all information is not available for access. This would require you to hardcode some values in order to view/use the app. Follow the steps listed below:

1. Navigate to Navigator.js

2. Comment out the line: const AppContainer = createAppContainer(AppNavigator);
3. Uncomment the following line
4. npm run

p.s. Although this would bypass the login screen, the app would be trying to connect to the server and this might cause expo to produce annoying warnings on your screen. In order to resolve this, comment out the function "componentDidMount" in both HomeScreen.js and DetailScreen.js

2.4. Other changes

Values for the pie chart and the bar chart are already hardcoded for your convenience. To change it back, follow the steps listed below:

1. Navigate to DetailScreen.js
2. For the bar chart, comment out the variable “data” inside chartData and uncomment the data variable underneath
3. For the pie-chart, inside thevariable “pieData”, change the variable <value: 24> to <value: {"standing"}> or any other postures