The University of Hong Kong
COMP4801 Final Year Project

Lightweight Thermography Fever Detection System

Group 1 FYP21014
Final Report

Wong Chi Ming 3035574141

Supervisor: Dr. Chim, Tat Wing
Department of Computer Science

Date of submission: 18 April, 2022
Acknowledgements

I would like to extend my sincere thanks to my supervisor Dr. Chim, Tat Wing for the patience and support over the months. Without his valuable advice and guidance, we would not be able to proceed and finish the project.

I am also grateful to my CAES tutor Dr. Locky Law for providing helpful advice and practical suggestions on academic writing. I am able to improve my English writing skills and reach academic standards with his support.

Also, I would like to thank my teammate Rex, Wu Ka Hung for having me together on this journey. He has inspired me to think outside the box. I would not be able to achieve this result alone.

Sincerely,
Wong Chi Ming
Abstract

After the outbreak of Covid-19, the body temperature sensor and fever detection system played an important role in finding patients. This project developed an easy-to-use lightweight thermography fever detection system to be used in public to improve the user experience and effectiveness. We developed an iOS application that integrates AIZOO's FaceMaskDetection machine learning model and makes use of the FLIR ONE PRO thermography camera module to measure body temperature. The system is able to detect the body temperature of multiple people at the same time and whether the person is wearing a mask or not. We measured the performance and accuracy of our system and there is still room for improvement. We also explored the methodology and possibility of saving and retrieving temperature records of the users automatically by using machine learning and face identification. It is possible to save and retrieve temperature records associated with face images using Google’s FaceNet model.
# Table of Contents

Acknowledgements .................................................. 2
Abstract .................................................................. 3
Table of Contents ....................................................... 4
List of Figures .......................................................... 6
List of Abbreviations .................................................. 7

1. Introduction .......................................................... 8
   1.1 Background ....................................................... 8
      1.1.1 Impact of Covid-19 ........................................ 8
      1.1.2 Body Temperature Sensor .............................. 8
   1.2 Motivation ........................................................ 9
   1.3 Objective and Scope .......................................... 10
   1.4 Contribution .................................................... 11
   1.5 Report Outline ................................................ 12

2. Methodology .......................................................... 13
   2.1 Camera Module ............................................... 13
   2.2 Forehead Detection ........................................... 14
   2.3 Face recognition .............................................. 16
   2.4 iOS App .......................................................... 16

3. Experiments and Results .......................................... 18
   3.1 User Interface .................................................. 18
   3.2 Forehead Detection and Temperature Reading .......... 20
   3.3 Accuracy and Performance Evaluation .................. 22
      3.3.1 Face Detection against covers on the face ........... 22
      3.3.2 Minimum and Maximum Detection Distance ....... 23
      3.3.3 Multiple faces Detection ............................... 23
      3.3.4 Accuracy of Temperature Readings ................. 25

4. Limitations and Difficulties ....................................... 29
   4.1 Low Confidence of FaceNet for One to One Image Comparison 29
   4.2 Porting the app into the new SwiftUI .................... 30
   4.3 Camera Temperature Reading Calibration ............... 30
   4.4 Camera Battery and Charging ............................. 31
List of Figures

Figure 2.1: FLIR ONE PRO iOS 13
Figure 2.2: Parallax error of using two cameras 14
Figure 2.3: The use of face boundary to find out forehead 15
Figure 3.1: UI of the iOS application 18
Figure 3.2: UI color indicator of mask detection 19
Figure 3.3: UI fever alert 20
Figure 3.4: Undetectable area of the image for forehead detection 21
Figure 3.5: Face detection against covers on the face 22
Figure 3.6: Minimum and maximum detection distance 23
Figure 3.7: Simulation on multiple forehead detection 24
Figure 3.8: Experiment environment for measuring accuracy of the system 25
Figure 3.9: Temperature change of the camera in stable environment over time 26
Figure 3.10: The IR-805 infrared thermometer for comparison 27
Figure 3.11: Temperature change of the camera in stable environment over distance 27
Figure 4.1: Confidence of 1 to 1 face identification in different angles using FaceNet 30
Figure 4.2: The Belkin Lightning Audio + Charge Rockstar 31
List of Abbreviations

FLIR: Teledyne FLIR LLC
CS: Computer Science
UI: User Interface
AI: Artificial Intelligence
SDK: Software Development Kit
CNN: Convolutional Neural Network
NUC: Non-uniformity correction
1. Introduction

This section introduces the project background, our motivation to develop a new system to overcome the problems of existing body temperature sensors, followed by the project objective and the contribution of this project.

1.1 Background

This background section is to introduce the impact of Covid-19 to society, and provide information about existing body temperature readers and their shortcomings.

1.1.1 Impact of Covid-19

Since the outbreak of Covid-19 in Wuhan on 31 December 2019, it has become a global issue affecting more than a hundred countries [1]. Many measures such as lockdown and social distancing have been taken to control the outbreak of the virus. Fever is one the most common symptoms and the World Health Organization encourages people to seek medical care as soon as possible if they have a fever. Body temperature sensors become an important part of the measures. The Hong Kong Government requires restaurants and places of public entertainment to measure the body temperature of every customer [2]. Body temperature sensors can be seen everywhere now in Hong Kong and become an effective way to identify hidden patients.

1.1.2 Body Temperature Sensor

There are two approaches to measure the body temperature in general:

1. Infrared thermometer

   It detects the temperature by measuring the infrared radiation of the object. There are two types of infrared thermometer. The first one is the tympanic thermometer, which measures the inner ear temperature, and the other one is the non-contact infrared skin thermometer, which measures the hand or forehead temperature. Non-contact infrared skin thermometers are more common in public because of public hygiene.
It is fast in response and relatively cheap compared to thermography cameras. However, people have to place their hand or forehead near the thermometer in order to measure their body temperature and it may cause inconvenience. Also, distance between the object and the thermometer affects the temperature readings significantly [3].

2. Thermography camera

It measures the temperature of the area of the whole frame instead of one point. Multiple points and objects can be measured at the same time. It generates a heatmap and gives a visualized representation of the temperature of the target area. As a result, no user involvement is required for a fixed thermography camera. People can simply walk past the place and the camera will measure their body temperature automatically.

However, thermography cameras are relatively expensive and larger in size. It requires more computation power to process the heatmap data so a computer may be needed. It requires a specialist to set up and maintain.

1.2 Motivation

With the use of a thermography camera, it is possible to measure the body temperature of multiple people simultaneously and improve the overall efficiency of the process. However, the final year project conducted by the students last year focused on single-person face detection using OpenCV and did not mention the case of detecting multiple people at the same time [4]. We want to explore the possibility of reading the body temperature of multiple people simultaneously by using a similar thermography camera.
1.3 Objective and Scope

The objective of this project is to develop an automated fever detection system using a portable thermal camera attached to a smartphone. We developed a mobile application to read the heat maps from the camera using the SDK provided by the camera producer and show it on the screen. If the body temperature is higher than 37.5 °C, the app will alert the user by displaying a warning label and emitting sound.

The forehead of a person gives a more accurate body temperature. In order to locate the measuring spot and improve the accuracy of the system, we implemented machine learning to detect the forehead of the human face. The model also enables the application to identify multiple people at the same time to improve the efficiency of the system. It also detects whether the user is wearing a mask or not.

Apart from that, we tried to implement face identification and record the temperature of the users correspondingly. We want to identify the user base on the images. Temperature reading of the user history will be displayed on the screen if there is any. Records will not be stored permanently and will be deleted after a short period of time due to privacy concern. However, this feature is not implemented at the end as we encountered technical issues on recording the user's face.

The final deliverable of this project is an iOS application including the major features and functionalities specified as follow:

1. Build an iOS application with basic UI using Swift and UIKit
2. Read the heatmap data from the camera module and show it on the screen
3. Implement a machine learning model to detect multiple human foreheads
4. Detect whether the person is wearing a mask or not using the same model
5. Measuring the performance and accuracy of the system
1.4 Contribution

Under the pandemics, measuring body temperature is mandatory under many circumstances for example before entering a restaurant in Hong Kong. Fever detection system is more important than ever now. Many places will choose to use the ordinary infrared thermometer because of its low cost and the ease of use. Although thermal cameras have some advantages over normal infrared thermometers such as longer detection distance, less user involvement, and the ability to measure the body temperature of multiple users at the same time, thermal cameras are usually much more expensive and harder to set up. We want to lower the cost of using thermal cameras and make it easier to use for measuring body temperature.

Our portable fever detection system utilizes an iPhone and a USB-C thermal camera module. It is possible to handheld it like normal infrared thermometers. Since both the smartphone and the camera module have built-in battery, it is possible to use it without chagrin for a period of time. Even if the system is running out of battery, we could still use a power bank to extend the battery life. Apart from that, the little camera module is easy to set up. Users will only need to download the app and plug in the camera. It is also very lightweight and space saving. Usually thermal cameras need to connect to socket power and require a tripod to set it up. Our system is smaller in size and more flexible so that it can be placed in more different places and locations.

On the other hand, we explored the possibility of saving and retrieving temperature reading records of the users automatically based on image by using face identification and machine learning. It is possible to save the identity of the user using the images of his/her face. If the same user shows up in the camera after a period of time, it is possible to retrieve the past temperature records and show it on the screen. Combined with the portability of the system, one of the use cases would be to use the system in examination halls. We could place the system at the entrance of the examination room to measure the body temperature of every student to make sure that they do not have fever before entering the room. However, body temperature fluctuates. Body temperature may rise during traveling and commuting. When students enter the examination hall, some of the students may have a higher temperature due to traveling. With this system, it is possible to scan and measure the body temperature of the students after the examination starts. Helpers could handle the device and measure the temperature of the students.
again and not interrupt them. Helpers can also see the temperature record of the students a few minutes ago and see if their temperature dropped or not. Helpers are able to identify if the students are actually having fever.

1.5 Report Outline

This report first introduces the background of the project during pandemics, as well as a brief introduction to body temperature sensors and our motivation. Second part of the report states the objectives of the project, which is to develop an iOS mobile application to make use of the thermography camera to detect body temperature. The third section introduces the methodology and detailed implementation of the mobile application including the technologies and frameworks to be used. The fourth section includes the current status and the future plan of the project. The last section concludes the paper.
2. Methodology

This section discusses the methodology adopted to the project. It includes the engineering choice of the thermography camera hardware, the machine learning model used for forehead detection, the system architecture for face identification and saving temperature readings and finally the framework of iOS application.

2.1 Camera Module

The CS Department provided a FLIR ONE Pro Thermal Imaging Camera for iOS to us. It is being used in last year’s final year project so we got it for free. It has the best specifications among similar products in the price range. It has a thermal sensor with resolution of 160x120 and a normal visual camera with resolution of 1440x1080 sitting side by side (See Figure 2.1). Since the machine learning model we are using is trained using normal visual images instead of infra-red heat map images, we need the images captured by the normal view camera in order to locate the forehead. However, if we use the camera associated with the mobile phone, it will introduce a parallax error. Since there is a certain distance (usually around 10cm, varies from phone to phone) between the embedded camera on the phone and the thermal image camera on the FLIR ONE Pro, the viewing angles and view positions of the camera are different. If we use the image captured from the embedded camera to locate the forehead, then retrieve the temperature readings from the camera on the corresponding position, parallax error will occur (see Figure 2.2). Fortunately, we could make use of the secondary normal visual camera on FLIR ONE Pro to minimize this error.

![FLIR ONE Pro iOS camera module](image)

*Figure 2.1: FLIR ONE Pro iOS camera module. It has a normal visual camera on top and a thermography camera on bottom.*
2.2 Forehead Detection

We implemented a machine learning model to detect and locate users’ foreheads. The FLIR ONE Pro SDK is capable of calculating and retrieving basic information from the camera with additional functions like maximum and minimum temperature of the whole screen or given areas. In order to read the temperature of the foreheads, we have to locate the forehead areas by ourselves.

There are many trained models available online but most of them are trained using normal visual images instead of infra-read heat map images. As mentioned in section 2.1, we are able to retrieve normal view images from FLIR ONE Pro’s secondary camera so it would not be an issue and allows us to choose from a wide range of models. Another problem is that most of the models are trained using facial images without masks. This lowers the accuracy of detecting faces with a mask on. These old models are not the best fit for our applications as users are likely
to wear a mask while using this application. As a result, we chose the FaceMaskDetection model from AIZOOTech. They made a few different models for mainstream deep learning frameworks such as PyTorch, TensorFlow and Keras. However, it is not directly usable in Swift and on iOS applications. So we found a converted version called SwiftMaskDetection by keithito. It makes use of Apple's official CoreML utility to convert the TensorFlow model into CoreML model that could be used on Apple devices. CoreML models could utilize the Neural Engine or Neural Processing Unit on iPhone to accelerate the detection process and improve overall performance. The support for a variety of deep learning frameworks also makes the porting of the application to Android possible in the future.

The SwiftMaskDetection model enables us to detect the box boundary of faces and whether the people are wearing masks or not. Since we did not think of the ability to detect mask-on or not at the early stage of the project, we consider this as an additional feature. In order to detect the location of foreheads, we make use of the face boundaries detected by the model. We define the upper part (40%) of the face boundaries as the forehead (see Figure 2.3). We then find the maximum temperature inside the forehead boundaries as the body temperature results. This could reduce the interference of wearing glasses or hats as they could not fully cover the whole forehead boundaries.

![Figure 2.3](image.png)

*Figure 2.3:* Make use of the face boundary detected by the machine learning model and define the upper part as the forehead area. Retrieve maximum temperature inside the forehead area.
2.3 Face recognition

We wanted to record the body temperature of the users and then show the past temperature records of the current detected users automatically. In order to do this, we have to implement face recognition. We have to first identify the user faces from the normal visual camera, convert it into vectors and then store it associated with the body temperature reading into a local database. Next time when a face is being detected, we need to convert it into vector data and compare it against the stored vectors on the database. If we find a near result we could then retrieve the past temperature readings from the database and display it on the screen. The process of finding similar faces can be done by using an existing CNN model called Facenet. It was proposed by Google in 2015. It achieves an accuracy of higher than 90% on different dataset [5]. The model is implemented using TensorFlow so it is possible to convert the model into CoreML model as mentioned in section 2.2. However, if there are many faces stored there could be a performance issue.

FaceNet was proposed in 2015 and most people are not wearing a mask. The performance of face recognition with mask on remains to be tested. As it recognizes the face features and wearing a mask will cover up half of the face, it is reasonable to assume that the performance of the model will drop significantly when wearing a mask.

Apart from accuracy concerns, since we need to save the user faces on the phone in order to perform face recognition, users may worry about privacy issues. We could store all the data on temporary storage and delete them after the app is being terminated or after a short period of time to avoid this issue, however users may not be able to verify it and do not trust on us.

2.4 iOS App

We built an iOS application to integrate the FLIR ONE Pro SDK and SwiftMaskDetetion machine learning model. We developed a native iOS application using Swift and XCode. We designed and developed an application with basic UI including the function to switch between normal visual view and heat map, to display and highlight faces and foreheads boundaries, to
indicate whether the user is wearing a mask or not, and to display body temperature. If users’ body temperature is too high for a few consecutive readings, the body temperature label will turn red and the application will emit an alarm sound effect to warn the users.

We combined the mask detection model with the heatmap data to read forehead temperatures. The application logic flow can be described as follow:

1. Capture a frame of normal visual image and heat map image.
2. Detects faces and masks on the normal visual image by using the machine learning model.
3. Calculate the forehead areas.
4. Map the coordinates of the forehead areas to the heat map as they have different resolutions. Measure the maximum temperature inside forehead areas using the head map data.
5. Draw face boundaries on the display image. Use boundary border color to indicate whether the face is wearing a mask or not. Draw forehead boundaries and temperature labels as well.
6. Give warning by emitting an alarm sound and highlight the temperature label in red if someone's body temperature is higher than threshold (37.5 °C).
3. Experiments and Results

This section illustrates the development process and the outcome of our project. We successfully developed the application and are able to detect the body temperature of multiple users simultaneously.

3.1 User Interface

We designed and developed a simple UI for the application (See Figure 3.1). The top section shows the camera connection status including whether the camera is charging or not and the camera's battery level. The center area shows the heat map image or normal view image retrieved from the FLIR ONE Pro camera. It also shows the face and forehead boundaries detected, together with the temperature reading label. Below the image view, there is a IR-Photo view switch to let the user switch between Infra-read (heat map) view and normal visual view. There are also three buttons at the bottom to connect and disconnect the camera. The emulator button connects to the example frames provided by the FLIR ONE Pro SDK for debugging purposes.

![Image](image.png)

**Figure 3.1:** User Interface of the application running on iPhone 11. Image on the right shows the Infra-red image after the FLIR ONE Pro camera is connected.
There is also a slider to control the emissivity at the bottom for simple calibration. As mentioned in section 1.1.2, thermal camera measures the infra-red intensity emitted by the object surfaces. The emissivity refers to the relative intensity of infrared emitted by the object surfaces, which varies from 0 to 1. An emissivity of 0 means that the surface is perfectly smooth and emits less infrared radiation. On the other hand, An emissivity of 1 means that the surface of the object is very rough and emits higher infrared radiation relatively. The problem is that the thermal camera is not able to measure the emissivity of the object surfaces and leads to measuring error. We have to set the emissivity as a parameter of the camera to get a more accurate result. As a result, we could affect and calibrate the temperature readings by lowering or increasing the emissivity setting of the camera.

The machine learning model is able to detect whether the people inside the image are wearing a mask or not. We use the boundary color to indicate it. Face boundaries in blue means the person is not wearing a mask. Face boundaries in green means the person is wearing a mask correctly (See Figure 3.2).

![Figure 3.2](image)

**Figure 3.2**: The use of boundary color to indicate whether the user is wearing a mask or not. Blue refers to false and green refers to true.
If the body temperature of a user is too high, the temperature label will turn red as a warning (See Figure 3.3). The application will also emit an alarm sound (iOS SystemSoundID 1005) and vibrate to warn the user as well.

![Figure 3.3: The temperature label will turn red if the body temperature is higher than 37.5°C.](image)

### 3.2 Forehead Detection and Temperature Reading

In order to read the forehead temperature, we have to first locate the forehead area. The SwiftMaskDetection model needs to locate the whole face before it can decide whether the face is wearing a mask or not. In order words, not only does it detect whether the person is wearing a mask or not, it also detects the location and boundaries of the faces on the image. We could simply define the upper area as the forehead area with a height of 42% and width of 85% based on trial and error.
Next step is to map the visual image to the model test data. The model only accepts images of size 260 x 260 pixels. The height to width ratio of the infrared image and visual image from the camera is 1.33, which means that the model cannot read the whole image. We decided to crop the upper and lower part of the image and use the center part as input data for the model. This results in undetectable regions on two ends (see Figure 3.4). We cropped and resized the visual image into 260 x 260 pixels and passed it to the model. Then, we have to map the detection results back to the resolution and position of the infrared thermal image which is in 640 x 480 pixels. We make use of the FLIR ONE Pro SDK to retrieve the maximum temperature inside the rectangular forehead areas. Finally, we have to calculate and map the forehead areas to the normal visual image which is in 1440 x 1080 pixels and then draw the boundaries and labels. Users will be considered as having fever if the body temperature is higher 37.5 °C for consecutive 8 frames. This is to avoid false alarms if the readings fluctuate and jump high suddenly for one or few frames.

![Image](image.png)

**Figure 3.4:** The upper part and lower part of the image, colored in red, cannot be used for forehead detection and body temperature measuring as limited by the machine learning model.
3.3 Accuracy and Performance Evaluation

This section discusses the accuracy and performance of our system. It evaluates the performance against covers on the face, minimum and maximum detection distance, multiple users detection, and temperature reading accuracy.

3.3.1 Face Detection against covers on the face

Since we are using the maximum temperature inside the forehead area as the body temperature reading, it has a certain degree of tolerance against covers on the face such as glasses and caps. Even with a hat and a pair of glasses, as long as the forehead is being exposed, the application can still read the temperature out of it (See Figure 3.5). It depends on the viewing angle of the users. If the forehead is fully covered, the model may not be able to locate the face at all.

![Figure 3.5: The application is able to detect the forehead temperature even if the user is wearing a mask, glasses and a hat if part of the forehead is exposed.](image)
3.3.2 Minimum and Maximum Detection Distance

We tested the minimum and maximum face detection distance of the application. We placed the phone and camera on the desk and walked away from it step by step until it was not able to detect the face anymore. We then measured the distance between the body and camera using measuring tape. The minimum distance is around 20cm and the maximum is around 200cm (See Figure 3.6). The minimum distance is decided by whether the whole face can be fitted inside the center of the screen or not. The maximum distance is based on the minimum size of the faces appearing on the image.

![Image of face detection](image)

**Figure 3.6:** Shows the images of measuring forehead temperature successfully in a minimum distance of 20cm and a maximum distance of 200cm

3.3.3 Multiple faces Detection

The SwiftMaskDetection model is capable of detecting multiple faces at the same time which means we can detect the forehead temperature of multiple users at the same time as well. Since we are learning and working from home most of the time, we simply used the images of the “Standard Chartered Foundation FinTech Academy Plaque Unveiling Ceremony 2020” on the
press release of the CS department website to test our application. As long as the model is able to identify the faces on the images, it can locate the foreheads and read the temperature of these points. No matter if it's a real person or not. It will work on a computer screen as well. As mentioned in the previous section (section 3.3.2), the ability to detect faces is based on the relative image size of the faces. Even the distance between the camera and the computer screen does not reflect the actual distance between the camera and the real persons, it should not affect the ability of detecting multiple faces significantly. The model is able to recognize more than 5 faces at the same time (See Figure 3.7). These photos simulate multiple users from a certain distance. It is reading the temperature of my screen so the readings are low but it does not matter in this experiment.

![Figure 3.7](image)

**Figure 3.7:** Simulate the case of detecting multiple faces from a distance at the same time using photos displayed on a computer screen. The application is able to recognize more than 5 faces at the same time.
3.3.4 Accuracy of Temperature Readings

We wanted to calibrate the camera in order to give out a more accurate forehead temperature reading. However, the FLIR ONE Pro camera will perform Non-uniformity correction (NUC) automatically itself, interval varies from few seconds to tens of seconds. The camera itself will produce heat from the processing integrated circuit, battery, etc. under working conditions. The heat produced by itself will affect the accuracy of the reading as well. So NUC is to measure the radiation emitted by the optics of the thermal lens itself and adjust offset for each pixel [6]. The camera will freeze during the process so as to measure the radiation of the lens. The NUC will perform more frequently when the camera starts as the device temperature increases. NUC will perform less and less frequently as the temperature of the camera stabilizes over time.

We set up an experiment to measure the temperature change over time. We did the experiment in an air-conditioned room at 19 °C. We modified the app to measure and show the temperature reading of the center spot. We set up the camera to be 100 cm away from a white board and record the temperature change over 10 minutes (See Figure 3.8). Emissivity is set to 0.70 and remains unchanged. We record the value of the center spot for every 30 seconds (See Figure 3.9).

![Figure 3.8: Measuring the temperature change of the camera pointing to a white board 100 cm away over 10 minutes.](image)
Figure 3.9: Temperature change of the center spot against white board 100 cm away, in a stable air-conditioned room over 10 minutes.

The result clearly shows that the temperature increases over time and tends to stabilize after around 8 minutes since the camera booted. It means that the camera needs to warm up for a few minutes before we can calibrate it to get an accurate result. This is somehow inconvenient for the users as we have to wait before calibrating it and it is hard to know whether the temperature has stabilized or not.

Apart from the temperature change over time, we also measured the temperature change over distance. We set up an experiment under the same settings and environment (19°C air conditioned room, Emissivity of 0.7). We first warmed up the camera for a few minutes to let the temperature stabilize. We then measured the body temperature using a FLUS IR-805 Infrared Thermometer (See Figure 3.10). This thermometer claims an accuracy of ± 0.3°C and the reading is 36.3 °C. After that, one of us sat still on a chair and looked at the camera. Another one held the camera 20 cm away, walked away step by step and recorded the temperature readings for every 20cm, until we reached 200 cm away from each other (See Figure 3.11).
Figure 3.10: Measuring body temperature using FLUS IR-805 Infrared Thermometer as a comparison.

Figure 3.11: Temperature changes over distance, from 20 cm to 200 cm
The result clearly shows that there is an inverse relationship between the temperature and the distance between the object and the camera. In order to get a more accurate result, calibration against distance can be done in the future. However we have to get the distance info as well. The LiDAR depth camera embed in the newer iPhone could be a possible way to achieve it. For now, we have to calibrate and measure the body temperature at a certain distance to get the most accurate result. In this case, 40 cm gives the most accurate result. It can be calibrated simply by using the emissivity settings at the bottom of the UI after the temperature of the camera is stabilized.
4. Limitations and Difficulties

4.1 Low Confidence of FaceNet for One to One Image Comparison

We could use FaceNet to identify whether the person in the images is the same person or not. However, The accuracy of FaceNet on single to single image comparison is not good enough for practical use. Usually, we should train a classifier using multiple images of a single face to improve the accuracy. We should first capture multiple images of the same person, so that we could identify the face on the new image accurately. However, since we are capturing images from the camera instead of inputting image data manually, we have to first group and organize the images of the same faces automatically. We could not come up with an algorithm to process the consecutive frames and group the facial images together. So eventually we did not implement this face identification feature.

We download the FaceNet source code implemented by davidsandberg on github. We also downloaded the pre-trained model 20180402-114759. It is being trained using the popular VGGFace2 dataset for face recognition which contains 3.31 million images. It achieved an accuracy of 0.9965 in the Labeled Faces in the Wild (LFW) test [7]. We run the test on a Ubuntu virtual machine. I captured the images on my face in 5 different angles: facing forward, left, right, down and up. I captured 5 similar images with a mask as well. I use the image of facing forward as the base image, then I compare the base image against the image of facing 4 different directions (See Figure 4.1) using the compare.py tool. The output value is the confidence varies from 0 to 1. The result indicates that the accuracy does not drop significantly when I am wearing a mask. However, with an average confidence of 0.664 without mask and 0.652 with mask, it is not accurate enough for practical use. Training of classifiers using multiple images of the same person is required. We could not think of an algorithm to capture and group the faces from consecutive frames automatically.
Figure 4.1: Test result of using FaceNet one-to-one comparison to identify faces of different angles, with or without a mask.

4.2 Porting the app into the new SwiftUI

We tried to develop the iOS application using the new SwiftUI. SwiftUI achieves better performance, makes use of the expressive markup language to build the UI and enables hot reload for rapid development. However, the FLIR ONE Pro SDK is only available in UIKit delegate. The implementation of the ViewController delegate makes it hard to port. Even if we could wrap and port the SDK to SwiftUI, it will make it more difficult to read and write the code. Since we are mainly working with the SDK, we decided to stick with the UIKit.

4.3 Camera Temperature Reading Calibration

We wanted to calibrate the camera by setting its parameters. However, after the experiment in section 3.3.4, we understood that it is impossible to prefix all the parameters to achieve higher accuracy as the temperature of the camera device itself will affect the temperature readings overtime. The FLIR ONE Pro SDK also does not allow us to control any other calibration parameters aside from the emissivity. It only allows us to enable or disable the auto Non-uniformity correction (NUC) or change the interval of performing NUC. If we want to calibrate the camera, we have to first warm up the camera for a few minutes before performing
calibration. Also, the camera itself will also perform NUC every time it boots up, before connecting to the phone. So the initial temperature readings of the camera will be different every time it reboots and will eventually tend to be the same after it stabilizes. It makes it even more difficult to calibrate the camera. For now, the most accurate way to calibrate it is to warm up the camera for about 10 minutes and then calibrate it at a fixed distance by using another thermometer. Users have to measure their body temperature from the same distance as well for the best result.

4.4 Camera Battery and Charging

The FLIR ONE Pro camera consumes more electricity than what the lightning port could provide. So it has a built-in rechargeable battery and can be charged via a USB-C port at the bottom. However, due to the limitation of the lightning port, it cannot be charged and transfer data at the same time using the 1 to 2 lightning splitter (See Figure 4.2). The extra lightning port can be used for lightning earphones only. In order to charge the FLIR ONE Pro camera and the iPhone at the same time, we have to use wireless charging to charge the phone and wired charging over USB-C cable to charge the camera. The FLIR One Pro camera has a battery life of around 40 minutes. No charging device is needed for a short period of usage. Power banks with wireless charging feature can be used for extended periods of use and retains portability.

![Figure 4.2: The Belkin Lightning Audio + Charge Rockstar. Although it splits a single lightning port into two, it cannot be used to charge the phone and connect to the FLIR ONE Pro camera at the same time.](image-url)
5. Conclusion and Future Works

5.1 Conclusion

This paper discussed the advantages of using thermal cameras to measure body temperature. Unlike the normal infrared thermometer, the system developed by us is able to measure the forehead temperature of multiple persons at the same time. It does not require user involvement as well, the process can be done automatically when the user walks by. This improves the over efficiency of measuring the body temperature. The paper described the methodology of combining the use of a machine learning model and the thermal camera. We used the machine learning model to locate the forehead and are able to measure the forehead temperature of multiple users automatically. Whether the user is wearing a mask or not can also be detected using the model. We successfully developed a portable fever detection system using native iOS application, SwiftMaskDetection and FLIR ONE PRO thermal image camera.

However, there is room for improvement. The accuracy of the system can be improved. Due to the nature of thermography cameras, the temperature reading varies over time. The SDK of the camera also limited our ability to calibrate the camera. More expensive camera modules may solve this problem but the price of the system will increase significantly. It may not be as portable as well. The distance between the person and the camera will also affect the accuracy of the readings. Many existing systems require users to measure their body temperature in a fixed distance to ensure accuracy. If this problem can be solved cost-effectively, this kind of system will be much more convenient and widely adopted.

Apart from the difficulty of calibration, we also encounter problems when exploring the possibility to record and show past temperature records of the same user using face identification. We tested the accuracy of FaceNet proposed by Google which is well known for its accuracy. However, if we only have one single image as the base image (i.e. input for the classifier), the accuracy of is not preferable. We need to first input multiple images of the same face to the classifier in order to get a more accurate result. However, we could not come up with an algorithm to group face images of consecutive frames so we did not implement this feature.
5.2 Future Work

In order to calibrate the camera in advance, a different camera module with a more flexible SDK is needed. Retrieving raw data from the camera may also be feasible but it requires more processing and calibration work. We have to first investigate the change of the temperature readings caused by the heat produced by the camera itself and compensate for it. However we have to also take the environment temperature into consideration. As for the distance-temperature error, since the camera module usually has fixed view angle, one of the possible ways to measure the distance between the camera and the person is to measure the image size of the head. Smaller head image means further away. Another solution is to utilize the embedded depth camera on smartphones such as LiDAR on the iphone series.

As for the FaceNet, we could possibly implement an extra machine learning model to classify faces on consecutive frame images. We could then use them to train the FaceNet classifier and give out more accurate predictions. If we could identify the users correctly and effectively based on images, it is possible to save and retrieve past temperature records of the users automatically. Also, the performance of FaceNet on iOS remains to be tested as the iPhone may not have enough computational power to handle detection multiple times per second. It is possible to extend the interval of performing face identification to 1 time per 1 second or 2 second.
References


Appendix I

Temperature change of the center spot readings pointing to a white board 100 cm away, in a stable air-conditioned room at 19 °C, over 10 minutes. The emissivity is set to 0.70.

<table>
<thead>
<tr>
<th>Time (seconds)</th>
<th>Temperature (degree celsius)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>18.0</td>
</tr>
<tr>
<td>30</td>
<td>19.7</td>
</tr>
<tr>
<td>60</td>
<td>22.7</td>
</tr>
<tr>
<td>90</td>
<td>23.1</td>
</tr>
<tr>
<td>120</td>
<td>22.8</td>
</tr>
<tr>
<td>150</td>
<td>23.0</td>
</tr>
<tr>
<td>180</td>
<td>22.6</td>
</tr>
<tr>
<td>210</td>
<td>23.9</td>
</tr>
<tr>
<td>240</td>
<td>24.0</td>
</tr>
<tr>
<td>270</td>
<td>24.0</td>
</tr>
<tr>
<td>300</td>
<td>23.9</td>
</tr>
<tr>
<td>330</td>
<td>24.3</td>
</tr>
<tr>
<td>360</td>
<td>24.3</td>
</tr>
<tr>
<td>390</td>
<td>24.0</td>
</tr>
<tr>
<td>420</td>
<td>25.6</td>
</tr>
<tr>
<td>450</td>
<td>24.8</td>
</tr>
<tr>
<td>480</td>
<td>25.3</td>
</tr>
<tr>
<td>510</td>
<td>25.1</td>
</tr>
<tr>
<td>540</td>
<td>25.0</td>
</tr>
<tr>
<td>570</td>
<td>24.6</td>
</tr>
<tr>
<td>600</td>
<td>24.9</td>
</tr>
</tbody>
</table>
Appendix II

Temperature changes over distance, from 20 cm to 200 cm, in a stable air-conditioned room at 19 °C. The emissivity is set to 0.70.

<table>
<thead>
<tr>
<th>Distance (cm)</th>
<th>Temperature (degree celsius)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>38.8</td>
</tr>
<tr>
<td>40</td>
<td>36.3</td>
</tr>
<tr>
<td>60</td>
<td>36.0</td>
</tr>
<tr>
<td>80</td>
<td>35.6</td>
</tr>
<tr>
<td>100</td>
<td>35.2</td>
</tr>
<tr>
<td>120</td>
<td>36.0</td>
</tr>
<tr>
<td>140</td>
<td>35.5</td>
</tr>
<tr>
<td>160</td>
<td>35.0</td>
</tr>
<tr>
<td>180</td>
<td>34.3</td>
</tr>
<tr>
<td>200</td>
<td>34.3</td>
</tr>
</tbody>
</table>