M.S. THESIS

Development of a life-logging platform using smart devices and its applications

스마트폰과 웨어러블 기반의 라이프 로깅 플랫폼 및 그 응용

BY

HYUN-JUN KIM

FEBRUARY 2017

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. THESIS

Development of a life-logging platform using smart devices and its applications

스마트폰과 웨어러블 기반의 라이프 로깅 플랫폼 및 그 응용

BY

HYUN-JUN KIM

FEBRUARY 2017

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Abstract

With the advance in sensing and related technology, smartphone and wearable device became the important machine for people's convenient life with extensive data of user information like location or motion. These data are closest data from the user, collected literally on user's hand. So using smart devices, we can figure out user's status and log user's daily life pattern more precisely. Applications about life-logging with smart device are got attention in this context.

We propose life-logging framework which collects as various data as possible. In our framework, we propose not smartphone life-logging platform only or wearable life-logging platform only, but, an integrated life-logging platform that uses both of them. Proposed platform collect many data as possible with protecting generality, which means that user of various OS version can use this platform. Our platform can be applied in many application like a psychological experiment.

To prove our platform is adaptable to many examples, in this thesis, we propose also inferring mood application that infers user's current change of mood such that activeness-whether he or she is more active or less active than past- or happiness. We collected Application extracts features from collected data using proposed method.

keywords: Smart phone, Wearable, Life-logging, Mood inference

student number: 2014-22553

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

Life-logging, capturing people's daily life and storing it, has been done in forms of writings like diary or image like picture or photo in the past and video or audio information using an additional device like a video camera in today. A life-logging device also has technical advances; there is some trial to solve problems like the inconvenience from the limitation of activity range using a wireless webcam. SenseCam [1] is invented to help people having memory impairment like Alzheimer's disease. InSense [2] is a life-logging system which considers user's activity. It takes pictures based on pre-defined interest and importance of user using user's activity. Narrative Clip [10] is a small device which has no inconvenience to wear like a necklace, so it has commercial success that Narrative Clip 2 is developed and on sale.

Advances in wearable device bring a great effect on life-logging systems. Wearable devices mean electric devices which person can wear such as a watch, belt or clothes, and use. By highly developed sensors and efficient power consumption technologies, wearable devices come to people's daily life with a various product. Like Apple watch or gear series, many companies commercially produce wearable devices. The biggest advantage of wearable devices is that wearable devices are attached to the human body. Attached to the body, wearable devices can collect physical data of person and health data like heart rate and body temperature in real time. SenseCam and InSense also collect these non-visual data and apply these to their life-logging system. In a case of a smartwatch like Apple Watch, it recognizes user's activity and provides information requires to a user using notification.

A smartphone is an important part explaining why a wearable device is advanced so fast. In the past, a mobile phone is used as the mean of communication, and nowadays it is necessary parts of daily life as a web browser, mobile game platform and even device of virtual reality. Interest and concern of smartphone become research of related works, in the process of research, sensor and battery are also studied and affect wearable devices. In life-logging, a smartphone can provide new kinds of data to the system, sensing data from device usage using a programmatical method. As many people keep the smartphone in their hands, it can be performed as a measurement of life pattern. In statistical data from KISA, people used smartphone 1 hour 46 mins a day in 2015, that increases seven mins compared to previous year [12]. This data is obtained from direct smartphone usage only, indirect usage like listening music is more than statistical data. Smartphone also collects other sensor data, such as acceleration or GPS, smartphone can be life-logging devices which take advantage of mobility

In this paper, we present life-logging platform that uses smart devices. Smart devices include smartphone and wearable device, a smartwatch. The entire platform collects data from both devices independently and also can call alarm to user's smartwatch to get survey data. A survey can be any type such as health or location, in this paper, we purpose mood survey app as an example of our platform. Also, we tested short period and collected data with mood survey data. Using collected data, we present simple mood inference application as an example. For example of mood inferring, some research is motivated as feature extraction or date type.

Jayarajah et al. have presented LiveLabs [6], which is not a life-logging platform, but a testbed for mobile sensing and behavioral experimentation. It provides smartphone application to many people in certain places which collect data using a smartphone. The app can provide survey notification for behavioral experimentation.

Emotion Sense [9] is an Android application which measures physical activity, sociability, and mobility. It also collects user's self-reported moods, thoughts, and symptoms. Emotion Sense is opened to everyone as source code, so many researchers use Emotion Sense for Psychology purpose.

Moodscope by LiKamWa et al. [3] is mood measuring application which collects mood by user survey. MoodScope collects data about social communication like email or SNS and web browsing history and other application usages. LiKamWa also analyzes collected mood data and estimate mood using collected data from the app. As it uses personalized model for increase accuracy and long-term data about two months with 32 participants, a result is not such bad.

Pielot et al. [7] proposed application named borapp detecting one mood, boredom, using smartphone usage. It mainly uses communication notification data like LiKamWa and battery and cellular data usage. Pielot claims random forest [4] is the best classifier of boredom, so we use RF for our example.

Meanwhile, Valenza et al. [8] present mood recognition monitoring system using wearable devices includes heart rate sensor. Although it applied to bipolar disorder and needed commercially not available devices, it shows that mood recognition can be possible using heart rate.

In addition, there are a lot of researches about life-logging with smart devices [15, 14, 16, 17, 18]. But, Previous research focused on smartphone only or wearable device only. But our paper present life-logging platform using a smartphone and available wearable device, smartwatch. And we set our goal to the Android version that no Android version problem occurred as possible and privacy. Our application's another goal is to use least privacy related data, and it is IRB-approved.

This thesis is structured as follows. In Chapter 2, we briefly review Random Forest for an application. Chapter 3 describes our life-logging platform both part of smartphone and wearable. In Chapter 4, we tested our platform by collecting data from users and applying it to mood inference. We conclude our thesis by summarizing our platform and inference model and discuss limitation of our platform.

Chapter 2

Preliminaries

In this chapter, we briefly review decision tree learning and random forest algorithms. The algorithms are not used for our proposed platform, but these used for application of our platform, mood inference.

2.1 Decision Tree Learning

Decision tree learning [5] is a learning method that uses a decision tree as a means of connecting classification results for a given input as one of the map learning. This learning method is one of the simplest and successful forms of machine learning. A decision tree represents a function that takes as input a vector of the attribute value and returns a decision. A decision tree reaches its decision by performing a sequence of tests. Each node in the tree corresponds to a test of the value of one of the input attributes and branches from the node are labeled with the possible values of the attribute. Each leaf node in the tree specifies a value to be returned by the function.

Determining attribute to use in each split node is key part of decision tree learning. In decision tree learning, attribute to split data is selected by finding the most important attribute that separates two different data with the best score. There are some approaches about how to measure score, and information gain is mainly used to measure score. Information gain is a change of two values, an entropy of parent and sum of entropy from separated nodes (child).

$$Gain(S,A) = H(S) - \sum_{v \in val(A)} \frac{|\{x \in S \mid x_a = val(A))\}|}{|S|} H(\{x \in S \mid x_a = val(A)\}) \tag{2.1}$$

$$H(S) = -\sum_{s \in S} P(s) log_2 P(s) \tag{2.2}$$

Completed Tree structure looks like Figure 2.1. A figure is a binary tree which is easy to construct. In each split node, all data are split whether an attribute is bigger than criteria or not. Decision tree learning algorithm can be expressed algorithm 1. PLURALITY-VALUE function chooses most common value among examples and IMPORTANCE function select which attribute is selected to split data in a single node. After construction of decision tree using given data, applying tree to other data is simple. If new data was given, compare criteria of each split node and find a class of leaf node that new data is arrived.

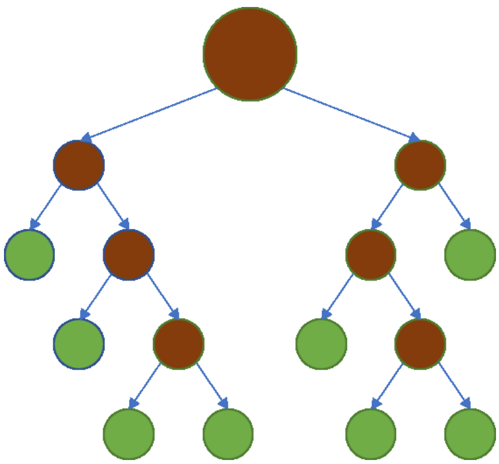


Figure 2.1: Decision Tree.

Algorithm 1: Decision Tree Learning

input : examples E, attributes $Attr$, parent_examples E_p

output: a Tree T

if $E$is empty **then**
  return PLURALITY-VALUE(E_p)

else if all$E$have the same classification **then**
  return the classification

else if $Attr$is empty **then**
  return PLURALITY-VALUE(E)

else
  $A \leftarrow \mathrm{argmax}_{a \in Attr}$IMPORTANCE$(a, E)$
  $T \leftarrow$ a new decision tree with root test A
  foreach value u_k of$A$ **do**
    $exs \leftarrow \{e : e \in E$ **and** $e.A = v_k\}$
    $subtree \leftarrow$ Decision Tree Learning$(exs, Attr - A, E)$
    add a branch to T with label $(A = u_k)$ and subtree $subtree$
  end
  return T

On some problems, decision tree learning encounters overfitting problem. Overfitting occurs when model trains too well with train data. As model learns the detail and noise in the training data to increase accuracy, the model cannot predict well when new data is input. Decision tree learning is easy to overfit given data so many techniques like pruning branch had been proposed.

2.2 Random Forest

In the previous section, we noted that there is overfitting problem in decision tree learning. To solve overfitting problem, the random forest is proposed by Breiman [4], it is part of ensemble learning. A base of random forest is decision tree learning. Input data is split in each node iteratively unless all leaf node is almost classified. Major differences with decision tree learning are three points, which is split method, the number of tree and technique called bagging.

In every split nodes, data is split by randomly selected criteria that classify data purely as possible. To find criteria, we calculate entropy for each classified data using randomly selected feature and find criteria that maximize entropy. That is, a designer does not need to design which feature is used to split at each node in the random forest. Node split function can be described following algorithm.

Only using randomness in node split does not improve prediction accuracy. Random forest uses not only one tree for classification, but many trees for classification. Leaf node distribution for input label in each tree is different with other trees. In decision process, each node suggests label distribution and most voted label is selected.

Bagging or bootstrap aggregating is the idea to address the overfitting problem. The main point of bagging is randomly sampling with replacement and voting. When train each tree model, train data is not directly used as input to construct a new tree. After random sampling with replacement, sampled data are input to construct a new tree. That is, each decision tree are trained with different data. Voting is to select a class that majority of tree classified when test data is input.

Figure 2.2 and 2.3 show how random machine learns in random forest. Figure 2.2 shows training process of random forest. When random forest model is trained using train data, data is randomly sampled with replacement. Sampled data is an input of one decision tree. On the construction of decision tree, randomly selected features is used to split data in each split node. Feature to split data is the feature among randomly sampled features that maximized information gain which means classify data well. The node having data of an almost same class is become leaf node and stop split. If all terminal node becomes leaf node, training one tree is ended and new tree with newly randomly sampled data from train data become a new input. Algorithm 2 shows train process in the random forest.

Algorithm 2: Random Forest Training

input : Training set $S_{tr} := (x_1, y_1), ..., (x_n, y_n)$, features F, number of tree N

output: Trained forest T

function `Train RF(`S_{tr}, F`)`

$H \leftarrow \emptyset$

for $i \in 1, ..., N$ **do**

$S^{(i)} \leftarrow$ A randomly sampled data from S_{tr} (with replacement)

$t_i \leftarrow$ TREELEARNINGWITHRAMDOMFEATURE$(S^{(i)}, F)$

$T \leftarrow H \cup t_i$

end

return T

function `TREELEARNINGWITHRAMDOMFEATURE(`S, F`)`

$N_t \leftarrow$ root node

while *true* **do**

foreach node n in N_t **do**

if data of node $n$is almost pure **then**

set n to leaf node

else

$S_p \leftarrow$ data subset of Sarrived node$n$

$f \leftarrow$ randomly sampled feature subset from F

$I_f \leftarrow$ Information gain from classified data using f from S_p

$f_{best} \leftarrow$ feature that maximizes I_f

$n_1, n_2 \leftarrow$ nodes split using f_best

$N_t \leftarrow n_1, n_2$

end

if $\forall n \in N_t$ is leaf node **then**

break

end

return t

Figure 2.3 shows how to use trained model to classify test data. Test data is entered each tree and data is determined by features of the split node when data has arrived split node. When data has arrived leaf node, distribution of the leaf node is an output value of each tree. After all tree returns distribution, class of test data is selected using whole distribution. Algorithm 3 shows how to classify given data briefly.

Algorithm 3: Random Forest Test

---

input : Given data S_{te}, Trained forest T

output: Class c

function `TestRF`(S_{te}, T)

- $d \leftarrow 0$
- **foreach** $t \in T$ **do**
  - $d \leftarrow d + \textsc{GetDistFromTree}(t, S_{te})$
- **end**
- $c \leftarrow$ class with maximum value from distribution d
- return c

function GetDistFromTree(t, S_{te})

- $n \leftarrow$ root node
- **while** n is not leaf node **do**
  - $n \leftarrow$ node selected in n
- **end**
- return distribution of n

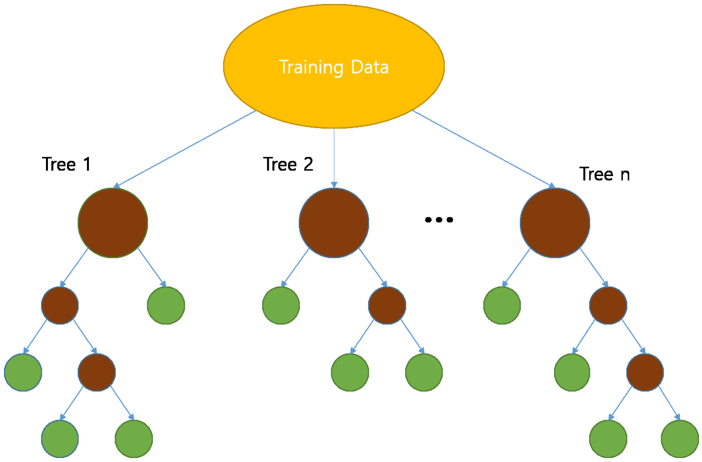


Figure 2.2: Conceptual diagram for training random forest.

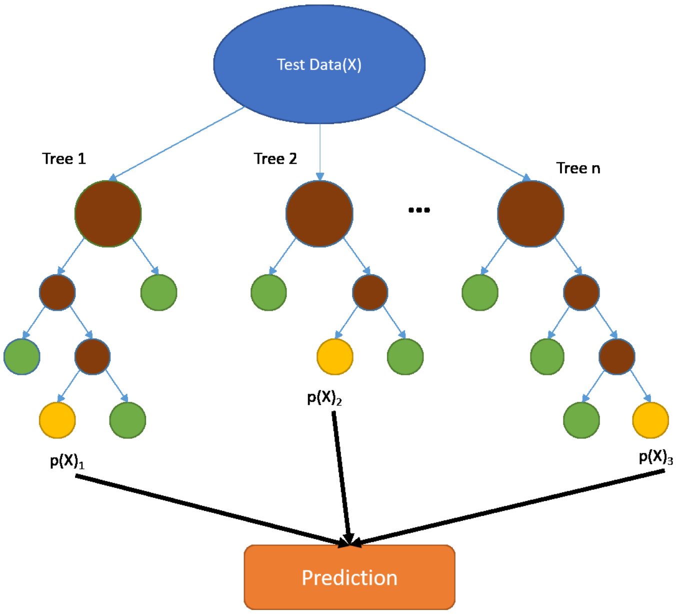


Figure 2.3: Conceptual diagram for testing random forest.

Chapter 3

Life-logging Platform

In this chapter, we explain our life-logging platforms. As illustrated in figure 3.1, our life-logging platform consist of two smart devices, smartphone and smartwatch, and their applications. By wearing smartwatch and carrying smartphone with applications running, a user can collect data easily. Our platform contains 3 applications, 1 smartphone application, 2 smartwatch applications. Smartphone application is sensing application, smartwatch applications are sensing application and survey application.

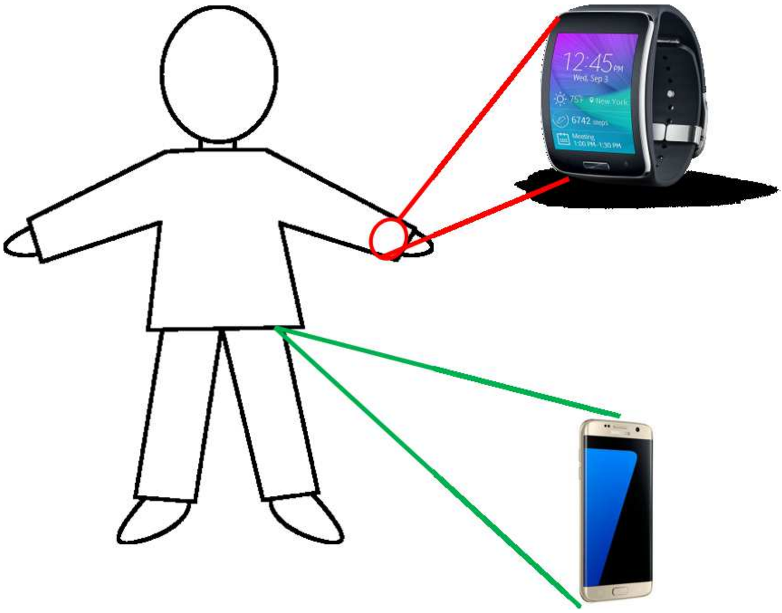


Figure 3.1: Diagram of Proposed Life-logging Platform

3.1 Smartphone

Smartphone applications is most important part of our platform because of its possibility. Smartphone can collected hardware sensing data such as acceleration and gyro from Inertial measurement unit or GPS data. But using some APIs, we can collect more rich information of smartphone usage. There two main smartphone OS nowadays, Android OS and iOS, we develop our smartphone platform using Android OS in this thesis as Android is more convenient to develop and maintain. One of our goals about platform is that our platform is available in more various Android smartphone. We tested our platform in various Android OS version and verified our platform works well in various versions.

3.1.1 Design

An important part of the smartphone platform design process is privacy. Privacy violation means collecting or exploiting private information about an individual, which means collecting information that may be sensitive to the user. Invasion of privacy may be an inevitable problem in life-logging in which a user's life is recorded as information such as a voice or text conversation made by a user via an instant messenger or phone call. Therefore, in this paper, we propose a platform that collects minimum information which might be sensitive to users. In addition, a series of processes to prevent privacy invasion can make a variety of devices to which the platform can be applied, because the operating system is gradually trying to provide limited information on privacy. For example, the Internet usage record can not be accessed by other applications, and only the details used in the Internet application are changed to be stored.

As smartphone platform is controlled by a participant, participant can start and stop logging service manually. Simplicity is a primary goal in UI design. For debugging convenience, we could display collected data on the screen. But since users only need to know whether platform works well, it is unnecessary to display all data on a screen. Only buttons are placed screen and a user gets pop-up messages and push notifications on upper notification bar when logging is started. Figure 3.2 are UI design of our smartphone platform. There are only 3 buttons, start logging, end logging, exit application.

3.1.2 Services

Smartphone platform consists of main activity and 10 data collecting services. Each service collects the different type of data and can be controlled separately. When experimenter does not want some services, easily disable the services in application building process. Collected data from each service are also stored separately to use easily. All services are restarted every 20 minutes, it is due to android OS that system often ter-

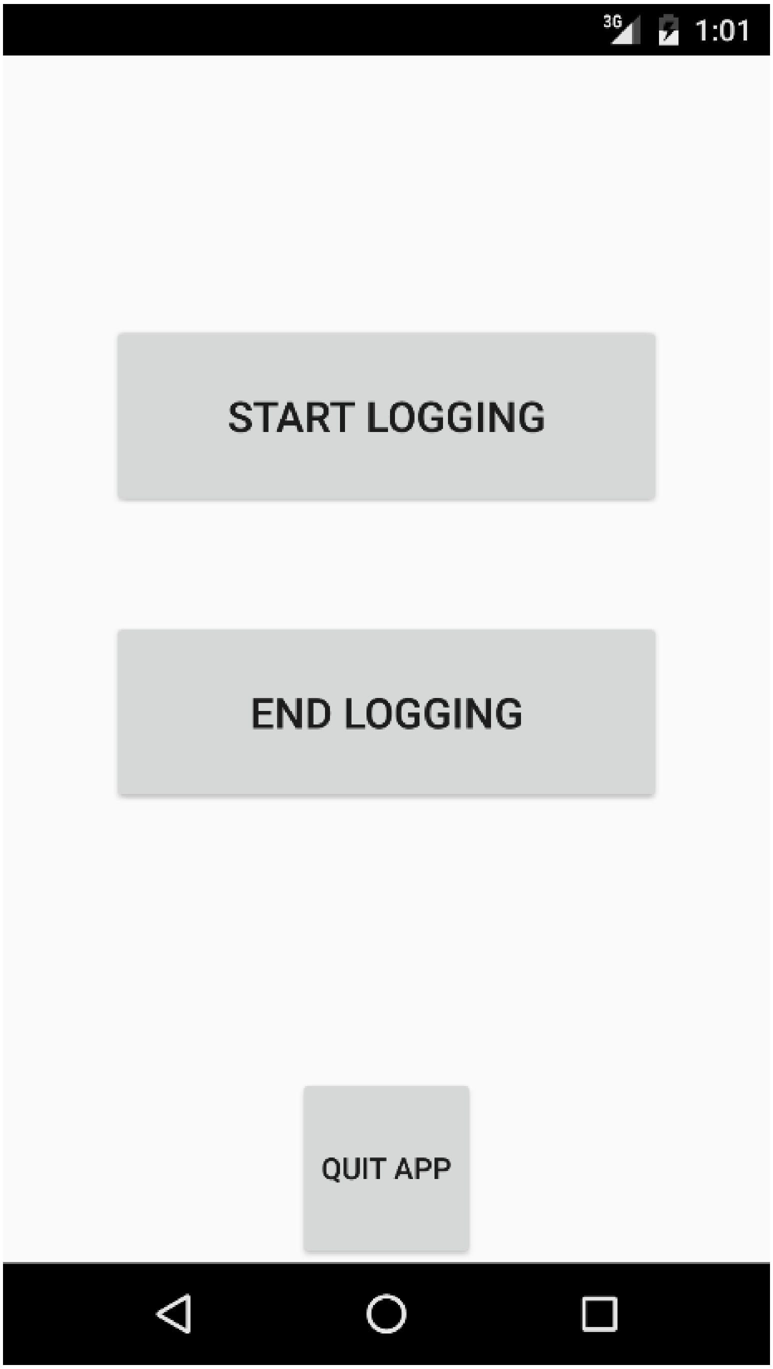


Figure 3.2: Smartphone Platform UI Design

minated background service without an alert.

Moving Service

Moving service is service that collects acceleration, gyro, magnet value from Sensor service. As these value can be used to estimate physical information, service collects data fast as possible. We considered that step counter is also collected, but for an older version, we delete this part. In version older than 4.4, Kit Kat, Android os does not support step counter API. The sampling rate is different for each Android devices; Android API does not support API to set sampling rate.

Location Service

Location service is service which collects location data using GPS. This service stores number of accessible satellite and GPS information such as latitude, longitude, altitude. To collect GPS data, a user must enable GPS, or service collect empty data. As this part uses GPS, most power consumption part of a smartphone except the screen, total application power consumption is high. It does not work well if a user is in a building. In that case, another service like Wifi service is used to collect location related data.

Notification Service

This service collects notification data. notification means all push notification arrived smartphone. It can catch notification even if user disabled pop-up, it only cares notification marked to upper notification bar. Notification service process not only notification but communications app. As other communications app don't provide APIs accessible to communication between users, we use notification push to collect communication data. To protect privacy as possible, all communication data are processed using the following sequence.

First, all notification sender are changed to an identical number. In cases of communication notification, all sender are displayed certain notification part, title. So extracting a title from each notification and changing to ID protects user's privacy. Other application notification also applied this process, and stored, but not used. Next, all content of notification is not stored except length information. And last, all data is stored after encoding to Base64. It is not a privacy problem; it is just protecting comma from the ticker, text to show notification bar without opening the notification center.

After above process, all notification data is stored secured. Smartphone vender providing SMS data also considered communication application.

Last part of secure privacy is that user can enable notification listener manually. Figure 3.3 shows a process that user enables and disables notification listener. Every time logging starts, a user must enable notification listener. After logging is over, a user must disable notification listener. Otherwise, an application is not quit and notify to disable.

Phone State Service

This service listens to phone event like someone call to a user. When the phone is ringing, service is collect phone number and stored it using previous explained changing. After the phone call is ended, service also logs time phone call is arrived and ended. Phone call event is also the trigger of another event, recording. Detail is in recording service section.

Power Service

This service is collected power related data like battery and screen. It collects current battery state every minute. It also logs events of battery and screen. Battery event occurs when a smartphone is charging or discharging, including AC/USB usage. Screen event occurs when a screen is turned on/off.

Figure 3.3: Enabling or Disabling Notification Listener

Recording Service

This service records surrounding sounds like conversation or noise. It can be very sensitive part of our platform. If a private conversation is recorded, it will invade user's privacy. After trade-off between privacy and detail, stopping record during the phone call is a compromise. We stop recording during user is got phone call using phone call arrived event. After the phone call is ended, recording service is resumed and recording is continued. Nevertheless, among the recorded sounds, there may be some conversation sound with noise from surroundings. This problem can not be solved until the recording is given up. As an alternative, transformed data is used instead of original recording. In this case, Fourier transform is performed after recording sounds for real-time life logging. Fast Fourier transform takes some time and it interferes real time life logging. The sound is recorded with 8kHz rate because of android API and maximum length of entire recorded file is 20 minutes, restart term of main activity.

RunningApp Service

Running app service is collected information about running applications. As this API is blocked by Google, we use third party library that accesses current running applications. It might be not general, but this library uses the kernel that all Android OS must use, this service can be applied other phones. It is due to Linux kernel of android os stores running process data as a file. We tested this service in all other kinds of phone available, and this services worked well. Category information is not accessible using API, category of application is fetched in post-process after logging.

Signal Sensing Service

This service collects signal data, such as cellular location, Bluetooth connection information. As some cellular communication has different communication style like CDMA and LTE, some value might not be collected. Bluetooth connection information is collected even if a smartphone is not connected to the Bluetooth device. If connectable Bluetooth device found, information of the device is stored. The information contains address and name of device and signal length between the device and smartphone.

Softsensing service

This service is collect data about soft sensing. A soft sensing means data not collected by hardware only, but other programmatical methods of collect data. Collected data are memory data and data TX/RX data. Data in clipboard, history data and bookmark of the web browser, a search keyword from web browser was collectible, but it is not available as version update [11].

Wifi service

This service is collect Wifi data, such as all AP user can connect or AP's current status - AP's current signal strength, AP's BSSID and SSID. As SSID can be any types of string so including comma, special character, etc. So SSID and BSSID is collected using Base64 encoding, which might look strange, and it stored. AP data can be used as location related data instead of GPS data from location service. As most of AP are in the indoor location like lecture room inside the building, AP information can be a substitute for GPS data in the indoor location that GPS data is not available. Wifi AP data is collected even if a user disables Wifi function.

3.2 Wearable Device

A wearable device we use is Samsung Gear S, which has 1GHz dual-core CPU, 512MB RAM, and 4GB internal storage. As its OS is Tizen, the applications are also developed using Tizen web application SDK. Gear S can be connected with a smartphone and communicated with a smartphone through the unique protocol, but it requires specific smartphone provider. For generality, communication part between smartphone and Gear S is not considered. Later devices like gear S2 and S3 do not require specific smartphone provider.

3.2.1 Sensing application

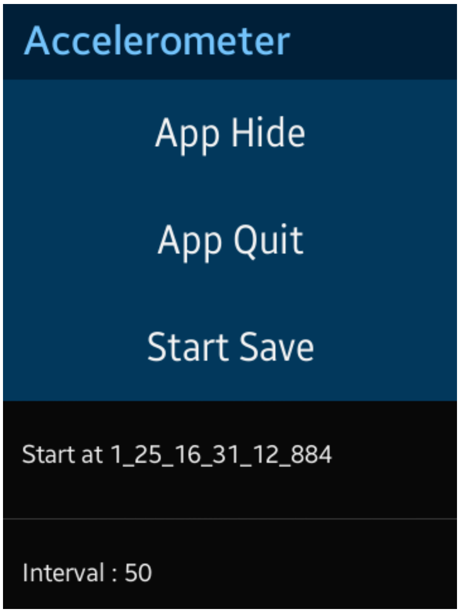
Gear S can collect various data from pose-related data to environmental data and health data. Collectible data from gear S are acceleration, angular velocity, magnet field, light and ultra-violet density, air pressure, heart rate. All these data have different sensing rate, so all data is collected asynchronously with own sensing rate. But when we use multiple loggers, memory leakage occurs, and application is quit abnormally. So all collected data are stored buffer and saved totally in every second. For the convenience of post-process, a flag is also logged with each collected data.

Sensing application has some limitations coming from device's problems. The main problem is that some data are collected when a screen is turned on. Not collectible data when a screen is turn off are magnet field, light, ultra-violet, air pressure. Only acceleration, angular velocity, and heart rate are collectible when a screen is off. It is due to Tizen API which is not working when a screen is off. As latest gear S devices were came out, API support for Gear S has been discontinued, so collecting problem can not be solved with the direct method. The indirect method to solve this problem is to turn Gear S screen on always. But, this solution is not applied to the application by other problem discussed later.

The second problem is battery usage related with a battery capacity of Gear S. With the sensing application, Gear S can collect data less than five hours even when fully charged. The reason that turning Gear screen on is not a proper solution is also power usage problem. If the screen of Gear is always turned on, power usage is increased and a platform is available less time than current. As gear S is not available when charged, attaching auxiliary battery is not possible and reducing power consumption is needed. To reduce power consumption, sensing application doesn't collect GPS data and record surrounding sounds. In spite of these trial, time that Gear collects data is extended to 6 hours.

Our application is controlled by users, not application developer or experimenter, so user-friendly UI design is needed. Unlike an Android application, Gear application doesn't have the proper feedback like push message through notification system to check application is fine, so UI to display that data is well collected is required. Since all data from gear is numeric, collected data was displayed at a screen. A user also has to start and stop logging manually, so buttons are placed in the application. Buttons were placed at the bottom of the application in prototype, but button could find until scroll was down, so buttons are at the top of the screen. Figure 3.4 shows UI design of sensing application.

(a) Data



(b) Button

Figure 3.4: Sensing Application

Also, to collect data on our platform, an additional experiment is considered. If another Gear survey application exists, sensing application launches survey application every specific time. Interval is applied after the survey is ended. In next subsection, survey application that collects mood related information will be proposed for our application, mood inference.

3.2.2 Survey application

Sensing application launches survey application every particular time, 30 minutes in our thesis. As we proposed mood inference application for life-logging platform, our survey application is related to mood. We made following survey application to collect participants' mood-related information. Survey application is written with English and Korean language and looks like following figures.

Figure 3.5 is the beginning part of the application. When the application starts, a participant can choose whether to respond to the survey or postpone the survey in (a). If a participant wants to postpone survey for some reasons, a participant can postpone by clicking postpone button. A participant can also set how long to postpone the survey time in (b). After time set by a user, survey application is launched again and a participant can start survey or postpone again.

Figure 3.6 is the middle part of the application. At the start of the survey, participant responds to location questionnaire in (a). A participant can choose a proper location by clicking arrows, or enter the location manually. Once the new location is stored, the location is added the existing list of location and reused. After answering location, participant replies to activity and number of people question in (b). People information is divided into two parts, one for people around participant and one for companions. Former one means the number of people just exist in the same location, latter one means the number of people acting same activity with the participant. Entering activity information is the same process of location, the participant can choose an activity or enter the activity manually.

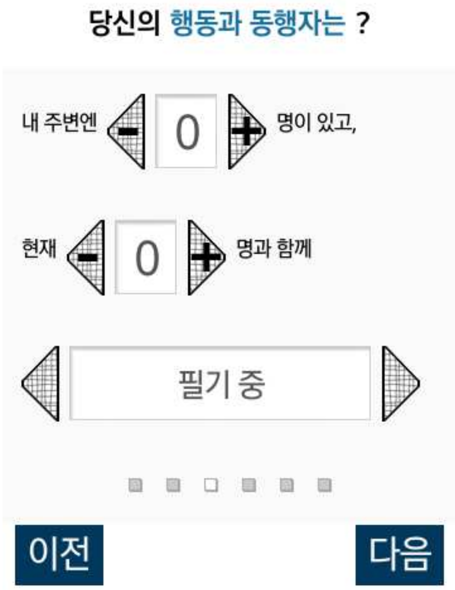
Figure 3.7 is the last part of the application. Health information responded by the participant in (a) contains heart rate and health status. Heart rate is a value taken from sensing application and can be measured manually using the button if value seems un-

(a) Main

(b) Delay

Figure 3.5: Survey Application - Start part

(a) Location



(b) Activity and Companion

Figure 3.6: Survey Application - Location and Activity with Companion

reliable. Health status is subjective and relative value. Participant select current health status compared to 30 minutes before. If participant's health seems to be worse than 30 minutes before, -1 is checked. Mood information, activeness, and happiness, in (b) also similar to health status in input method. If participant's mood seems to be better than 30 minutes before, one is checked.

(a) Health (b) Mood

Figure 3.7: Survey Application - Health and Mood

Chapter 4

Example : Mood Inference

Using proposed platform, many application about life-logging is available. Especially, using a function in gear sensing application that launches survey application with the particular interval, psychological experiments are also available through the survey. In this thesis, we propose mood inference application using our platform. As human mood can be affected by external stimulation like talk with a friend or where the user is, it might be represented in smartphone usage and data collected with smart devices. In section 4.1, we will explain data collection process of user's daily life. And we will propose our mood inference application in section 4.2.

4.1 Data Collect

Collecting data is an important process in our thesis to verify the proposed platform is valid, in additional to simply collect data for mood inference. Six people participated data collection, and all devices were the different models to verify our platform's validity. All participant were graduate students, and they spent most of their daily life on a university campus. Since their daily life is limited to campus, setting participant to graduate school has an advantage of ease to categorize their location and activity and disadvantage of bias coming from similar life patterns. Data collection took place over two weeks, collected data was 7 8 days data, except for weekends and days that participant did not collect data for the personal reason. Each day, data were collected for 6 hours, from when a Gear is fully charged to when Gear is fully discharged. In the process, the participant responded to the survey application described the previous chapter every 30 minutes.

Data collected from our platform is used for mood inference with collected survey data. Survey data contains location data, activity data, companion data, health data and mood data. As mood is too subjective to describe or quantify, we use simple labeling method for participant's mood. Collected mood data is consists of two data, activeness and happiness, each data has three labels, down and no change and up. Down means participant's mood is worse than before, 30 minutes in a thesis. Up means, the mood is better and no change means mood is not changed compared to the past.

Figure 4.1 and 4.2 are distribution of collected mood. 4.1 shows distribution of activeness and 4.2 shows distribution of happiness. Two figures show the similar distribution of mood. In many cases of the survey, the participant is not more active or more inactive. Happiness also shows similar, participant is not more happy or less happy in many cases. Counts of two state, ”up” and ”down,” are similar, and less than a count of ”no change” state of mood.

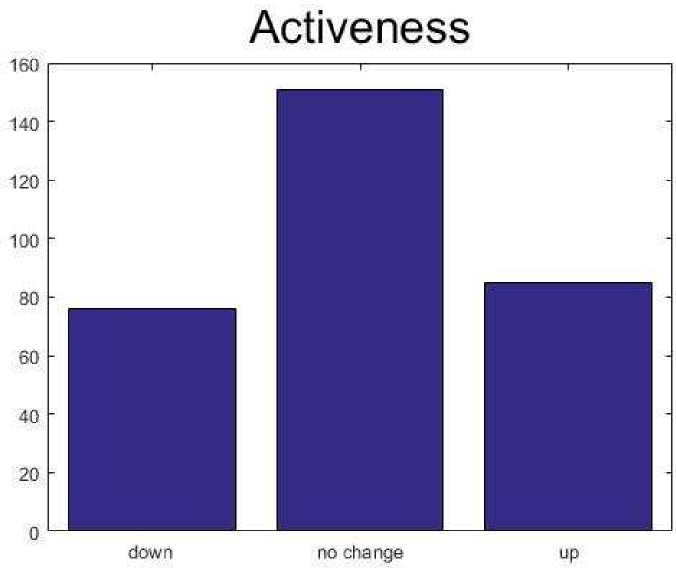


Figure 4.1: Distribution of Activeness

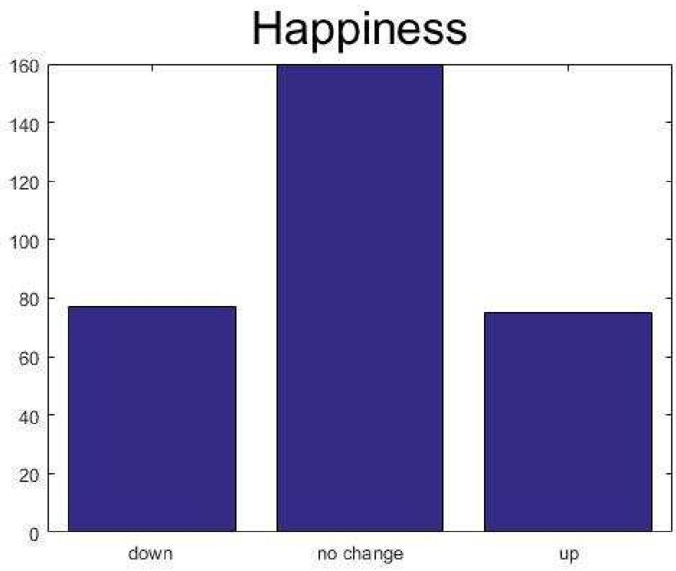


Figure 4.2: Distribution of Happiness

4.2 Mood Inference

Inferring mood is a very challenging application since mood is subjective and not quantitative value. So we tried to infer participant's mood is change or not compared to the past, 30 minutes before. We extract simple but massive features from collected data, train with features and tested. As mood data and smartphone data are sensitive to privacy invasion, there is no comparable data set or completely comparable application. We will explain our proposed application and analyze inference result, but not compare with other application.

4.2.1 Feature Extract

To infer mood, we need to extract features from collected data. As collected data is very plentiful, a dimension of extracted features is about 1469. Some features are from MoodScope[3], Borapp[7], Monarca[8] and Jigsaw[13], other is designed from collected data. Types of used feature are various, from acceleration and gyro to social data like SNS and SMS. AGM features, acceleration and gyro and magnet field, are Jigsaw features.

We focused on smartphone usage to infer current mood change, so features related with smartphone usage are mainly used. Social data like SNS and SMS are the similar types of information, so the same feature is used, dimensions are the only difference. To extract external stimuli can affect the mood as a social feature, we represent social data as a histogram by the number of message and length of messages. How frequent and long the participant receives the message is related to social mood like happy after sending SNS message. And which sender sends the message more frequent is also an important issue, histogram, and message frequency for all contact list are used as features. App data are also useful, app usage and category of app is also a feature of our application. Category information is from Google play store. App history is not event-driven data but periodically accessed data; app count data is extracted as the feature that how long the application is running.

Power related features, features from battery and screen, are also related to smartphone usage. Battery state is similar with other continuous changing value like memory, moments of change and histogram are used for features. Current battery state, charging or discharging and being connected with PC or AC charger, can also be features. Screen features are frequency and mean duration and vector expressed by on/off status. A frequency of screen on/off represents how frequent user uses the smartphone. Mean of duration shows how long user uses a smartphone at one time.

Sound feature is only intensity histogram by frequency. It is a result of consideration about privacy. As mentioned in recording service of the smartphone platform, finding and changing voice to text data using speech-to-text invade privacy severely. Also, the surrounding sound is collected to find a relation between location and noise. Therefore, an intensity of surrounding noise based on frequency is extracted as the sound feature. The sampling rate of recorded sound is 8kHz, so the maximum frequency of intensity histogram is 4kHz, half of sampling rate. To reduce a dimension of features, the bin size of a histogram is 10Hz.

As Gear data is also collected, features from Gear data is needed. All Gear data is continuous data and change of them is important, moments of change and histogram are used. AGM data from gear is processed by jigsaw feature. But as we discussed in the previous chapter, many data of gear are not continuous in all day. In particular situations like answering the survey, all data is collected continuously.

Location and activity that participant replies are used as current situation information. As estimating current location and activity using given data is not accurate, survey data is used instead. All answer are categorized to number from 0 to 14, a weighted sum of previous and current information is used as feature after expressing 15-dimensional vector by one-hot encoding. Weighted sum of a vector is due to historical information about location and activity.

Used feature is explained briefly on Figure 4.3 and Figure 4.4.

Data Type	Description	Dimension
Acc	mean (Gear/SmartPhone)	6
	standard deviation (Gear/SmartPhone)	6
	mean crossing rate (Gear/SmartPhone)	6
	Hightest value in freq domain (Gear/SmartPhone)	6
	sum of subband 1 in freq domain (Gear/SmartPhone)	6
	sum of subband 2 in freq domain (Gear/SmartPhone)	6
	sum of subband 3 in freq domain (Gear/SmartPhone)	6
	sum of subband 4 in freq domain (Gear/SmartPhone)	6
	sum of subband1 / sum of subband2 (Gear/SmartPhone)	6
	sum of subband3 / sum of subband4 (Gear/SmartPhone)	6
	(subband1+subband2)/(subband3+subband4) (Gear/SmartPhone)	6
	spectral entropy (Gear/SmartPhone)	6
Gyro	mean (Gear/SmartPhone)	6
	standard deviation (Gear/SmartPhone)	6
	mean crossing rate (Gear/SmartPhone)	6
	Hightest value in freq domain (Gear/SmartPhone)	6
	sum of subband 1 in freq domain (Gear/SmartPhone)	6
	sum of subband 2 in freq domain (Gear/SmartPhone)	6
	sum of subband 3 in freq domain (Gear/SmartPhone)	6
	sum of subband 4 in freq domain (Gear/SmartPhone)	6
	sum of subband1 / sum of subband2 (Gear/SmartPhone)	6
	sum of subband3 / sum of subband4 (Gear/SmartPhone)	6
	(subband1+subband2)/(subband3+subband4) (Gear/SmartPhone)	6
	spectral entropy (Gear/SmartPhone)	6
Mag	mean (Gear/SmartPhone)	6
	standard deviation (Gear/SmartPhone)	6
	mean crossing rate (Gear/SmartPhone)	6
	Hightest value in freq domain (Gear/SmartPhone)	6
	sum of subband 1 in freq domain (Gear/SmartPhone)	6
	sum of subband 2 in freq domain (Gear/SmartPhone)	6
	sum of subband 3 in freq domain (Gear/SmartPhone)	6
	sum of subband 4 in freq domain (Gear/SmartPhone)	6
	sum of subband1 / sum of subband2 (Gear/SmartPhone)	6
	sum of subband3 / sum of subband4 (Gear/SmartPhone)	6
	(subband1+subband2)/(subband3+subband4) (Gear/SmartPhone)	6
	spectral entropy (Gear/SmartPhone)	6
Battery (Smartphone)	1st, 2nd, 3rd moment of changes	3
	histogram by every minute (30 min window size)	30
	Battery state	3
Screen	Frequency on/off	1
	Mean duration on/off	2
	Histogram by every minute (5 min window size, on/off)	10

Figure 4.3: Feature Table

Data Type	Description	Dimension
SNS	Histogram by # of messages	5
SNS	Histogram by length of message	5
SNS	Last time SNS arrived(sec)	1
SNS	Histogram of all contact list	150
SNS	Message arrival frequency	150
SMS	Histogram by # of messages	5
SMS	Histogram by length of message	5
SMS	Last time SMS arrived(sec)	1
SMS	Histogram of all contact list	50
SMS	Message arrival frequency	50
SNS, SMS	Sum of arrived SNS and SMS	1
App	Histogram by # of launches	5
App	# of used app	1
App	most used app	1
App	Histogram of all app List	250
App	Histogram of all category List	30
Comm. App	Time used in Communication App	1
Phone	Histogram by # of messages	5
Phone	Last time phone call(sec)	1
Day	Current Day(0~6)	1
Hour	Current hour(0~23)	1
Age		1
Gender	1-Male, 2-Female	1
Noti	Last time noti arrived(sec)	1
Noti	# of arrived noti	1
data	Sum of data(tx, rx) in time window	2
cellLoc	ratio of changes	1
Memory	1st, 2nd, 3rd moment of changes	3
Memory	histogram by every minute (5 min window size)	5
UV	1st, 2nd, 3rd moment of changes	3
UV	histogram by every minute (5 min window size)	5
Light	1st, 2nd, 3rd moment of changes	3
Light	histogram by every minute (5 min window size)	5
Pressure	1st, 2nd, 3rd moment of changes	3
Pressure	histogram by every minute (5 min window size)	5
HR	1st, 2nd, 3rd moment of changes	3
HR	histogram by every minute (5 min window size)	5
Battery(Gear)	1st, 2nd, 3rd moment of changes	3
Battery(Gear)	histogram by every minute (5 min window size)	5
Location	Weighted sum of previous and current Location	15
Activity	Weighted sum of previous and current Activity	15
Sound	Intensity histogram by frequency (to 4kHz, 10 Hz bin size)	399

Figure 4.4: Feature Table

4.2.2 Inferring

Extracting feature from collected data, we can split these data into train data and test data randomly. Train data has the same size per each label, ”down” and ”up” and ”no change,” so training is not bias. But as collected data are bias to ”no change,” subtracting train data from whole collected data worsen bias of mood response to the survey. After separating train set from the original data set, test data set is more bias than the original data set. In figure 4.5 and 4.6, bias of test set can be shown. Those figures show that almost of test data have a label of ”no change” regardless of mood type. While data labeled ”no change” of activeness and happiness are above 90, the number of data which has ”up” and ”down” label less than 30 in activeness and happiness both. The ratio of ”up” or ”down” and ”no change” varies by from 3 times to 5 times. This means that scoring with accuracy only is not proper our applications. By classifying all data to ”no change,” accuracy is high but will be a useless result.

To infer mood using given data set, the random forest is used. After parameter tuning, 200 trees are trained with seven maximum depth for our application. Maximum depth prevents random forest model from meaningless node split like separating only one data from nodes. It can cause overfitting and extension of learning time, so maximum depth is required for better performance. At each node in training tree, a size of the randomly sampled feature is 293, which is 20% of feature dimension.

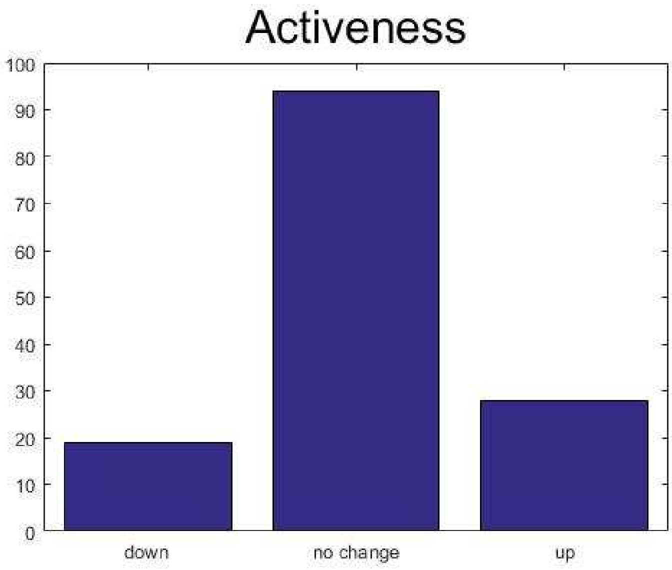


Figure 4.5: Distribution of Activeness (Test Set)

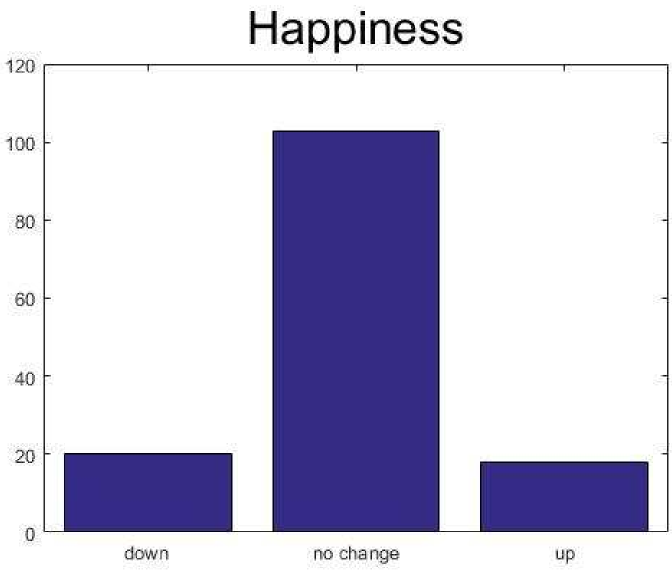


Figure 4.6: Distribution of Happiness (Test Set)

4.2.3 Result

Confusion matrices are shown in figure 4.7 and 4.8. In inferring activeness, our model does not classify when a user is more activeness or not. GT means ground truth and mod means moderate, ”no change”. In many test cases, our model confused ”up” and ”down” and ”no change” states. But when it comes to inferring happiness, our model classify relatively. At least 50% of data are correctly classified each state. Figure **??** is a table that measures score of our model. $Acc-$ is the true negative rate, measures how correctly model classifies negative response among negative samples. It can be calculated using following equation 4.1

$$Acc- = \frac{\text{True Nagative}}{\text{True Negative} + \text{False Positive}} \tag{4.1}$$

$Acc+$ is sensitivity or recall, measures how correctly model classify positive response among positive samples. It can be calculate using equation 4.2

$$Acc+ = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{4.2}$$

$Precision$ is proportion of correct labels among model classified as positive and be calculated by following equation 4.3

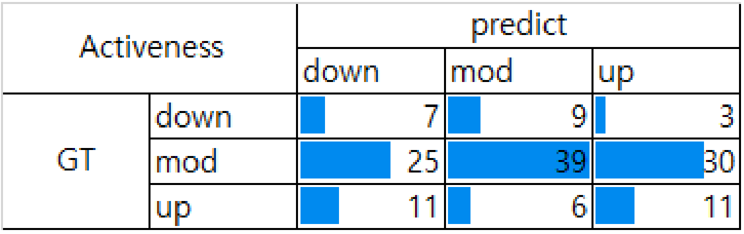
$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{4.3}$$

$F1\ score$ is harmonic mean of precision and sensitivity, $Acc-$, the equation is following.

$$F1\ score = \frac{1}{\frac{1}{Acc-} + \frac{1}{Prec}} \tag{4.4}$$

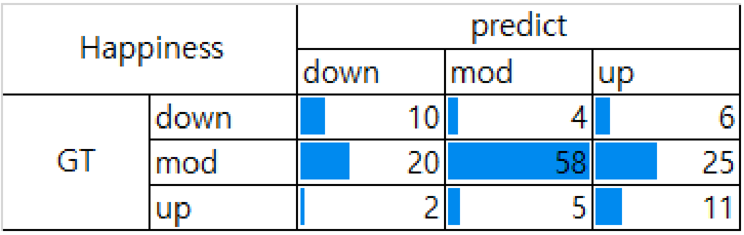
$G - mean$ is geometric mean of $Acc-$ and $Acc+$.

Sensitivities of models with both moods are above 70%, which means model detect negative state correctly. But comparing recall, our model detect user’s mood changes of happiness more correctly than activeness. Both moods show low value in precision; it is due to imbalanced data. ”no change” data is more than other data and ”no change” data lowers precision.



Activeness		predict		
		down	mod	up
GT	down	7	9	3
	mod	25	39	30
	up	11	6	11

Figure 4.7: Inferring Result : Activeness



Happiness		predict		
		down	mod	up
GT	down	10	4	6
	mod	20	58	25
	up	2	5	11

Figure 4.8: Inferring Result : Happiness

unit(%)	Activeness		Happiness	
	Up	Down	Up	Down
$Acc-$	70.79646	70.4918	47.79675	81.81818
$Acc+$	39.28571	36.87211	61.11111	50
$Precision$	25	16.27907	26.19048	31.25
$F1\ score$	30.55556	22.58065	35.66667	38.46154
$G-mean$	52.73793	50.96142	67.60852	36.96021

Table 4.1: Inferring Result : Score

Chapter 5

Conclusion

We developed life-logging platform that uses smart devices. As well as collecting many kinds of data as possible, the platform also considered user's privacy problem. Our smartphone platform was developed for Android devices for wide usage and designed to minimize the influence of Android operating system. And our proposed life-logging platform contains other smart wearable device, smartwatch. Although the wearable platform is device dependent, our platform shows smart watch device can be used as life-logging platform. As our platform is designed considering psychological experiments through a survey, simple survey application is added for our application of platform. Using survey application and our platform, we collected actual life-logging data during two weeks. And we proposed mood inference application as an example of our platform using collected data. For mood inference, a high dimensional feature was extracted and trained with random forest. Results were not excellent but showed that data could infer mood a little.

Bibliography

[1] S. Hodges, E. Berry, and K. Wood. Sensecam: A wearable camera that stimulates and rehabilitates autobiographical memory. Memory, 19(7):685–696, 2011

[2] Blum, Mark, Alex Pentland, and Gerhard Troster. ”Insense: Interest-based life logging.” IEEE MultiMedia 13.4 (2006): 40-48.

[3] LiKamWa, Robert, et al. ”MoodScope: building a mood sensor from smartphone usage patterns.” Proceeding of the 11th annual international conference on Mobile systems, applications, and services. ACM, 2013.

[4] Breiman, Leo. ”Random forests.” Machine learning 45.1 (2001): 5-32.

[5] Russell, Staurt J., and Peter Norvig. ”Artificial Intelligence (A Modern Approach).” (2010): 698-707.

[6] Jayarajah, Kasthuri, et al. ”LiveLabs: Building In-Situ Mobile Sensing & Behavioural Experimentation TestBeds.” Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services. ACM, 2016.

[7] Pielot, Martin, et al. ”When attention is not scarce-detecting boredom from mobile phone usage.” Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2015.

[8] Frost, Mads, et al. ”Supporting disease insight through data analysis: refinements of the monarca self-assessment system.” Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. ACM, 2013.

[9] http://emotionsense.org/code.html

[10] http://getnarrative.com/

[11] https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html

[12] 2015년 모바일인터넷이용실태조사 요약보고서, 한국인터넷진흥원 인터넷통계정보검색시스템

[13] Lu, Hong, et al. ”The Jigsaw continuous sensing engine for mobile phone applications.” Proceedings of the 8th ACM conference on embedded networked sensor systems. ACM, 2010.

[14] Wang, Rui, et al. ”StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones.” Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2014.

[15] Hwang, Inhwan, and Songhwai Oh. ”A Non-visual Sensor Triggered Life Logging System Using Canonical Correlation Analysis.” Cyber-Physical Systems, Networks, and Applications (CPSNA), 2015 IEEE 3rd International Conference on. IEEE, 2015.

[16] Burns, William, et al. ”Design and evaluation of a smartphone based wearable life-logging and social interaction system.” Computer-Based Medical Systems (CBMS), 2014 IEEE 27th International Symposium on. IEEE, 2014.

[17] Rekimoto, Jun, Takashi Miyaki, and Takaaki Ishizawa. ”LifeTag: WiFi-based continuous location logging for life pattern analysis.” LoCA. Vol. 2007. 2007.

[18] Su, Xing, Hanghang Tong, and Ping Ji. ”Activity recognition with smartphone sensors.” Tsinghua Science and Technology 19.3 (2014): 235-249.

초 록

센서와 전력소비 기술의 발달로 웨어러블 기기와 스마트폰은 인간의 삶을 매우 편리하게 만들어주었다. 특히 웨어러블 기기와 스마트폰은 사용자의 몸에 밀착되어 있으므로 기기의 센서를 통해 측정되는 값들은 사용자의 생활상에서의 특정한 패턴을 인식하여 기록하기에 용이하다. 따라서 이러한 일상생활을 기록하는 라이프 로깅은 중요한 가치가 있고 이에 대한 연구도 많이 진행되었다.

이 논문에서 우리는 스마트폰이나 웨어러블 기기 어느 한쪽만의 데이터를 수집하는 플랫폼이 아닌 스마트폰과 웨어러블 기기를 모두 사용하여 데이터를 수집, 생활패턴을 인식하여 기록하는 플랫폼을 제안하였다. 이 플랫폼에서 우리는 최대한 많은 데이터를 수집할 수 있고 운영체제의 버전에 상관 없이 최대한 다양한 사람들이 이 플랫폼을 사용할 수 있으며 심리학 실험과 같이 다양한 분야에 적용 할 수 있도록 하였다. 한편, 데이터를 수집하는 과정에서 개인의 사생활을 최대한으로 보호하고자 하였다.

그리고 제작한 플랫폼이 실제 실험에 적용할 수 있음을 보이기 위하여 이 플랫폼을 사용하여 주어진 기간동안 데이터와 감정 상태를 수집하였고, 이를 통해 사용자의 감정의 변화를 추론할 수 있는 분류기를 학습하였다. 그 결과 감정의 변화를 어느정도 감지함을 보이는 성과를 얻었고 제안한 플랫폼이 유효함을 확인하였다.

주요어: 스마트폰, 웨어러블, 라이프 로깅, 감정추론

학번: 2014-22553