Mobile Computing Laboratory Report

Francesco Corti
TU - Graz
Padua, Italy
corti@student.tugraz.at
12002944

Christian Seifert
TU - Graz
Graz, Austria
christian.seifert@student.tugraz.at
01607219

I. Application Description

As we decided to go for Option 1, the goal of our application was to predict what kind of activity the user is currently doing. The two activities that can be predicted with our first model are Sitting and Standing.

The four activities which can be predicted from the second and the third model are Sitting, Standing, Walking and Walking Down the Stairs.

These activities cover a wide variety of movements and can be therefore predicted correctly in everyday situations. As the estimation process requires a model in order to be able to predict, training data as well as verification data needs to be gathered, which the application supports.

Furthermore, the application is able to train and use a personalized transfer learning model. This makes it possible for the user to predict his movements with the smartphone in different positions not covered in the training set. This is done without having to retrain the whole model but by only updating the weights of the last part, called head.

A. Motivation Statement

As neither of us have had any previous experience regarding Android Application Development we decided to go for Option 1 as it provided more support and a better idea on what can be done in a suitable amount of time.

Furthermore we used this Option as a chance to get to know and learn Kotlin. In addition to that, we both were interested in deepening our knowledge about machine learning deployment which was covered by Option 1. Due to all of these reasons we decided to go with Option 1.

II. Algorithms and Models

The first task we needed to solve was data collection and training of a simple model, which we did via the sensors of the smartphone. The data it provides are stored into a Kotlin Room database and can be saved into a file by pressing a button in the GUI.

The data collected was then imported into a python file and a tflite model was generated. In order to achieve good results we implemented a Feedforward Neural Network for our first model. We used with one hidden layer and two output units to produce the probability of the two actions that we wanted to predict (Sitting and Standing).

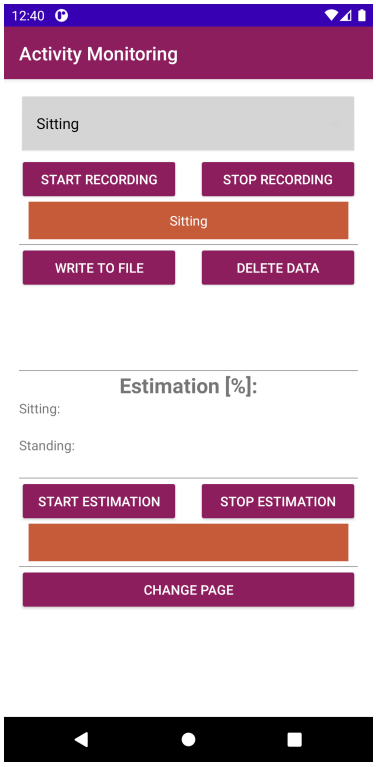

Fig. 1. First Fragment of the Application for the first Task

After this the model was imported into the android studio project and used only after a GUI button was pressed in order to predict if the user was either standing or sitting.

To prepare the data and use the model without having the application exit unexpectedly or over strain the application we decided to implement a concurrent queue, into which we add our sensor data. This queue then asynchronously works with the model feeding it the prepared sensor data and handling the results so they can be graphically shown and read from the application output in the terminal. This follows the Producer-Consumer pattern and made us to decouple our application in an efficient way.

For the transfer learning part, which was the second task, we developed a completely new fragment. We added three sets of buttons in order to start and stop the data collection, training and inference of the transfer learning model and a

Fig. 2. Second Page of the Application dedicated to Transfer Learning

section dedicated to showing the estimation results of all 3 models estimating simultaneously as well as a button set in order to start/stop the models estimation.

The data collection saves new samples and convert them into the needed format used to train the transfer learning model. The sensor data obtained from the accelerometer are preprocessed by using a high-pass filter to eliminate force of gravity is applied, then for the second and third model the data are normalized by dividing them by the maximum value. Moreover for the first model the average and the standard deviation of the three axis are extracted as features. The data are taken with a frequency of 50 Hertz. In order to inference the models 200 values are needed, so the action estimation is done every five seconds. Also, to retrain the transfer learning model which has the generic model as base model five samples of data need to be collected.

III. Performance / Accuracy

As can be seen in the first model confusion matrix, which can be seen in figure 4, the output for sitting and standing is very accurate. The Accuracy reach the 100%, as can be viewed in figure 3, without overfitting on the validation data. This result was achieved because the data used for training and test were collected while putting the phone in the same position.

As you can observe from the second model confusion matrix, figure 5, bad performance results were achieved. This is due to the testing data not being recorded with the phone on

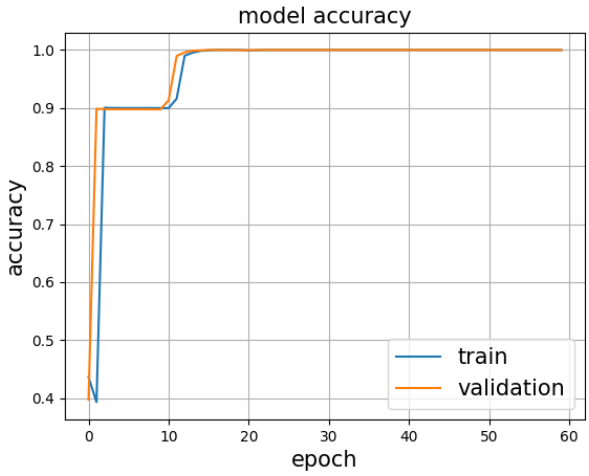

Fig. 3. Accuracy of the first model

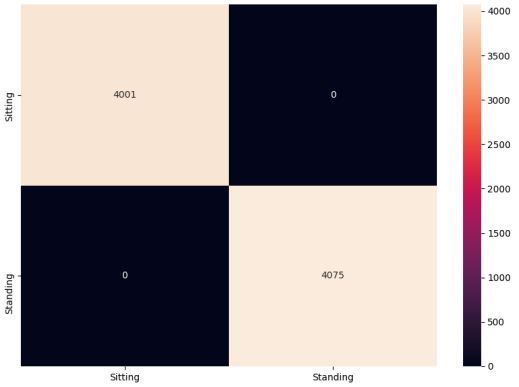

Fig. 4. Confusion matrix for the first model

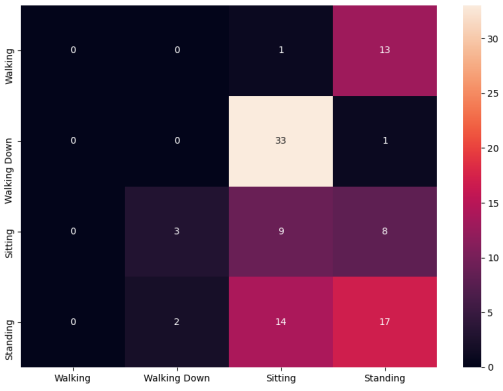

Fig. 5. Confusion matrix for the second model

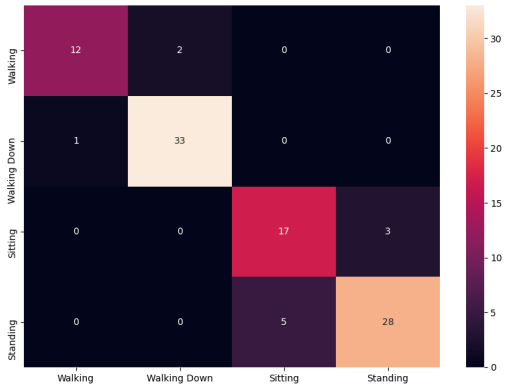

Fig. 6. Confusion matrix for the third model

the belt which was the position the generic model was trained on.

In contrast to that the confusion matrix for the third model, which can be seen in figure 6, shows a very good result with only very few miss-classifications. This shows that our retraining of the generic model head worked well compared to the poor results observed in the second model.

IV. Application Screenshots

Our application consists of two separate fragments with the first page committed to the first task, which is seen in figure 1 and the second page committed to the second task which dealt with the transfer learning task, seen in figure 2.

As explained in the previous segments, the data can be recorded with the first set of buttons. Above this set of buttons the activity type can be picked via a drop-down menu which is locked while recording to prevent the user from changing the activity by accident.

Data can be written into a file and the database can be emptied in the case of user failure, but before deleting the data a extra popup window needs the confirmation from the user in order to delete the data, preventing miss-clicks or unwanted actions. The status results can be seen in figure 8.

When estimating the probability, the result is shown in the GUI, representing how likeable it is that the user is performing a specific activity. This can also be seen in figure 9 where the user is currently standing.

On the transfer learning page there is an activity drop-down menu in order to label the data for the following training.

The three button sets are already described in previous segments. The last section of the transfer learning page shows the estimation results of all the three models and gives to the user a better grasp of the performance of all them. These three models represent the model of our first task, the generic model and the trained transfer learning model.

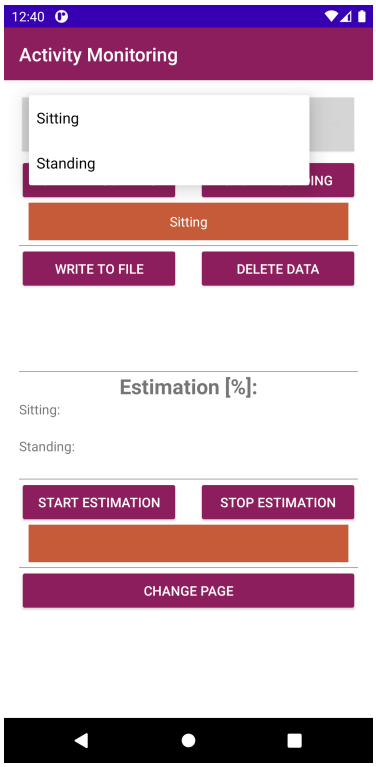

Fig. 7. Available Activities in the Drop-down for data recording

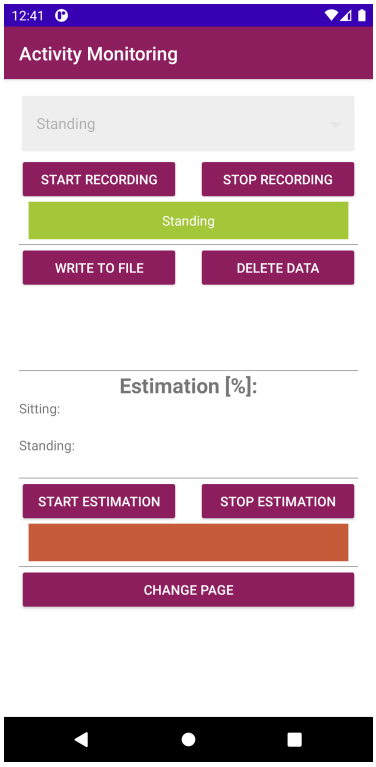

Fig. 8. Recording Data for generation of the first Model

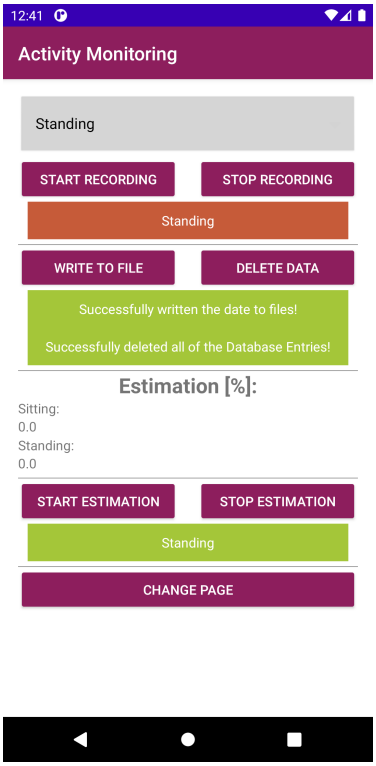

Fig. 9. Estimation on the first fragment with the first Model

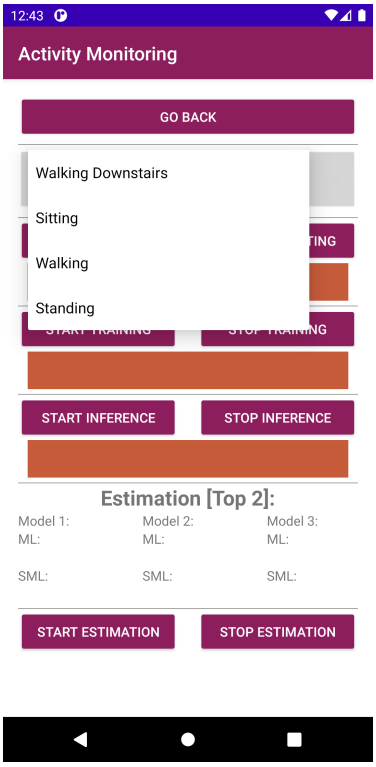

Fig. 10. Available Activities in the drop-down for data collection for the transfer model

Fig. 11. Estimation Results shown in the according Tabs

V. GitLab Repository Explanation

Our project can be found on the TU-Graz GitLab under the following Link: GitLab Project If there are any problems with being able to look into the project please let us know. (Project is Public)

VI. Application Limitations

We tried to make our app as crash resistant and stable as possible, but due to this being our first app there are some cases that lead to a crash of the app. One explanation for that is definitely implementation of functionalities in an ”un-android” like way due to our knowledge with other programming languages and lack of knowledge with android. Our first model is only trained for two activities as we were struggling to get our application to run without crashing before the first task deadline and wanted it to work in a simpler way rather than not at all. Furthermore we have had troubles with the application behaving weirdly after tilting the phone so it switches into horizontal mode and back. We couldn’t quite figure out why this was the case but this can be easily avoided.