

# Design Document

## Core Body Temperature Estimation to Detect Ebola Virus Disease

CS 461, Fall 2017, Group 34

Claude Maimon, Brian Lee Huang, and Bianca Beauchamp



### Abstract

The end goal of this project is a research paper as well as a prototype of the software. The paper should outline the problem that the project is trying to solve, the steps taken to solve the problem and how successful the implemented solution was. The prototype of the software as well as the information in the paper should allow the project to be continued by someone else. The whole process should be explained in detail allowing whoever wants to continue the project to continue without any problem. The main body of the research paper will be about the program that is developed to predict a person's core body temperature using a thermal image. The program will first be able to extract data from a thermal image. The data from the image will come from the top half, and then be narrowed down to the head of the person. It will then use the data to create a mathematical model that will predict core body temperature. A high accuracy rate is not strictly required as this is an exploratory project. The main goal is to determine if this method will be effective to detect a elevated core body temperature. If a high accuracy rate is achieved then there this method should continue to be perfected. If the outcome is a low accuracy rate then this method may not be best and a new method should be considered.

## CONTENTS

1	Introduction	3
2	Importing images and data	3
3	Images into right format	3
4	Pixel selection	3
5	Summary Statistics	4
6	Model	4
7	Model Testing	5
8	Evaluation	5
9	Production Model	5
10	User Interface	6
11	Conclusion	7
	References	7

## 1 INTRODUCTION

## 2 IMPORTING IMAGES AND DATA

The first step for image processing is importing the image from the camera. It's important that the image will stay in the right format to keep the pixel data. This means that the image needs to be in a format which maintains the temperature data. The FLIR Tools software will be used to achieve this.

FLIR Tools is a software for importing and analyzing images from FLIR cameras. The software can be used to import images to a personal computer, search the image library using various filters, store search criteria and manipulate images. The software is free to use and has a simple installation process. FLIR Tools is suitable for the FLIR A315 camera. Other than importing images the tool can be used to create PDF reports, add header logos to images and sort the image folder by specific variables.[1]

This software is the best option because it's free and it offers all the features needed for this process. Even though FLIR Tools+ offers more features, it is not necessary and this option will make the project cheaper.

Other than importing images, the core body temperature will need to be collected from many people. These temperatures are needed to compare to the estimated skin temperature from the camera. People's temperature will be collected and stored in a file. Every temperature will be paired with an image, so it is clear which temperature belongs to which person.

The thermal camera may not arrive in time for data collection. If this occurs, black and white images taken from a web camera will be used as mock thermal images. The results from these images won't be accurate but they will make a good place holder so that work can be done.

## 3 IMAGES INTO RIGHT FORMAT

In order to process the thermal images, they have to be in the right format. After importing the images from the camera to the computer, the program will transfer the images into a two-dimension array format. This format will allow for easy image processing. Every element in the two-dimensional array will hold a value of a pixel in the image. This structure will allow for easy processing of the pixel values.

The program will use OpenCV with Python to manipulate the image. OpenCV (Open Source Computer Vision Library) is a free open source computer vision software. It has interfaces for C++, C, Python, and Java and it supports various operating systems. OpenCV's library has more than 2500 algorithms which offer many features. Some of those features are facial recognition, gesture recognition, motion understanding, bio-medical analysis and more. The software is used all around the world, has a strong user community and offers technical support. OpenCV is a good option for manipulating the image because it is free and has a large user community. It has shown to have the best performance in comparison to other image processing libraries and there are many easy to understand tutorials available.[1]

## 4 PIXEL SELECTION

In order to get a good temperature estimation from the camera, we need to only select specific pixels. Our program will go through the pixels' values in the two-dimension array and select specific pixels. It will only select values that are higher than 98.8 and lower than 105. The pixel values might not be represented as degrees. We will find the pixel values that correspond to these degrees and will operate according to those values. This approach will work because we only

want to analyze the person's head in the picture. We don't want low and high pixels from the background to affect our results.

Since this is a research project we will try the previous method first, but we might choose to get the pixels values in a different way as well. The second way to do this is to cut the person's head from the picture before getting the pixels' values. This way, we would already eliminate the background so we don't need to worry about it when analyzing the image. This approach might be less effective since the cutting of the head is not perfect and might have some background in it.

## 5 SUMMARY STATISTICS

Once the pixels of skin have been isolated from the image, they need to be simplified into a single value that is a good representation of the entire set of data. This single value would represent the temperature of the subjects skin. This is important because the value be used in the model to relate to core body temperature.

The mean will be used for the summary statistics because it is the most accurate representation of all the data points and it is simple to find. In this case the mean used would be the arithmetic mean which is the average of the entire data set. The arithmetic mean is found by adding up all of the values in a data set and then dividing by the number of data points in the data set. The mean is the best used as a representation of the entire data set because it minimizes the sum of squared deviations from the typical value, meaning that it is a value that is the closest to all values in the set. This will be important because the value from the summary statistics will be what is used to relate the data from the camera to the measured value in the model.

## 6 MODEL

The condensed data needs to be related to the measured value in order to produce an equation that represents the relationship between skin temperature and core body temperature. The value produced by the summary statistics represents the skin temperature of a subject and the measured core body temperature of the same subject will be provided. These two pieces of data will be taken from many subjects and then used to create a model. The model will then be able to take just the summary statistics, which is the subjects estimated skin temperature, and produce the core body temperature.

The model must be able to represent the relationship as accurately as possible. This means that the type of model used will need to be determined by trial and error since there is no known relationship. The order in which the different models are tried should go from least complex to most complex.

The simplest model is a linear regression model. This model is used for modeling the relationship between one dependant variable and one or more explanatory variables. This type of model is typically used for predictions, forecasting and error reduction. The predictive model is made by finding the equation of the line of best fit for a set of dependant and explanatory variables. Then the model can be given an explanatory variable and predict the dependant variable.

A slightly more complex model is a polynomial regression model. This model is used for modeling the relationship between dependant and explanatory variables as an  $n$ th degree polynomial. The model is made by fitting a nonlinear relationship between explanatory variables and the corresponding conditional mean of dependant variables. Polynomial regression models are used for nonlinear phenomena.

A even more complex model is a ridge regression model. This is a type of regression model that is used when there are explanatory variables that have a high correlation. It is very similar to the linear regression but it is more complex since in order to account for the explanatory variables with a high correlation it uses the prediction errors.

The linear regression will bet the best model to try and test first because it is the simplest of all the models. If the linear regression does not produce a high enough accuracy during testing, the next model tried and tested will be the polynomial regression since it is slightly more complex. If the polynomial regression is not accurate enough, then the ridge regression will be tried and tested. If all three do not produce the accuracy that is desired then other models besides these three may need to be considered or it may not be possible to produce an accurate relationship between skin and core body temperature.

## **7 MODEL TESTING**

Once the model has trained on the set of data that has both the subjects estimated skin temperature and their measured core temperature, it needs to be tested on a new set of data to determine how accurate the model is. This will be done by providing the model with only the estimated skin temperature and comparing the core temperature the model calculates to the measured core temperature. To quantify this comparison it is best to calculate the error between the calculated core temperature and the measured core temperature. This will need to be done for a large set of data to get a good idea of how well the model is working. The error calculation will provide valuable information that will allow for an accurate evaluation of any error in the relationship between the calculated core temperature and the measured core temperature.

Absolute error will be used for finding the error of this model. Any error at all is detrimental in this case and absolute error provides a simple evaluation of if the model is accurate or not. Absolute error averages the size of the error and it weights each error the same. This is done by subtracting a measured quantity from a calculated quantity and taking the absolute value of the result, repeating this for all sets of measured and calculated values, adding all of these values together and dividing by the number of sets.

## **8 EVALUATION**

## **9 PRODUCTION MODEL**

The production model is the second piece of the overall project. What the production code does is it takes the model produced by the learning portion of the project as well as a thermal image, It then analyzes the image to produce a temperature which then gets fed into the model to predict a temperature. The output of the production model would then produce a single temperature based on the model. The model will be written in python like the other pieces of the project to eliminate any need for cross communication of languages. Although this is an important piece of the project most of it can only be done after the image analysis portion is complete. The production code also requires the model created by learning portion of the project. As long as there is an established format for model that will be produced we can create the code based on that format, then debug when the learning portion is complete.

We would not explicitly start on this portion at the very beginning, but instead we would need to research what kind of libraries we would need to process the image and model. Although it is a different part, the image processing is still integral in both the learning portion and the production portion. The first piece of the project that would be developed would be the image processing portion, which would be indirectly working on the production model. Both the learning, and production code rely on processing the thermal image to get a temperature to be used as input and analyzed.

Besides the image processing portion, there most likely will not be a need for any outside libraries. This means that the research for this portion of the project will be minimal and would mostly focus on ways to optimize the run time of the code.

We would like the production model to be accurate as possible since that is what will be used to predict a patient's core body temperature. This means we would want to test as many models as possible and run as many test cases in those models to produce the best possible case for core temperature prediction. To test the program itself we would use unit tests and mutation tests to test various temperature cases to make sure the program works with fringe cases as well as normal cases. The goal of the production model is to be as accurate as possible, but to also keep the false negative rate as low as possible. We are aiming for a 40

The timeline for the production code will roughly be, we wait till the image analysis is almost complete, then we start working on the code that analyzes the temperature. Second we work on the code that parses the model that is produced by the learning portion. Third we integrate the image analysis and debug anything that comes up from combining the two pieces. Fourth, we test the code to make sure everything runs smoothly and without any errors. Lastly, we test which model will be the best and give the lowest false negative rate.

## 10 USER INTERFACE

Once the project gets started a user interface is going to be one of the first things constructed. The first iteration of the user interface will most likely just be created out print statements that print out the results to the screen, and the input to the program will be done through command line. The user interface will be created using python's file input and print statements. It will be up to us to properly parse the input files and properly input the relevant information. This simple interface will most likely be used for most of the project, if not the whole project. As the project becomes more and more complete the user interface should also improve with the project. The user interface will slowly develop itself as the project continues to progress as we would need to print relevant information to the screen. The user interface for the majority of the project would just take command line arguments for input, and then output information using print statements. We would not need to explicitly work on the user interface until the end of the project when we want to fine tune what is printed to the screen.

This project contains two major pieces of code, the training portion of the program and the production model. The training piece takes in a picture and the actual core temperature, and then produces a model that is some line that will be used to predict core temperature. The production piece of this project takes in the model created by the training portion and a image. It will then produce a core body temperature. The user interface won't deviate from command line, until the project is reaching completion. Then a new user interface will be made from python that is nicer but still simple to use. Instead of using command line for input and output the user interface should allow the user to select the input file using the mouse and keyboard and then allow the user to choose where the output file should go. The output of the training portion should output a text file that is a model for the production model to take as input. The output of the production model should just output a temperature that is predicted from the model. Instead of outputting a text file, the production model should just output the predicted temperature to the screen. This user interface would be for our end product that will be handed over to a graduate student to finish the research and implementation.

There will be not extensive testing of the user interface. To test the command line user interface we would simply need to test which piece of data we would need to print to the screen and which piece of data we would need to take

as input. For the simple non-command line user interface we would need to test if it is grabbing the correct information from the input, and then correctly outputting the information to the screen. This would require some research into how the user interface libraries in python work and would need to be done before the project is near completion.

The time line of the user interface will roughly be, we first create a command line user interface and develop is as we develop the main programs. Second we do research into the various user interface libraries in python. Third we create a simple user interface using those libraries for the final product.

## **11 CONCLUSION**

## **REFERENCES**

- [1] C. Maimon, "Claude's technical review," [https://github.com/maimonc/Ebola-Virus-Project/blob/master/Documents/TechReview/Claude\\_Maimon\\_Tech\\_Review/Tech\\_Review\\_Claude\\_Maimon.pdf](https://github.com/maimonc/Ebola-Virus-Project/blob/master/Documents/TechReview/Claude_Maimon_Tech_Review/Tech_Review_Claude_Maimon.pdf), 2017.