LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
Department "Institut für Informatik"
Lehr- und Forschungseinheit Medieninformatik
Prof. Dr. Heinrich Hußmann

Bachelor Thesis

Investigating how people hold their mobile devices measured by the perception of the front facing camera

Anita Christina Baier
anita.baier@campus.lmu.de

Bearbeitungszeitraum: 25. 10. 2016 bis 14. 03. 2017
Betreuer: Mohamed Khamis
Verantw. Hochschullehrer: Prof. Dr. Florian Alt

Zusammenfassung

Mobile Endgeräte sind aus unserem Alltag nicht mehr weg zu denken und mit ihnen werden einer großen Zahl der Bevölkerung auch immer bessere Frontkameras zugänglich. Mittlerweile fungieren diese als Werkzeug für eine zuverlässige Gesichtserkennung und zuverlässiges Eye-Tracking. Deshalb ist es wichtig zu verstehen, wie viel eine Frontkamera von einer Person vor ihr abbilden kann und wie das durch verschiedene Gegebenheiten beeinflusst wird. Für diese Untersuchung stellen wir die Android Forschungs App "HDYHYP" vor, welche unbemerkt Fotos von Smartphhone Nutzern während ihrer gewohnten Aktivitäten macht und Daten über die Umstände sammelt. Wir zeigen, dass zumeist alle Gesichtsmerkmale einer Person innerhalb dessen liegen, was die Kamera aufnehmen kann, es aber auch Einschränkungen bei verschiedenen Interaktionsformen oder Positionierungen des mobilen Endgerätes gibt.

Abstract

Mobile devices are becoming ever more ubiquitous in our everyday lives and with them, better and better built-in front facing cameras are available to a large amount of people. In the meantime, the cameras these serve as tool for reliable facial recognition or eye tracking. For this manner it is important to understand, what the front facing camera can capture of the person in front of it and how this is influenced by various conditions. To investigate that, we introduce the research application "HDYHYP" for Android, which secretly captures photos of smartphone users during their everyday activities and collects data of the circumstances. We show, that most of the time, all facial characteristics of a person lie within the perception of the camera, but also limitations exist depending on the interaction form and on the position of the mobile device.

Aufgabenstellung

Task description: Applications that rely on facial recognition, eye tracking, etc.. have recently became feasible on commodity mobile devices. This is largely due to the recent advances in processing power and front-facing cameras of mobile devices.

However, a persistent problem is that the angle covered by the front-facing camera is limited; the way users hold their smartphones does not guarantee that the user's face is always in its view. The way users hold their smartphones is influenced by many factors, which include the currently running app (e.g., texting vs authentication), the context (e.g. where the user is, what he is doing), etc..

In this project, we would like to investigate what factors influence the way users hold their phones in a way that does not show their faces.

Scope of the thesis: The goal of this bachelor thesis is to study the factors that influence the user's way of holding the phone.

Tasks:

- Survey of related work on how people hold their phones
- Building a mobile app that takes pictures from the front facing camera, and tracks the usage of the phone (e.g., which apps are running) as well as sensor data (e.g. accelerometer).
- Performing an in-the-wild user study
- Analyzing the data to better understand what influences the user's way of holding the phone.

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, 14. März 2017

. . . .

Contents

1 Introduction

In modern days, front facing cameras are integrated into almost every models of mobile devices, weather it is a smartphone or a tablet. The fact, that not only a large number of people already use smartphones, but it is also expected to rise constantly [29], together with the circumstance, that the quality of the built-in cameras more and more increase, make it almost inevitable to apply other functions to the front facing camera than just taking selfies (which has been often appreciated because of the smoothing effect of low quality). By now, different reliable face and eye detection algorithm exist, which makes the front facing cameras of mobile devices a cheap face detector, which is available to a large number of people.

So the front facing camera is more and more used to detect facial characteristics. However, on several works in this field, various limitations appear. Looking trough the lense of a camera on the user, one can imagine several situations: Once an eye is covered by rims of glasses, other times there is just a ceiling lamp visible and so on. Several problems have been reported so far, which mostly appeared during lab studies and which have different reasons.

In this work, we are interested in what the front facing camera of a mobile device really sees from the person in front of it and when and why it captures the face and its landmarks and when and why not. We assume, that people hold their devices in different ways, depending on different conditions which then influences the perception of the camera on the user. Therefore we conducted a study over two weeks, during which we secretly captured photos of 11 participants on their regular activities. Of course they knew, that pictures could be taken, but they did not know, when that happened. We developed the Android application “HDYHYP”, which takes the photos and collects a range of further information in order to learn more about their circumstances. Afterwards we evaluate the taken pictures with regard to which facial landmarks are visible on them.

The thesis is structured as follows: In chapter 2 we first give an overview on the current state of investigation, how users hold mobile phones and also on the current usages of front facing cameras in the literature as well as the revealed limitations. In chapter 3 we transfer these findings to what we think are the consequences for the perception of front facing cameras. Based on that, we then present the general concept of the research application and explain when and which data is collected. In the second part of chapter 3 we go into technical details of the application. In chapter 4 we talk about the conduction stage of the study and present their quantitative results. In chapter 5 we discuss about the results and what they mean in a qualitative manner. Finally in chapter 6, we conclude this thesis and give an outlook on conceivable future research issues based on this work.

2 Related work

This sections gives an overview about previous research, which is related to our work and served us as a motivation. First we talk about research papers, which address the way of holding a mobile phone. Then we look at the scope of front facing cameras in mobile devices.

2.1 How people hold mobile phones

Grasp recognition is not a new research field in HCI. Taylor et al. already 2008 equiped a multi-functional handheld device shaped as a bar of soap with "a 3-axis accelerometer and with 48 capacitive sensors" to investigate how users grasp different objects, including a mobile phone. Therefore the authors asked participants in a study to hold the prototype as it were a mobile phone in five predefined modes of holding: gamepad, PDA, phone, camera and remote [1]. Later, Wimmer et al. conducted a study with a similar prototype named HandSense. They assumpted "hold up, pull out, grasp right, grasp left, hold left, hold right" as different grasp states, but they found, that it is hard to distinguish them. That is because the way of holding a device differs a lot across the participants [2]. Khanh et al. recently stated "four main holding styles and thirty six sub-cases" to hold a smartphone while walking. This is also due to the fact, that mobile phones can be put into a pocket or moved freely around the body [3]. Apart from that, Löchtefeld et al. "propose[d] an algorithm that automatically detects whether the user is operating the device with one or two hands as well as with which hand the user holds the device" by using the sensor data of the accelerometer, the device orientation and touch points [6].

2.2 Uses of front facing cameras

In addition to just taking pictures, the front facing camera is used for several purposes: One application is to interact with a device by markers the front facing camera captures [4]. Ming et al. build a large virtual screen out of several mobile devices and used their integrated front facing cameras to calibrate the system. Furthermore they mentioned "many applications utilize the front facing camera (FFC) in a single device scenario, e.g. gesture/face recognition, augmented reality or photo/video applications" [5].

Another field is to detect emotions of people in front of the camera in order to investigate an implicit communication channel: The authors of SmileAtMe used the Google Mobile Vision API to detect, if a person is smiling or not, when a funny picture is shown to him [7]. Also Yanqing et al. used the front facing camera to draw conclusions on emotional responses to photos [9]. Vasiete et al. observed the opposite case by using the front-facing camera to see people in stressful situations. They reported, that the facial expressions information were not reliable enough to detect the stress level of people, but they combined this with the information of several sensors [8]. As the developers of BTC did not have a research question in mind, when they made an application, which captures pictures from the photographer with the front facing camera, while he is taking pictures with the rear camera of his smartphone, they still found, that it "was very suitable for capturing moods, atmospheres or affective responses" [10].

Another use of the front facing cameras of mobile devices is to design safty systems for the users. The authors of CarSafe inter alia used the front facing camera to monitor the head pose and the blinking rates of the eyes of a car driver. The system was tested with pre-recorded video clips, but not unter real driving scenarios [12]. The authors of EyeProtector used the input of the front facing camera to alert the user, when the distance between the users eyes and the display gets to short. You et al. found, that the distance depends on the displaying of different types of media content: users tend to get nearer to the dispaly, when they want to see more details, e.g. on a still picture in contrast to a video. They also hyphothesized, that the distance was primarily influenced by the posture of the user. While testing their system, they observed, that the light conditions and long hair or bangs sometimes affected the detection of the users eyes. Furthermore they came up with situations, where the hole face could not be captured, although the distance was as intended [12]. A similar health risk detecting system was designed with Smart pose, a system to lower risk of neck tensions. It used the orientation sensor, the accelerometer and the detection of the eyes on pictures captured by the front facing camera for "identifying incorrect postures of smartphone users". The authors determined a tilt angle of the screen from 56.12° to 75.34° as an indicator for an abnormal neck position [13].

Further works, which rely on the front facing camera for several gaze detection algorithms report similar findings and problems: Like You et al. also Dostal et al. spotted, that the faces of users were often not fully visible due to the distance of the face to the screen. They also trace that back to the "narrow field of view of the camera" [14]. This is supported by Hohlfeld et al., who determined 20 cm as the best distance for accurate gaze detection. They also mentioned, that the accuray depends on the resolution of the camera [15]. Dostal et al. furthermore report about limitations of the gaze tracking by long hair and glasses (rims and reflections) [14]. In a lab study with controlled light conditions, Hohlenfeld et al. rated the influence of glasses rather marginal, but they found the influence of the ambient light situation, as well as the distance from the face to the screen very high [15]. Also Kunze et al. notice the dependancy on lighting conditions [17] and Wood et al. observed a decrease of accuracy of eye detection, when light sources reflect or draw deep shadows. What is more, Wood et al. and Khamis et al. mentioned, that the face respectively eye detection sometimes sufferd from the angle in which users holded a device, but they did not give further details about the specific tilt angles [16][18].

3 Concept

Weather as for inputting or for investigating purposes, the detection of the face and the eyes is a large application for mobile devices' front facing cameras. But also the detection of the mouth is important for some purposes. One question resulting of that is how often and when this is theoretically possible, because the camera actually captures these landmarks. What is also particularly noticeable, is that the major of the in section 2 mentioned applications were tested under controlled and constrained conditions in the lab.

Our approach is to get a basic idea of how people hold their phones in their everyday life. Therefore we chose the front facing camera as the measuring instrument. So our goal is to investigate the perception of the front facing camera on the user and to understand which conditions have an influence on what it can capture from the users face.

3.1 Aspects that influence how people hold their phones (hypotheses)

Based on the findings in 2 and the result of a brainstorming, we formulated the following hypotheses:

(H0) Media type: The perception of the camera depends on the media type of the displayed content type or applications, like text or media. Therefore the foreground application has to be known.

(H1) Obscured facial landmarks: The perception of the camera on facial landmarks as eyes and mouth can be affected by several objects:
a) If a user has long hair or bangs, the eyes or the mouth is significantly often covered by hair.
b) If a user wears glasses, his eyes are significantly often covered by the rims of them.
c) The eyes or the mouth of a user is significantly often covered by other objects then mentioned in a) and b).

(H2) Other persons: Other persons than the user can not or rarely captured by the front facing camera.

(H3) Sense of privacy: The perception of the camera depends on the environment of the user regarding his sense of privacy. Therefore it is important to know, if the user is located at a public, semi-public or private space.

(H4) Device position: The perception of the camera depends on the device position:
a) If the device is located on a surface, the user is often not visible or he is so far beyond the camera's perception, that only the upper half of his head is visible.
b) If the user holds the device in one hand, only the right or the left half of his face is significantly often visible.

(H5) Orientation: After a change in the device's orientation, the perception of the camera is often affected by fingers or hands.

(H6) Backwards tilt: If the device is tilted backwards, the angle between the camera and the user's face increases until the whole head but no facial landmarks are visible. Therefore the rotation of the device and the users head has to be known.

(H7) User posture: The perception of the camera depends on the user's posture: If the user's state is steady, the landmarks eyes and mouth are significantly more often visible then when the the user's state is moving. Therefore the velocity of the device has to be known.

(H8) Interaction: The perception of the camera regarding the visibility of all landmarks eyes and mouth depends on the interaction of the user with the device. Therefore several interactions has to be observed, in this case we observe switching on the screen (short and active interaction), receiving a notification (short and passive interaction), using an application (long and direct interaction).

3.2 Approach

The goal of the study is to identify users habits holding a mobile device during their everyday lives and determine general correlations between different conditions and how people are holding the device then as measured by what the front facing camera can see. Therefore we chose to conduct a field study, where the participants were investigated in their ordinary environment and during their usual activities. During two weeks pictures of the participants are taken, when conditions matching our hypotheses in section 3.1 occur.

For research purpose the application HDYHYP (acronym for "How do you hold your phone?") was developed, which captures photos trough the front facing camera while the user is interacting with his phone and collects additional data. To avoid impacts on the taken photos, we decided, that the participants should not know and not notice, when the photos are actually taken and when data is collected. No further hardware should be needed, only data provided by the device were processed. In addition experience sampling[1] is applied to some of the captured pictures. We decided, that the data should rather be available to us on a daily basis during the study instead of just after it's completion so it is possible to evaluate right from the beginning and to counter issues with the program of different nature as soon as possible.
It is important to us, that the participant's privacy is affected as minimal as it is possible, when an application can secretly capture photos of him. So we decided, that the paricipant has to send all collected data proactively to us and he has to have the chance to delete pictures of his choice. Furthermore, the application was once intended to run all the time, the user is able to manually start and stop the data collection now.

3.3 Research application

In the following sections we talk about the general functionality of the research application "HDY-HYP" and give an overview about the decision processes and the data collection process within the program.

3.3.1 General concept

From the background, the application HDYHYP listens for specific *triggers* in order to decide about and initiate a) *capturing sessions*, where one or more pictures are taken from the person in front of the device and additionally occasional b) *questionnaires* related to the latest photo are displayed. The photo collection only operates when the device's screen is turned on, so it is constrained to the time when a person actually interacts with the phone. The user does not receive specific feedback, when a photo is taken, but he knows, that the application is running and able to take pictures by a status bar notification. Every participant has to register within the application with an individual pseudonym to clearly assign the data consisting of photos and

[1] Experience sampling methods, where participants answer questions several times during one study, allow to learn more about individuals immediate environment, behaviors, etc. (cf. [20, p. 7]).

additional information to him during the evaluation.
While and after every taken photo, further data about the capturing conditions is gathered and then collectively stored into a database, where it is related to the taken photo. The pictures and the data sets are stored locally on the device and the participant has to transfer them at least once a day to us. Therefore he receives a reminder every evening. Then at the latest but also anytime he wants, the participant can review and delete taken photos in a simple gallery integrated to the application. To counter problems as early as possible, a simple option to get in contact with us and to share a log is also given.

All in all the essential functions of the research application include:

- It takes pictures while the participant is using his phone, but he does not have to notice it.
- While taking the photos, several data will be tracked and stored as a related snapshot.
- Up to six times a day the participant receives a questionnaire.
- At least once a day, the participant has the chance to review and delete photos before he has to submit them proactively for evaluation.

3.3.2 Capturing photos

The photos are captured by the front facing camera of the participants device, no additional hardware is needed. The participant does not know, when the photos are taken and does not receive feedback which indicates, that a photo is currently taken. The pictures are not taken arbitrary, there are rather two kind of triggers to capture pictures and collect data: when specific *events* (E) based on our hypotheses in section 3.1 take place or it is an appropriate time for a questionnaire, which occurs *independent*ly (I) and is defined more or less random within certain limits. When one of these triggers appears, a *capturing session* is started.

Event triggers: There are four event based conditions which lead to a capturing session:

- (S) The participant turns on the screen.
- (A) The participant launches an application.
- (N) The device receives a status bar notification.
- (O) The display orientation changes.

These events can take place any time during the day or night, but they are only detected, while the phone is in use.
Other interesting aspects as 'the participant is opening the keyboard', 'the participant captures a photo with the back camera' and 'the user starts to move faster' could be considered, but these are not integrated into the study.

Independent trigger: The user receives questionnaires, which are related to photos which are not linked to an event but independently taken. The capturing sessions for these photos are only triggered within 10 a.m. and 10 p.m., a period, which is in the time where people use their mobile devices the most frequently [23, p. 5 f.] and the longest in a row [22, p. 4715]. Moreover the user should receive a questionnaire up to six times a day, so we decided, that this trigger should take place on random times within intervals of two hours.

Capturing sessions: As most of the events lead to a continual state or interaction, we not only wanted to capture exactly the moment, when the trigger fires, but also take further images on an appropriate timescale, so we can portray course of the interaction. Once the program decided to take photos, the capturing session starts. This includes repeatedly 1) keeping trigger information, 2) taking one or several pictures and 3) initializing further data collection (cf. 3.3.3 - Collecting data).

The setting of a capturing session depends on the trigger, which caused it. A capturing session spans over one to up to four taken pictures, each followed by it's own data collection: The events *(S)* and *(O)* capture up to two pictures, while the *(A)* event captures up to four. The *(N)* will only cause one picture, because only the moment at which the notification is received is considered in this work as we do not track, if the participant taps on the notification. Also the non-event driven but independent, random scheduled pictures only take one picture. Nevertheless the terminology capturing session consists as defined, because there is still data collected linked to the pictures.
The first photo is always captured immediately after the event occurs. Then the temporal lag between two photos, measured in seconds (cf. 3.1), is based on previous findings about smartphone usage: Böhmer et al. presents average times, users spend on applications from starting to closing depending on its categories [23, p. 4 f.]. To cover all types, we gear towards the lowest average time, which is revealed as 36,37 seconds.

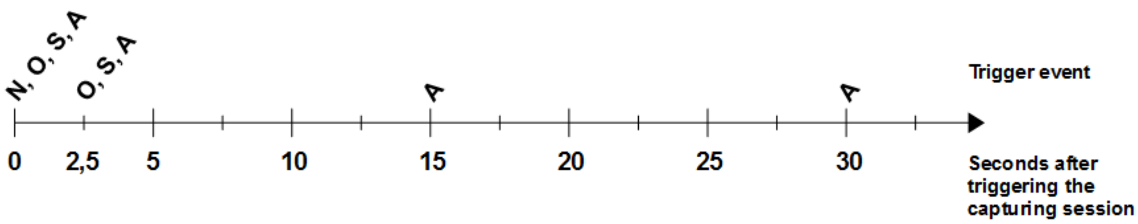

Figure 3.1: A timeline representing the moments and frequencies of shots in a capturing session, after a trigger occurred.

Van Berkel et al. finds an "application session (...) typically defined as a continuous period of time in which an application is both active and visible" [22, p. 4713]. That means means, that the same foreground application can generate several application sessions in a row. This can be the case, when the screen is switched on and off or the user launches the home screen in between. With our schedule we want to cover the moments, when the user first enters an interaction and also when he is really into it. For a general mapping of our decided time lags on all event driven capturing sessions, that we only consider new foreground applications. So we decided to abide by van Berkel's classification and concentrate only on the first element within sessions of the same application.

A session which plans multiple pictures to take can be interrupted. In this case, a possible currently being taken picture will be finished capturing as well as it's linked data collection, but no future pictures will be taken connected to this trigger. While the last action of the session is finishing, a new capturing session can already start. The interrupting conditions are a) *(I)* or another *(E)* comes in or b) the user turns off the screen. In the complete database, the lack of data sets on the timeline determined amount of shots indicates the interruption of a capturing session.

3.3.3 Collecting data

In addition to the pictures themselves, the application collects several information for two main purposes: to a) detect events which trigger a capturing session, to b) decide if it is necessary to trigger a capturing session and to c) store them as a photo related data set for further analysis.

There can not be made a clear distinction between these, because some information about the conditions in a) and b) are also processed and stored in c). In general there are information which are monitored all the time the application is running and information which are only requested, once a picture is taken. See table 3.1 for an overview of the moments and purpose of the collected data.
It may occur, that some of the aimed information cannot be detected or requested. If this concerns information, which are relevant for the decision processes, it may hold back triggering a whole capturing session and no other information will be stored. Otherwise the concerned information is marked as not available in the data set. For every photo in a capturing session one data set is stored persistently.

information collected		**information collected and stored into the data set**	
to detect events	**to decide about capturing session**	**while taking a picture**	**after taking a picture**
screen on/off current foreground apps current screen orientation time	resulted trigger of the previous step latest foreground apps latest status bar notification latest screen orientation latest independent trigger	trigger photo name time current foreground app face detection information location acceleration rotation current screen orientation battery information	time of survey submission device position holding hand user posture user position parallel activity

Table 3.1: Overview of information, broken down by the moment of collection and processing.

Another approach to cluster the collected information and stored data is to talk about them in their way of collecting. On the one hand there is data automatically collected by the program, which is provided by the device's sensors or the operating system or other services. On the other hand there is data, which is actively given by the participant itself. In the following, we will use this way to go into details, which data is collected and for which purpose.

Information collected automatically: Besides taking the photos, this is the other most considerable part of the research application. It is all about information, which are provided by the device in any manner and are automatically requested or collected by the program and also the program is responsible for which of these information have to be processed and stored. They include all types of information and data mentioned in a), b) and c) at the beginning of 3.3.3.

One of the most important information to know is the state of the screen, because the screen switched on is not only an indication for the *(S)* event, it also forms a key decision-making basis to determine all other events as a trigger and furthermore a switched off screen leads to the termination of a capturing session. To detect the events *(A)* and *(N)* it is necessary to log the current foreground app and the current appearing status bar notifications. To decide, if the detected event can be classified as a trigger (cf. 3.3.2 Event triggers), it is important to also log the latest foreground application and the latest status bar notification. The current foreground application, then identified as a new foreground application and a new status bar notification respectively, is then stored as persistent data. For the event screen *(O)* orientation, only changes are tracked. It is not necessary to make a great distinction between the current and the latest orientation, a change directly triggers a capturing session and the current orientation is stored.
The system time is necessary to keep in view to know the current period of the *(I)* trigger, also the time of the latest carried out one is tracked. Moreover the date and time, along with the

participant's individual pseudonym gives the photo it's file name to bring the pictures into context among each other afterwards. To see the dimension of time lag between taking the photo and the user actually filling in the survey, also the time of the survey submission is stored (also see below 3.3.3 Experience sampling).
Once an event or an independent alarm is identified and a capturing session is initialized, the trigger is stored. While a picture is captured, a face tracker tries to track the eyes and the mouth of the captured person, it tries to predict, if the eyes are and respectively which eye is closed or in to what extend opened, and it tries to estimate the pose angle of the face. But also further information about the conditions of the phone and indications about the environment at the time of the capturing of the photo are requested: The location of the user in GPS data, the rotation of the device in all directions, the acceleration of the device. Also the level of the battery and if it is currently charged or not, and the value of the screen brightness as well as the level of the ambient light is stored.
Apart from these specific data, general information about the sensors obscured to the mobile device are stored during the inital set up of the research application. To determine the position of the front face camera, the participant is asked about it during the first meeting with the study leader.

Experience sampling: In addition to the automatic data collection also experience sampling is included into the application in order to get familiar with the users behavior and immediate conditions outside the device. This is realized as a input prompt, which is presented to the participant several times a day. The questionnaires are related to a independently taken photo and are displayed immediately after taking it, regardless what else is displayed on the screen at that moment. These experience samples are a combination of the from Consolvo et al. identified characteristics: They are delivered randomly, but also scheduled within certain constraints, which influences the amount of the survey per day (cf. 3.3.2 Independent trigger), they provide written questions and they store written responses. [21, p. 25 ff.]
A questionnaire is target on the user's posture, position, parallel activities and the hand, in which he is holding the device, as well as the position of the phone. Therefore it consists of six questions, already providing reply suggestions, two of them with an additional free input field if the answer *other* respectively *yes* is chosen:

- The phone was...
  a) in my hands b) on a surface
- The holding hand was...
  a) dominant b) non dominant c) both d) none
- My posture was...
  a) walking b) standing c) sitting d) lying
- I were at/in...
  a) transit b) car c) home d) work/uni e) other
- While using my phone I was doing something else...
  a) no b) yes

We wanted to minimize the effort for the user in order to increase willingness to fill in the complete survey. Therefore we avoided many open-ended questions and preferred single choice questions to reach a time required to complete questionnaire under two minutes, suggested by Consolvo et al. [21]. Furthermore, all surveys are constructed equally. Nevertheless, the user also has the option to submit the survey with only partially answered questions. So in this methods of data collection it is also possible, that some elements of the data set are missing.

According to Consolvo et al., inappropriate situations are one of the the main reasons, why participants do not fill in surveys, when they are asked to [21, p. 28 f.]. To avoid this issue, the user has the option to dismiss the survey. For this case and also to cover the possibility, that the user might miss the survey, because it appears simultaneously to the user is switching off the screen, an entry point to the survey is provided trough a push notification. Within the survey, it is possible to see the related photo and also the time of capturing, so the user can give his answers as near to the related capture as possible.

3.4 Implementation

After we discussed the general compose of the research application, we want to go into technical details in this section.

3.4.1 Inner architecture

The research application HDYHYP is implemented on Android Devices with minimal API level of 21 (Lollipop). The core of the program are three services: the *ControllerService* (CS), the *CapturePictureService* (CPS) and the *DataCollectorService* (DCS), all of them operate in the background. On the one hand the *CS* controls the general procedure within the application and is responsible for the decisions, if pictures have to be taken at a certain time and data has to be collected. These tasks are performed by the *(CPS)* and the *(DCS)* on the other hand. While the *(CPS)* only gets started, when it is needed, the *(CS)* has to operate constantly while is is defined to take photos, as well as the *(DCS)* also runs constantly in the background to monitor sensor data while the screen is on, but it is stopped, when the screen is turned off.
Once the application is initialized via the *MainActivity* the *CS* is started to run constantly in the background. Therefore it is realized as a foreground service, which is maintained automatically by the system and is restarted by the BroadcastReceiver *EventBroadcastReceiver*, when it detects a system reboot. Also a *Storage* is instantiated to keep track of all persistent data and the BroadcastReceiver *RandomAlarmReceiver* is set up to manage alarms for the independent triggers. Then the following happens in chronological order: 1) The *(CS)* detects or was informed by several BroadcastReceiver objects and a *mNotificationListenerService* object (cf. 3.4.4) about an event or a independent trigger, 2) then the *(CS)* starts a capturing session, which means, it 2.1) directly starts the *(CPS)* and 2.2) sets several alarms for the BroadcastReceiver *EventAlarmReceiver*, when the future pictures of this session have to be taken. 3) When the *(CPS)* is started by the *(CS)* or by the *EventAlarmReceiver* it takes a photo with the front facing camera and then while it stores it via the *Storage*, it 3) tasks the *(DCS)* to store hatched and further data via the *Storage*. 4) If the capturing session was about a independent trigger, the *DCS* now creates a *SurveyActivity*, which prompts the questions to the participant and uses the *SurveyPictureFragment* to present the related picture.
On 22 p.m. every day an alarm is fired, so the BroadcastReceiver *ReminderAlarmReceiver* creates a push notification with the NotificationCompat.Builder, which provides an entry point into the *PictureReviewActivity*, which uses objects of the *PictureItem* and an *PictureReviewGridAdapter* to display all taken pictures in a simple gallery.

In this architecture, the android.content.BroadcastReceiver is extended multiple times, which could also be aggregated into one class. To avoid many case-switches in one receiver, they are split up into several receiver according to their assigned intent filter. The *EventAlarmReceiver*, the *RandomAlarmReceiver* and the *ReminderAlarmReceiver* are responsible for future tasks, set by the *CS* with the help of the android.app.AlarmManager. So the *EventAlarmReceiver* listens for alarms to take pictures of a capturing session and the *ReminderAlarmReceiver* receives a daily alarm at 22 p.m. to create a push notification for the data transfer. The

RandomAlarmReceiver receives the independent triggers, therefore the *CS* sets random alarms within the appropriate periods during the initialization process. The *EventBroadcastReceiver* on contrast cares for the system intents ACTION_SCREEN_ON and ACTION_SCREEN_OFF (for the screen state), ACTION_CONFIGURATION_CHANGED (for the orientation) and ACTION_BOOT_COMPLETED (to restart the *CS* after a system reboot).

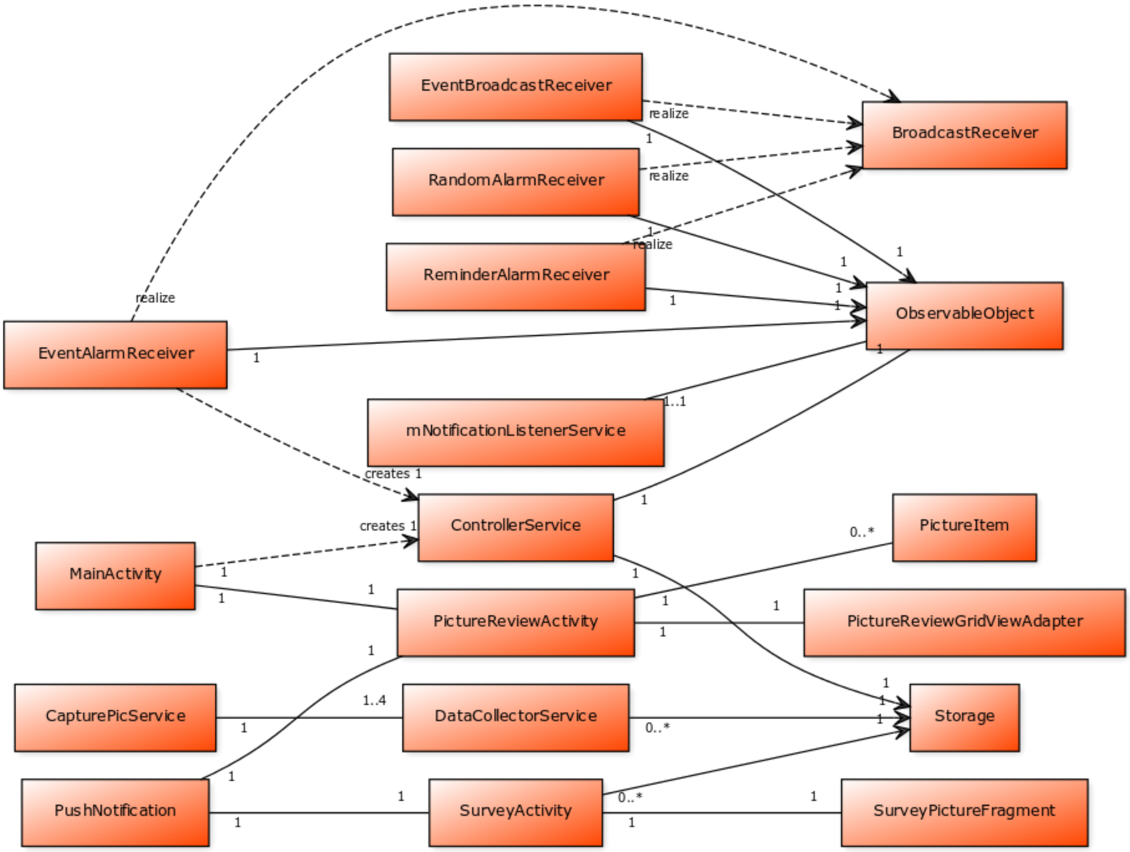

Figure 3.2: General implementation of the “HDYHYP” research application: Classes in an UML diagram.

3.4.2 User Interface

The user interface of the application consists of 3 Activities, 1 Fragment and Status Bar Notifications in different forms. Below we present the most relevant UI elements provided by the android.widget.* package, followed the screenshots in figure 3.3. The notifications just serve the purpose to give feedback about the running program (“Thanks for participating :)”) and to remind the participant of accomplishing tasks (“A questionnaire is waiting for you...” and “Please transfer data if you have WIFI”) and will not be further explained.

MainActivity: The *MainActivity* as an entry point and central control interface includes 7 Buttons. The upper 3 Buttons redirect to the necessary permission screens in the settings of the OS, the middle ones start and stop the *CS*, and the lower ones start intents to switch to the *PictureReviewActivity* and to open Android’s share dialog, where last one contains the study leader’s email address and a log file, which is created automatically via the Runtime.getRuntime().exec() method. Before the user starts the background service, he can choose his pseudonym via a Spinner.

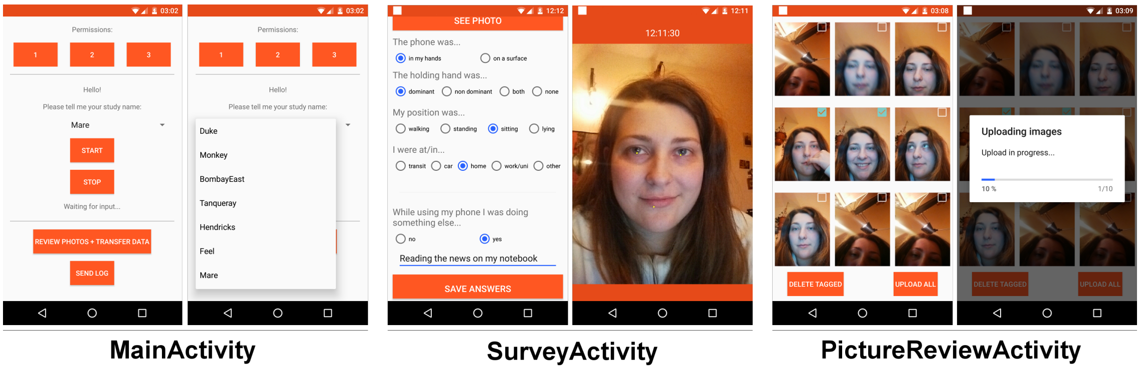

Figure 3.3: Screenshots of the UI elements of the "HDYHYP" research application, from left to right: MainActivity, SurveyActivity, PictureReviewActivity.

SurveyActivity: The *SurveyActivity* concists of six RadioGroups containing several RadioButtons for the answers to ensure single-choice. Two EditText fields offer the change to put in free text. One Button transitions the Activity to the attached *SurveyFragment*, which shows the related picture in an ImageView along with the point of capture in simple textual form. The other Button submits the answers, stores them via the *Storage* and destroys the *SurveyActivity*.

PictureReviewActivity: For review purpose, a simple gallery is prodived in the *PictureReviewActivity*, which loads all pictures in a custom ArrayAdapter named *PictureReviewGridViewAdapter*, which is embedded into a ScrollView. Therefore it uses instances of the class *ImageItem*, which represents a picture and a CheckBox. By checking them, the user can chose pictures for deleting and submit his selection via a Button, after that the *PictureReviewGridViewAdapter* is reloaded. Because all pictures are loaded at once, a circular ProgressBar is displayed to the participant during this operation. With another Button, the user uploads the remaining pictures and receives a ProgressDialog feedback on the uploading progress.

3.4.3 Data collection

To collect all data, different procedures and permission are necessary:

Camera: The access to the front facing camera only takes place in the *(CPS)*, therefore we use the API android.hardware.camera.

Face tracking: While taking the picture, a face detection algorithm runs on the photo. The first approach was the Camera.FaceDetectionListener, but as it was deprecated since API level 21 and also the android.gms.vision.face.FaceDetector provides more information, we use this API instead [26].

Sensors: Several sensors of the device are tracked: the gyroscope, the accelerometer and the ambient light sensor. They are the reason, why the *(DCS)* is running the whole time, because their listeners have to be registered before taking a value. To avoid significant delays between the capturing and the data collection, they are not registered at that moment but as soon as the *(DCS)* is created. Nevertheless, to save battery power, they are paused when the screen is off.

Foreground application and status bar notifications: The first attempt was to use the getRecentTasks(...) method of the android.app.ActivityManager, but this method is deprecated since API level 21 due to privacy reasons [27]. Now an instance of the android.app.usage.UsageStatsManager constantly requests the latest entry of a list of all current foreground applications in a while loop. These list includes applications as well as status bar notifications. Therefore, the *mNotificationListenerService*, an implementation of the android.service.notification.NotificationListenerService, gets notified by the system about incoming notifications in addition.

Location: The location is requested via the getLastKnownLocation() method of the android.location.LocationManager, which allows to access the system location services. The screen brightness is directly requested from the system settings and the battery level, as well as if it is currently charging or not is requested via the android.os.BatteryManager.

Further data: The orientation and the screen state is provided by the system and accessed via the *EventBroadcastReceiver*.

Permissions: For the data collection, the user has to give several access permissions: 1) usage access for the UsageStatsManager to detect foreground applications, 2) app permissions for the camera and location and 3) notifications access for the *mNotificationListenerService* to receive information about incoming notifications.

3.4.4 Decision processes

Always bringing together all necessary information to make decisions on this base and to put things in operation is the main responsibility of the *CS*. To fulfill it's function, it needs several information, which it can collect for it's own or with the help of several other classes, therefore the relevant objects are registered as observables. We talk about how they get information in 3.4.3 Data collection, in this section it is about processing these information. The *CS* has to manage - but not necessarily in this order:

1. detect triggers and initiate capturing sessions
2. interrupt capturing sessions
3. stop data monitoring of the *DCS* and foreground application detection of the *CS*
4. start data monitoring of the *DCS* and foreground application detection of the *CS*

2) and 3) is the case, if the screen is turned off. The *CS* directly causes the deregistration of the *DSC*'s listeners, which are constantly monitoring sensor values. Whereas they are immediately 4) registered again, as soon as the screen is switched on, even before 1) a capturing session is initialized because of this event. The detection of the foreground application in the *CS* works analog. Another basis of decision-making for 2) is the case, when 1) appears.
The 1) detection of a trigger can be divided into two types: the classification of an event as a trigger or the occurrence of an alarm for an independent picture with a survey. The decision about a survey goes hand in hand with the decision about the *R* trigger: After a trigger was detected, this information is forwarded do the *DCS*, which directly creates the survey then along collecting all other data. The classification of an event, however, depends on the particular event:
(S) + (O): The *EventBroadcastReceiver* only receives an intent, if the status of the screen changes, so no further distinction between old and new states is necessary, to treat *(S)* as a trigger. Same applies for the *(O)* event. The only necessary distinction is a case-switch within the *EventBroadcastReceiver*, to identify which intent for what event was received.
(A) + (N): Because of the *CPS* constantly queries the current foreground application in a while loop, a test, if it is a new foreground application or still the same running has to be done here.

Therefore the latest as new detected foreground application is cached in a global variable and is compared to the as local variable stored current foreground application in every iteration of the while loop. What's more, is that the detected foreground apps also contain status bar notifications, so also a comparison between the local current foreground app and the current value of the notification listener is done. Similar to the first distinction, the latest as new detected status bar notifications will be stored temporarily as a global variable and compared to incoming status bar notifications in the *mNotificationListenerService* to decide, if it is a notification from a different application, to avoid overload by e.g. multiple notifications of one conversation in a messenger.
(I): In our first implementation, the *(CS)* randomly sets daily repeating alarms for every 2 hour period between 10 a.m. and 10 p.m. For this approach, after receiving the alarm, the *RandomAlarmReceiver* has to check if the screen is on, if not, the alarm will simply be shifted in the AlarmManager for 5 minutes. Once the *RandomAlarmReceiver* receives an alarm, he checks, if there has already been taken a independent photo in the current period. This can be the case, if the *(CPS)* has been restarted and has set new independent alarms. The information is deposited in the applications shared preferences (android.content.SharedPreferences) as an array of integers. Also the original starting time is stored and then restored after taking the picture succeeded.
Later we realized a less on coincidence depending design: Every time, the *(CS)* gets the information, that the screen is turned on, it first checks on the same way, if a independent picture has already taken. If not, he cancels the current capturing session (which it just initiated because of the *(S)* event) and sets an alarm for the *RandomAlarmReceiver* in 5 seconds. If the alarm is not canceld in the meantime, due to the user turning off the screen, the BroadcastReceiver then receives the alarm, which causes the *(I)* trigger and sets a flag, that a picture hast already been taken in the concerning period.

3.4.5 Communication, storing and transferring data

The BroadcastReceiver and the *(CS)* communicate via the observer pattern, where the *(CS)* is the observer and the individual BroadcastReceiver are observables. The communication within the services and the activities is handled by extras (android.os.Bundle), which are added to the intents, which cause the concerning service to operate. With this method, data can be passed forward from the *CS*, to the *CPS* and then to the *DCS* to store them persistent. Also it is used to tell services, what they have to to. This mainly happens in the *DCS*, which is always addressed with the static method *startDataCollectionService(...)* in the *CS*, also the *CPS* uses this interface. In this method the intent gets an extra with a special command *DCSCOMMANDREGISTER*, *DCSCOMMANDUNREGISTER* or *DCSCOMMANDCOLLECT*. In the implemented android.app.Service method *onStartCommand()*, the *DCS* reads the command and decides, if he have to register/unregister it's listeners or if he has to collect and store data.

The taken photos as well as created log files are always stored in a public folder "HDYHYP" in the root folder of the intern storage. The user's pseudonym and information about the last independent alarm is stored in the shared preferences of the application. All other collected data, which is meant to be transferred to the study leader is stored in a SQLite database with one table for all automatically collected data and one table for the survey data (cf. 3.4), and which is also stored in the shared preferences. Therefore the *Storage* inherits from the abstract class android.database.sqlite.SQLiteOpenHelper, to access the database. Furthermore it provides methods to access all relevant data in the shared preferences.
As soon as the *(DCS)* and the *SurveyActivity* collect data, the set will be stored, even if the storing of the related photo failed or the participant deletes a photo.

To transfer data, our first approach was to use the Dropbox API and to load all files in a shared folder, respectively subfolder for every participant, of our Dropbox. After implementing

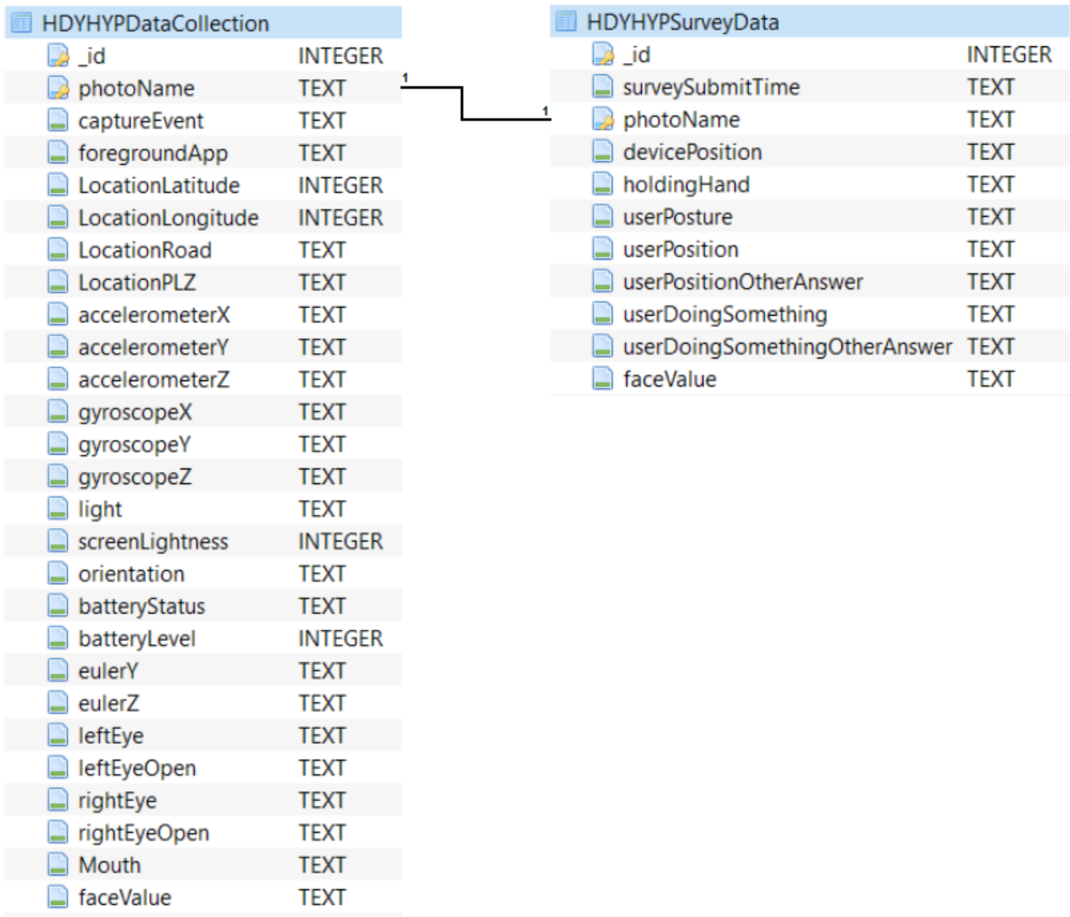

Figure 3.4: SQLite database scheme with two tables.

we noticed our misunderstanding of the API; it only uploads files into the dropbox, which the user authenticates. The user's could upload all data into their own Dropboxes and share the folder with us, but we assessed that way unsuitable, as we did not want to charge the participant's personal space. Also creating a common account was not an option, due to data privacy reasons.
Therefore we switched to a FTP Server, on which the application uploads the files via the external Jsch library. The logic is in the *PictureReviewActivity*, where the *PictureReviewGridViewAdapter* already initialized a ArrayList with *PictureItems* during displaying the pictures. The android.os.AsyncTask<Params, Progress, Result> *AsyncTaskConnectAndUploadToSFTP* uses this list to iterate through all files in the picture folder and uploads and delete them one by one. In addition the database is read from the shares preferences, names with the current date and uploaded, but not deleted.

3.5 Limitations

Lee et al. base their mobile posture-aware system "Smart pose" on the hypothesis of a steady correlation between a user's face and the user holding his device [13], but we rather think, that people hold their phones in different angles compared to the screen, e.g. significantly tilted back, so that either the eyes cannot be recorded by the front camera or the angle is very high, which could influence the accuracy of application possibilities, e.g. as eye tracking. So although we measure the rotation of the device, we can not directly derive the rotation of the human's head on the same axes and therefore calculate the angle between the screen and the head.
Also we do not measure the distance between the display and the person in front of it. We

could record the values of the proximity sensor, but as the in mobile devices installed ones sometimes only give extreme values of very near objects and very far objects[2], we decided to put the participant's device's battery life over this information to not miss relevant data due to dead batteries.

The answers from the experience sampling surveys like the posture of the users like sitting and lying, the position of the device and actions users perform parallel to the interaction with their devices seem very interesting to us. But while the information from the questionnaire about the location can be easily transferred to the pictures of the event driven capturing sessions because of collecting related GPS location data, these cannot be easily automatically identified trough the pictures. Myong-Woo Lee et al. use the accelerometer to identify several user postures and movements, but this requires additional hardware, as the sensor is mounted on the chest [24].

Moreover, the research application does not make a distinction between the actual participant and other people, who use his phone. This puts the participant in charge of caring for the personal privacy of people in his environment. As a result, this may slightly influence the results of the study, depending on the strategy of analyzing.

[2] Google Inc. *Sensor Event | Android Developers.* `https://developer.android.com/reference/android/hardware/SensorEvent.html` (accessed February 28, 2017) [25]

participant	visual aid	screen lock	camera position	output image ratio	hair length
1	no	fingerprint	top left	9:16	long
2	no	fingerprint	top right	9:16	short
3	no	pattern	top right	9:16	long
4	no	pattern	top right	9:16	long
5	no	pattern	top right	9:16	long
6	no	fingerprint + pin	top left	9:16	long
7	glasses (constant)	pin	top right	9:16	long
8	no	pattern	top left	9:16	long
9	contact lenses (constant) glasses (occasional)	pattern	top right	9:16	long
10	glasses (occasional)	pin	top right	9:16	short
11	glasses (occasional)	swipe	top left	3:4	short

Table 4.1: Overview of the study conditions of the 11 participants: visual aid, screen lock, front facing camera position from the users view, aspect ratio of portrait photos. Italic participants deleted pictures, which showed other people.

4 HDYHYP Study: perception of FFCs

The goal of the study was to get a basic insight on the perception of of mobile devices' front facing cameras and what it can capture from the person in front of it. In the following sections, we talk about the execution and the evaluation stage of the study.

4.1 Participants

For the field study 11 participants were acquired via mailing lists, social media and personal contact. They did not have to meet special requirements except of using a mobile device on a regular basis with an Android version higher then 5.0 (Lollipop) on which the study application could run for two weeks.

The participants age was between 19 and 34 years (26,36 on average), 5 were female, 6 male. 3 out of 11 participants were students during the study, the remaining 8 practiced varied professions but mostly office jobs: 1 DRG-Coder, 1 Hotel Manageress, 1 Project Manager, 1 Primary School Teacher and 3 Software Developer. All of them are right-handers. 4 participants used visual aid: 1 person wore contact lenses and 1 person glasses throughout the whole day, 2 participants wore glasses occasionally during the day. All participants used smartphones during the study, to determine the position of the front facing cameras at their devices, they were asked about the model: 7 devices had their front facing camera on the upper right side (from the users view), the remaining 4 ones had them positioned on the upper left side. Also see table 4.1 for an overview of the participants data.

4.2 Procedure

For each participant a personal appointment was arranged for a half hour setup meeting. To all participants the goal and the procedure was explained first and they signed a declaration of consent. Together with the study leader, the participants then installed the research application HDYHYP on their personal devices and gave all necessary permissions for the application to run properly. They first started the service, to check the system on actually taking a picture and then they were introduced to the user interface and it's functionalities. Also they were familiarized with the surveys and its questions. They were instructed to answer the questionnaires as soon as possible and to choose the most suitable answers.

All participants were asked to only delete pictures, they did not want, that we could see. This included photos of strangers, but they were instructed not to eliminate redundant or from their perspective irrelevant photos. Also they were told to activate access on their GPS location as often as possible, but also to have a look on their battery life as well and if necessary turn off the access. Moreover, they were encouraged to get in contact with us, if any problems appear and they were prepared for possible upcoming updates by explaining how to handle and install them for their own.

During two weeks every participant then uploaded an average number of 174 photos on the FTP server. Due to the volume of data, they were suggested to transfer them only when they are connected to WIFI in the first place. After this time a second meeting was set for every subject to uninstall the application and to conduct a final semi structured interview about demographic data and their experiences during the study.

For their attendance at the study the participants were rewarded with a 15 Euro Amazon voucher or 1,5 MMI credit points.

4.3 Encountered problems

Already in the preparation phase of the study a problem appeared. As we first attempted to get 15 to 20 participants, we hat do admit, that it is not easy to find that amount of participants for such a controversial request as the allowance of taking photos secretly. Finally we acquired 11 participants, where 8 of them were people from the personal environment of the study leader with the approach to compensate the data amount of the remaining 4 to 9 participants with the assumption, that participants with such a relationship will delete fewer pictures. Retrospect we could not find a significant distinction between friends and strangers on the deleted amount of pictures. However, we took an advance of the situation and rose the diversity by hiring participants with different professional settings instead of mainly computer science students, which are mainly addressed by social media and the mailing lists.

During the study there were two points, where we had to accept a loss of data. First we noticed, that most participants did not receive as much surveys as they were supposed. It turned out, that the implemented algorithm depended to much on the condition, if the participant uses his device exactly at the time, when a photo for the survey was intended. We then adjusted the algorithm and reduced the random factor (cf. 3.4). The second case could be traced back to an mistake in the code. As a safety measure, which was meant to serve the program's stability, the *(CPS)* was blocked, while it is taking a photo. During the fix of the first survey problem, this measure was inadvertently expanded on the further processing of the service. As a result the application was not able to take photos, when another picture was taken recently.
Both cases affected a loss of data (experience sampling answers and photos), but they do not distort the information content. This could also appear due to independent conditions, e.g. the device is dead because of it's battery, or the user manually stops the data collection, both mentioned in the final interviews. Therefore we accepted that circumstance but fixed these issues within a day and sent the updates to the participants.

At two participants we noticed a decrease of transferred data. It turned out, that there was a non-visible error on displaying the pictures for reviewing, which lead to an incomplete file list for the transfer. This issue also could have been fixed by an update, but as this occurred towards the end of the study, the pictures should be transferred on the study leader's notebook via USB during the final meeting.

Another problem was, that most of the final meetings had to be canceled on a short call because of private circumstances of the study leader. Instead of the last appointment, the participants received uninstalling instructions in an email then. Also the planned semi structured interview had to be replaced by a questionnaire with free text questions.

Altogether, the daily data transfer and the close contact to the participants made it possible to quickly notice, identify and encounter issues.

4.4 Results

In the following sections we present the results of the study and test them on the in 3.1 formulated hypotheses. The significance level is defined as 0.05 for all hypotheses. As “often” we define in more then 20% of all possible cases, “rarely” as less then 5%.
It must be noted, that the hypotheses *(H0) Media type*, *(H3) Sense of privacy* and *(H6) Backwards tilt* were not tested due to deficient data: While we were able to track foreground applications, we could not determine the types of contents which were displayed in which moment, as the major part of the detected applications supported several media types (e.g. Facebook and Chrome as two of the most frequently used applications during the study). Although we tracked the participants’ locations and could map private spaces (home), semi public spaces (work) and public spaces (transit) from the experience samples to the data sets, we missed the data collection of the subjects’ security senses on these locations. Also as already mentioned in section 3.5, we did not measure the angle between the user’s face and the device’s display, therefore we decided not to analyze an influence of the device’s back tilts.

4.4.1 General findings

In total, the participants transferred 26740 pictures. For quantification they were sorted in human terms into several clusters depending on what is visible of the face. This covers a) how much of the participant’s face is visible (where the face is constrained by the skin) and b) how many of the facial landmarks eyes and mouth are fully visible. For the numbers of transferred pictures of each cluster see table 4.2. Due to the different sizes of the samples, our statistic analyzes base on their relative probabilities. In detail the the clusters are:

none: No face is visible.

partially 0e 0m: The face is cropped by the image boarders and the mouth and the eyes are cropped or hidden by the image boarders.

partially 0e 1m: The face is cropped by the image boarders and both eyes are cropped or hidden by the image boarders but the mouth is fully visible.

partially 1e 0m: The face is cropped by the image boarders and the mouth and one eye is cropped or hidden by the image boarders but one eye is fully visible.

partially 1e 1m: The face is cropped by the image boarders and one eye is cropped or hidden by the image boarders but one eye and the mouth is fully visible.

partially 2e 0m: The face is cropped by the image boarders and the mouth is cropped or hidden by the image boarders but both eyes are fully visible.

partially 2e 1m: The face is cropped by the image boarders but all facial landmarks are fully visible.

whole: The whole face and all facial landmarks are fully visible.

participant	number	n	p 0e 1m	p 1e 0m	p 1e 1m	p 2e 0m	p 2e 1m	p 0e 0m	w	w pl	∑
1	total	639	2	51	20	179	344	25	966	77	2303
	probability	27.75%	0.09%	2.21%	0.87%	7.77%	14.94%	1.09%	42.34%	2.95%	
2	total	158	2	33	13	9	39	28	597	27	906
	probability	17.44%	0.22%	3.64%	1.43%	0.01%	4.30%	3.20%	65.89%	2.87%	
3	total	295	1	294	175	44	246	117	257	13	1443
	probability	20.44%	0.07%	20.37%	12.13%	3.05%	17.05%	8.11%	17.81%	0.90%	
4	total	934	84	988	103	121	372	377	325	49	3433
	probability	27.21%	2.45%	28.78%	3.00%	3.25%	10.84%	10.98%	9.47%	1.43%	
5	total	778	2	326	76	313	492	147	458	130	2721
	probability	28.59%	0.07%	11.98%	2.79%	11.50%	18.08%	5.40%	16.83%	4.78%	
6	total	1379	5	82	42	70	215	162	1742	324	4021
	probability	34.29%	0.12%	2.04%	1.04%	1.74%	5.35%	4.03%	43.32%	8.06%	
7	total	369	8	394	75	87	123	116	56	20	1248
	probability	29.57%	0.64%	31.57%	6.01%	6.97%	9.86%	9.29%	4.49%	1.60%	
8	total	518	1	63	25	81	371	43	534	85	1721
	probability	30.10%	0.06%	3.66%	1.45%	4.71%	21.56%	2.50%	31.03%	4.94%	
9	total	408	0	230	78	87	212	114	222	15	1366
	probability	29.87%	0.00%	16.84%	5.71%	6.37%	15.52%	8.35%	16.25%	1.10%	
10	total	2192	7	335	139	91	886	269	2183	178	6280
	probability	34.90%	0.11%	5.33%	2.21%	1.45%	14.11%	4.28%	34.76%	2.83%	
11	total	60	0	102	21	437	331	51	189	38	1229
	probability	4.88%	0.00%	8.30%	1.71%	35.56%	26.93%	4.15%	15.38%	3.09%	
	total ∑	7725	112	2544	734	1520	3432	1448	7527	957	26740

Table 4.2: Total numbers and probability of pictures and their quota on classifications regarding visibility of the face. The first parameter is the visible part of the face (n= none, p = partially, w= whole), the second and third stand for the visible facial landmarks (e = eye(s), m = mouth, pl = partially landmarks).

whole partially landmarks: The full face is within the image boarders but parts or all of the eyes and the mouth are not fully visible.

Figure 4.1 shows, that the pictures, which show nothing (n, median 28.59) is the main cluster of all pictures. Then pictures which show the whole face and it's landmarks without any constraints (w, median 17.81) and which show all landmarks but not the whole face (p 2e 1m, median 14,94) follow. Pictures showing only one eye rank in the midfield (p 1e 0m, median 8.3), rounded off by pictures showing two eyes (p 2e 0m, median 4.71), showing parts of the face but no landmarks (p 0e 0m, median 4.28), showing the whole face but only partially or no landmarks (p 0e 0m, median 2.98, showing one eye and the mouth (p 1e 1m, median 2.21) and showing only the mouth (p 1e 1m, median 0.09).
Especially remarkable statistical outliners are on the clusters n (min, 4.88), p 0e 1m (max, 2.45), p 1e 1m (max, 12.13) and p 2e 0m (max, 35.56), where the first and the last mentioned outliners both are the submissions of the same participant. We assume the image ratio of 3:4 as the reason of the p 2e 0m outliner, because this participant was the only participant with this output format. The p0e 1m outliner can be traced back to the fact, that this participant went into holidays during the study and often wore sunglasses since that moment.

4.4.2 Obscured facial landmarks

The analyzes of H1a) and H1b) are tested on the pictures, on which landmarks could potentially have been visible, defined as the ratio of pictures without n and p 0e 0m (T1) compared to all submitted pictures. H1c) is tested on the ratio of all pictures, on which the participants are wearing glasses compared to all submitted pictures (T2). H1a), H1b) and H1c) are the ratio of pictures with the defined obscuration compared do all submitted pictures (cf. 4.2).

a) The median of the ratio of all pictures of users with long hair and potentially visible landmarks is 66.71 (σ 5.95). The median of the ratio of pictures, where hair obscured landmarks

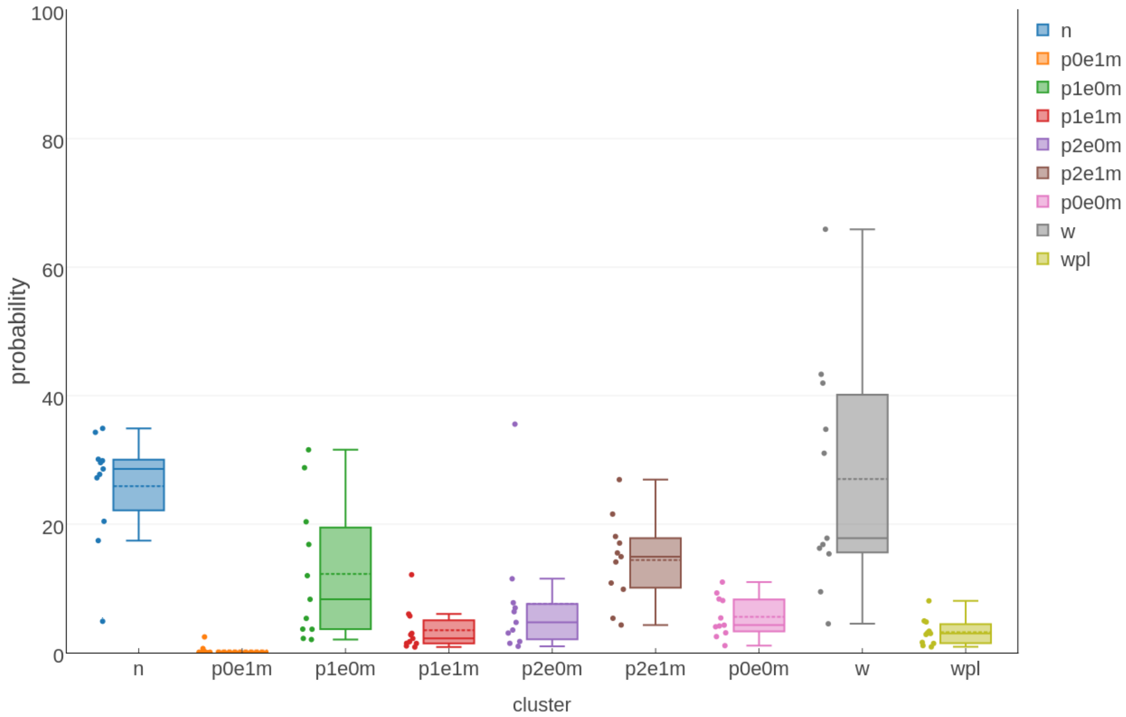

Figure 4.1: Statistics of all clusters regarding the relative probability numbers of submission.

is 0.14 (σ 0.21), which is less then 20% of 66.71. This means, that facial landmarks are not often obscured by hair (disproving of H1a).

b) The median of the ratio of all pictures on which users wear glasses is 28.6 (σ 24.81). The median of the ratio of pictures on which the rims obscured at least one eye is 2.12 (σ 0.51), which is less then 20% of 28.6. This means, that eyes are not often covered by rims of glasses, when the user is wearing them (disproving of H1b).

c) The median of the ratio of all pictures potentially visible landmarks is 66.01 (σ 9.48). The median of the ratio of pictures on which various other objects obscure facial landmarks is the rims obscured at least one eye is 0.39 (σ 2.87), which is less then 20% of 66.01. This means, that facial landmarks are not often covered by objects (disproving of H1c).

4.4.3 Other persons

The median of all pictures, where other persons than the user himself can be recognized is 0.64 (σ 0.33). This is less then 5% of the median 61.73 of all pictures where strangers could have been visible except the photos from participants, which deleted pictures of strangers (σ 12.33). Therefore H2 is approved.

4.4.4 Device position

a) The median of the ratio of all pictures which were located on a surface is 0.38 (σ 0.19). The median of all none, p 0e 0m, p 1e 01 and p 2e 0m pictures which were captured, while the phone was located on a surface is 0.36 (σ 0.22). This is greater then 20% of 0.38, which approves H4a).

b) The median of the ratio of all pictures which were located in one hand is 0.59 (σ 0.41). The median of all w, wpl and p 2e 1m pictures which were captured, while the phone was located

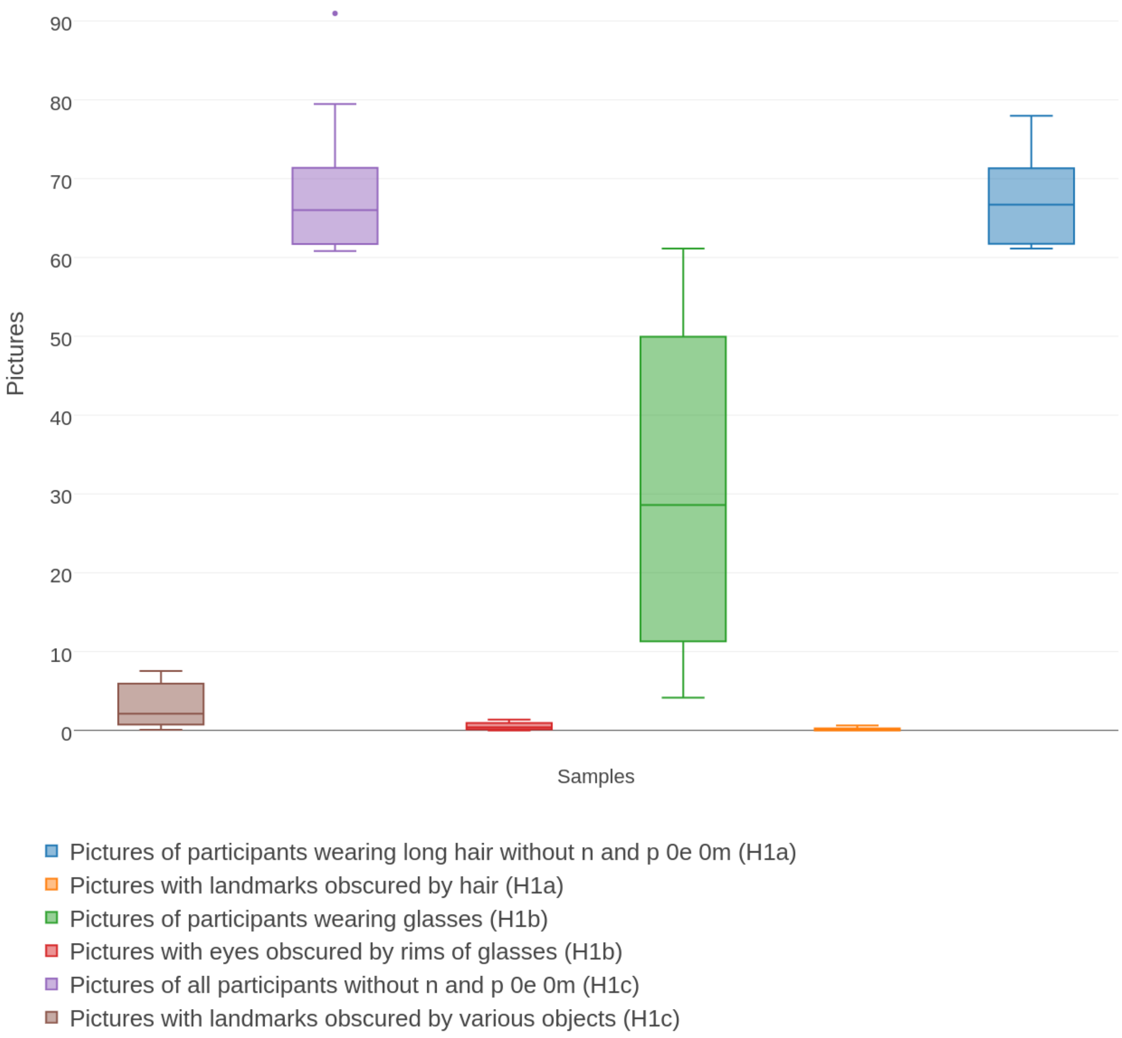

Figure 4.2: Comparison of obscuration (H1a - H1c).

in one hand is 0.07 (σ 0.1). This is smaller then 20% of 0.59, which disproves H4c).

4.4.5 Orientation

The median of the ratio of all pictures, which were taken during the orientation event (without n) is 1.80 (σ 1.68).The median of all pictures, which where taken during the orientation event and are affected by a finger is 0.08 (σ 0.9), which is less then 20% of 1.80, which disproves H5.

4.4.6 Interaction

To investigate the dependency of the visibility of the facial landmarks eyes and mouth on several interaction forms, we chose the following exemplary interactions: Receiving a notification for short and passive interactions, using an application for long and direct interactions and switching the screen on for short and active interactions (cf. 4.3).

Considering the influence of receiving a notification on the visibility of all facial landmarks, the result of an ANOVA test shows, that there is no significant difference between pictures with all facial landmarks visible and pictures without all facial landmarks visible ($F = 0.18$, $p = 0.67 > 0.05$).

Considering the influence of using an application on the visibility of all facial landmarks, the result of an ANOVA test shows, that there is a significant difference between pictures with all facial landmarks visible and pictures without all facial landmarks visible ($F = 2.60$, $p = 0.12 > 0.05$).

Considering the influence of switching on the screen on the visibility of all facial landmarks, the result of an ANOVA test shows, that there is no significant difference between pictures with all facial landmarks visible and pictures without all facial landmarks visible ($F = 0.12$, $p = 0.74 > 0.05$).

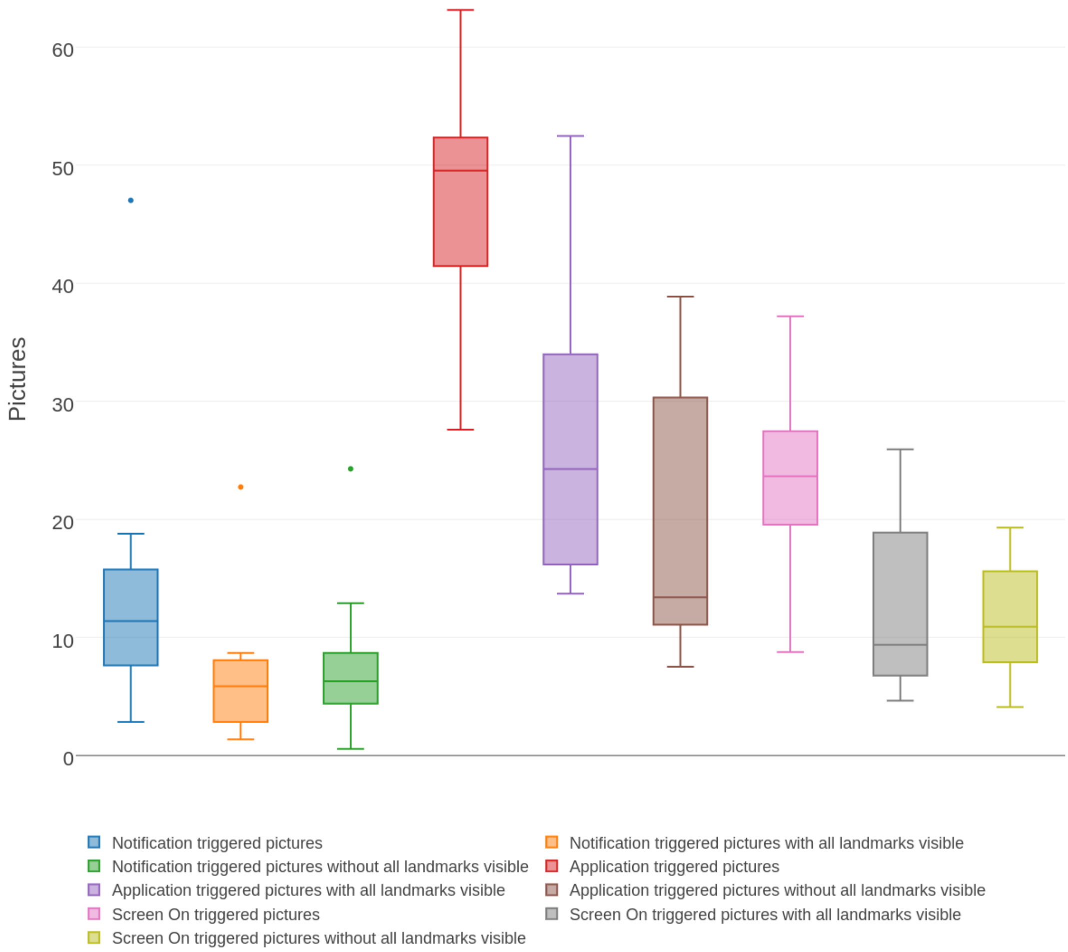

Figure 4.3: Comparison of different interactions and the visibility of all facial landmarks eyes and mouth.

5 Discussion

During the evaluation of the data, we noticed the following aspects:

Facial landmarks are generally not often obscured by objects or fingers, but on applications which need to track facial landmarks this must not be ignored. It is also mentionable, that mostly, obscuring objects did not cover a whole landmark (see 5.1 for an example). A lot of objects, which covered facial landmarks were found on every participants' pictures. These objects where not specific to single users but could be found across all participants' samples. Almost every time these objects obscured the mouth and were in decreasing number of frequency: 202x sunglasses, 423x finger or hand, 202x collar or scarf, 30x cigarettes, 30x telephone or headset, 17x drinking vessel, 13x pillow or blanket, 10x toothbrush, 6x other not identifiable objects, 4x newspaper, 4x pencil, 3x food. Sunglasses were not considered on the analyze of H1a), which turned out to be a mentionable factor on H1c). Is is mentionable, that the study took place during winter with very low outside temperatures. In the last days of the study, one participant went into a sunny holiday spot. Since then, the main covered landmarks were his eyes by sunglasses, while many other participants' most obscured facial landmark still was the mouth by a scarf.
Also eyes are not often covered by rims of glasses, pictures on which this occurred were photos, on which visible landmarks already have been covered by other conditions. These can be summed up to two cases: first when the face was already turned away, so that one eye was covered by the users head or nose itself. Second when the user was driving in his car and the phone was mounted on the center console so half of the face of the user was cut off the picture.

Other persons than the user were very rarely visible on the pictures. We observed, that in only 11 cases of 99 pictures with other people, the visible person was not a to the participant familiar person. Furthermore on only 2 photos of these 11, the strangers face was clearly visible (see 5.2 for an example). Therefore the front facing camera may not be useful to detect shoulder surfing attacks or to indicate, if the user is interacting with other people.

We showed, that the user is barely visible, when the device is positioned on a surface. Apart from that, one handed device use does not effect, that half of the user is cut off the picture. While reviewing the pictures, it seemed, that mostly all landmarks are visible while using the phone with one hand, but this observation requires further research to proof. Also the relation between one handed use and the position of the camera was not considered in this work. Furthermore these analyzes are based on the experience sampling questionnaires, which provided very small data and should be verified with a second investigation.

Although we could not prove, that fingers do not significantly often affect pictures after orientation changes, we noticed several outliners here. In contrast to H1, where we only considered fingers, which cover facial landmarks, all photos with visible fingers were included here. The outliners here coincide with the outliners on H1. We suggest, that these parameter strongly depends on individual habits. Furthermore, we did not take pictures into account, which were rather dark and on which nothing was visible. We estimate, that these photos could contain a high dark figure, because close to the camera located fingers could be the reason of dark pictures.

While there was no significant influence of receiving a notification and switching on the screen on the visibility of all facial landmarks, the influence of using an application is statistically significant. We suggest, this applies on long and active interactions, because the users concentrate on the interaction rather then performing them incidentally like switching on a screen.

In general, almost 1/3 of all pictures, did not contain any faces at all. Together with pho-

tos not containing any facial landmarks, the 1/3 is even exceeded. For applications, which are using the front facing camera in order to recognize a human face, this means a careful planning phase is strongly required. To keep being efficient it is important to define, what is needed to see and to strictly narrow down the possible situations, where this can be captured.
Nevertheless, there is much to read from every picture, even from pictures, which do not show faces. While reviewing all photos it was possible for us to reproduce, what the user is doing only by seeing the pictures and not comparing further data. It also stroke, that objects from the background can be easily recognized by a human, which allows conclusions to the location of the user without using traditional location information.
If a face were visible, than mostly one ore more landmarks were also available on the picture. If the whole face was visible, then also all facial landmarks were visible most of the time. Special cases like a to much tilted head, so the chin or the nose is covering facial landmarks accounted for the minor part of our pictures. Sometimes also the ambient light mattered: on the so called blue blue hour, it was difficult to identify faces and facial landmarks. Same applies, when a backlight spot was very bright in a dark environment. Apart from that, the light from the screen often sufficed to light the face enough to make it possible to identify the face.

We assume, that participants did not change their regular usage behaviour because of the application in a manner, that adversely influenced the results. On a scale from 1 - 5 in the final questionnaire only 2 participants stated with a 4 or 5, that they changed their normal habits. Upon request they described, that they avoided to use their phones in special situations, so they did not have to delete the pictures later. This reduces the amount of total pictures, but should not distort the results.

Figure 5.1: Example photo where the mouth is not fully obscured by an object.

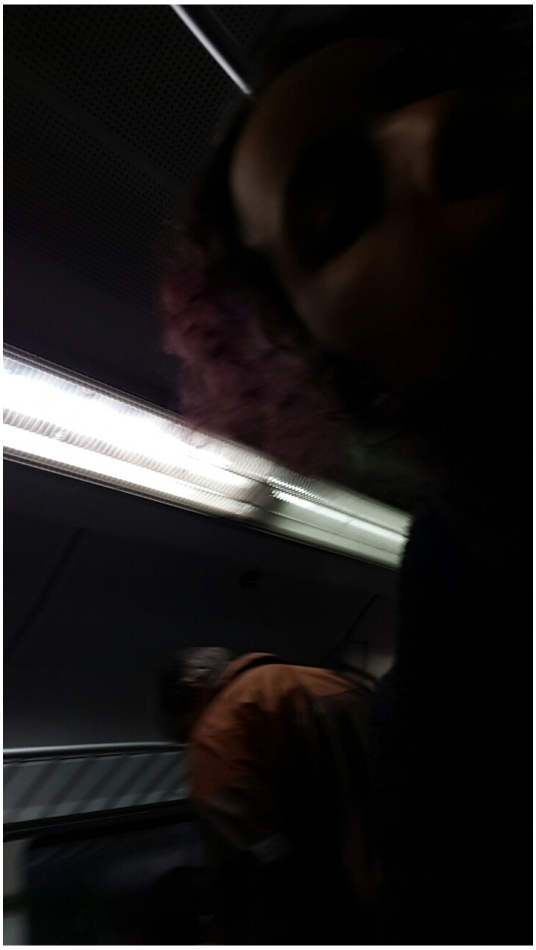

Figure 5.2: Example photo of a participant with a stranger in the background.

6 Conclusion and future work

In this work, very basic research was conducted, which give insights into how users are holding their phones. We saw, that in 1/3 of all pictures, no facial landmarks were visible, while in the remaining pictures at least one eye could be captured by the front facing camera. Also the the major part of them showed both eyes as well as the mouth. Facial landmarks rarely were obscured by objects but rather cropped by image boarders. We investigated, that the perception of a front facing camera on a user depends on the interaction form and on the position of the mobile device. But also many of our hypotheses where disproved, which shows the need of further research on this field. Therefore, many of our hypotheses should be further investigated with more parameter or more accurate data in the future. The research application “HDYHYP” collected a wide data set, which can be used for evaluation of further aspects, which would have gone beyond the scope of this work. But the code can also be extended and adjusted as desired to collect even more information.

A first step for further evaluation would be to develop a protractor to measure the angle between the screen and a user’s face. The current idea is to compare the size of the recorded forehead with a calibration image, when a certain threshold of the pitch/roll orientation value is measured by the gyroscope. This also requires an accurate distance meter or a fixed distance, which requires lab condition.

Another reasonable step is to further investigate the collected photos. Therefore further development from the sorting into separate clusters to tagging them with several keywords (e.g. “left eye visible”, “upper lip visible” etc.) would be valuable to apprehend more aspects of the pictures.

In our work we only scratched on the surface of the immediate environment of the user by taking experience samplings. It would be interesting to learn more about these conditions and influences on the perception of the front facing camera. For example a second person, which the user is talking to could significantly influence the users focus on the device. As other people are barely captured by the front facing camera, a detection via the microphone would be a conceivable approach.
Also the experience samplings only provided a small amount of information in this study. To get reliable data in the future, this part of the application should be extended by getting more samplings and collecting specific additional data, on which the samplings can be mapped on.

Many different objects were identified on the users photos. To learn more about human’s behaviour and habits, many algorithms which detect these objects can be developed. For example if an application detects cigarettes or drinking vessels, it can be applied on healthcare concerned intentions.

Although the rims of glasses did not often cover eyes, reflections on the glasses appeared on some photos. When the resolution of front facing cameras will improve in the future, these reflections could be used to draw conclusions on the environment of users. Nevertheless, the conditions of the cases in which reflections appear still have to be investigated.

An interesting question seems to us, if and in which cases it is more reasonable to adapt the users behaviour or to guide the user in a way, that lets him hold his device, as desired by a developer. Here, also instinctive change in behaviour while being observed can play a role, when using the front-facing camera. Several systems from passive indications to active gamification concepts are imaginable and are only waiting for someone to try them out.

Contents of the attached CD

- This bachelor thesis as a PDF-file and as LaTeXsource code
- Source code of the Android research application HDYHYP
- During the study collected databases and further data of the evaluation

References

[1] Taylor, Brandon T. and Bove, V Michael: The bar of soap: a grasp recognition system implemented in a multi-functional handheld device. In CHI '08 Extended Abstracts on Human Factors in Computing Systems, 2008, pp. 3459-3464.

[2] Wimmer, Raphael and Boring, Sebastian: HandSense: discriminating different ways of grasping and holding a tangible user interface. In Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, 2009, pp. 359-362.

[3] Khanh, Nguyen Huu and Lee, Kyung Ho and Chang, Geun Song and Lee, Seon-Woo: A Heading Estimation based on Smartphone Holding Styles. In Proceedings of HCI Korea, 2016, pp. 1-7.

[4] Garcia-Sanjuan, Fernando and Jaen, Javier and Fitzpatrick, Geraldine and Catala, Alejandro: MarkAirs: Around-Device Interactions with Tablets Using Fiducial Markers - An Evaluation of Precision Tasks. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, 16, 2016, pp. 2474-2481.

[5] Li, Ming and Kobbelt, Leif: Dynamic Tiling Display: Building an Interactive Display Surface Using Multiple Mobile Devices. In Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia, 12, 2012, p. 24.

[6] Löchtefeld, Markus and Schardt, Phillip and Krüger, Antonio and Boring, Sebastian: Detecting Users Handedness for Ergonomic Adaptation of Mobile User Interfaces. In Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia, 15, 2015, pp. 245-249.

[7] Brem, Jonathan and Hahn, Jürgen: Detecting Users Handedness for Ergonomic Adaptation of Mobile User Interfaces. In Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia, 15, 2015, pp. 245-249.

[8] Vasiete, Esther and Yeh, Tom: Multimodal Frustration Detection on Smartphones. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems, 15, 2015, pp. 1307-1312.

[9] Cui, Yanqing and Kangas, Jari and Holm, Jukka and Grassel, Guido: Front-camera Video Recordings As Emotion Responses to Mobile Photos Shared Within Close-knit Groups. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 13, 2013, pp. 981-990.

[10] Güldenpfennig, Florian and Reitberger, Wolfgang and Fitzpatrick, Geraldine: Of Unkempt Hair, Dirty Shirts and Smiling Faces: Capturing Behind the Mobile Camera. In Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design, 12, 2012, pp. 298-307.

[11] You, Chuang-Wen and Montes-de-Oca, Martha and Bao, Thomas J. and Lane, Nicholas D. and Lu, Hong and Cardone, Giuseppe and Torresani, Lorenzo and Campbell, Andrew T.: CarSafe: A Driver Safety App That Detects Dangerous Driving Behavior Using Dual-cameras on Smartphones. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, 12, 2012, pp. 671-672.

[12] Ho, Jimmy and Pointner, Reinhard and Shih, Huai-Chun and Lin, Yu-Chih and Chen, Hsuan-Yu and Tseng, Wei-Luan and Chen, Mike Y.: EyeProtector: Encouraging a Healthy Viewing Distance when Using Smartphones. In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, 15, 2015, pp. 77-85.

[13] Lee, Hosub and Choi, Young Sang and Lee, Sunjae and Shim, Eunsoo: Smart pose: mobile posture-aware system for lowering physical health risk of smartphone users. In CHI '13 Extended Abstracts on Human Factors in Computing Systems, 13, 2013, pp. 2257-2266.

[14] Dostal, Jakub and Kristensson, Per Ola and Quigley, Aaron: Estimating and Using Absolute and Relative Viewing Distance in Interactive Systems. In Pervasive Mob. Comput., 10, 2014, pp. 173-186.

[15] Hohlfeld, Oliver and Pomp, André and Link, Jó Ágila Bitsch and Guse, Dennis: On the Applicability of Computer Vision Based Gaze Tracking in Mobile Scenarios. In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, 15, 2015, pp. 427-434.

[16] Wood, Erroll and Bulling, Andreas: EyeTab: Model-based Gaze Estimation on Unmodified Tablet Computers. In Proceedings of the Symposium on Eye Tracking Research and Applications, 14, 2014, pp. 207-210.

[17] Kunze, Kai and Ishimaru, Shoya and Utsumi, Yuzuko and Kise, Koichi: My Reading Life: Towards Utilizing Eyetracking on Unmodified Tablets and Phones. In Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication, 13, 2013, p. 283-286.

[18] Khamis, Mohamed and Alt, Florian and Hassib, Mariam and von Zezschwitz, Emanuel and Hasholzner, Regina and Bulling, Andreas: GazeTouchPass: Multimodal Authentication Using Gaze and Touch on Mobile Devices. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, 16, 2016, pp. 2156-2164.

[19] Bolger, Niall and Laurenceau, Jean-Philippe: Intensive longitudinal methods: an introduction to diary and experience sampling research. The Guilford Press, 9, New York, 2013.

[20] Zirkel, Sabrina and Garcia, Julie A. and Murphy, Mary C.: Experience-Sampling Research Methods and Their Potential for Education Research. In Educational Researcher, 44, 2015, pp. 7-16.

[21] Consolvo, Sunny and Walker, Miriam: Using the Experience Sampling Method to Evaluate Ubicomp Applications. In IEEE Pervasive Computing, 2, 2003, pp. 24-31.

[22] van Berkel, Niels and Luo, Chu and Anagnostopoulos, Theodoros and Ferreira, Denzil and Goncalves, Jorge and Hosio, Simo and Kostakos, Vassilis: A Systematic Assessment of Smartphone Usage Gaps. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16), 5, 2016, pp. 4711-4721.

[23] Böhmer, Matthias and Hecht, Brent and Schöning, Johannes and Krüger, Antonio and Bauer, Gernot: Falling asleep with Angry Birds, Facebook and Kindle: a large scale study on mobile application usage. In Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11), 2011, pp. 47-56.

[24] Lee, Myong-Woo and Khan, Adil Mehmood and Kim, Tae-Seong: A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation. In Personal Ubiquitous Comput., 15, 2011, pp. 887-898.

Web-references

[25] SensorEvent | Google Developers. https://developer.android.com/reference/android/hardware/SensorEvent.html, accessed February 28, 2017.

[26] Face Detection Concepts Overview | Mobile Vision | Google Developers. https://developers.google.com/vision/face-detection-concepts, accessed February 28, 2017.

[27] ActivityManager | Google Developers. https://developer.android.com/reference/android/app/ActivityManager.html, accessed February 28, 2017.

[28] MotionSenors | Google Developers. https://developer.android.com/guide/topics/sensors/sensors_motion.html#sensors-raw-data, accessed March 08, 2017.

[29] Smartphone-Nutzer weltweit 2012-2020 | Statista. https://de.statista.com/statistik/daten/studie/309656/umfrage/prognose-zur-anzahl-der-smartphone-nutzer-weltweit/, accessed March 13, 2017.