CS CAPSTONE FINAL REPORT

JUNE 5, 2018

EBOLA PREDICTION MODEL

PREPARED FOR

PROFESSOR BILL SMART

PREPARED BY

GROUP 34
EBOLA TEAM
CLAUDE MAIMON
BRIAN LEE HUANG
BIANCA BEAUCHAMP

Abstract

ADD ABSTRACT

CONTENTS

1 Introduction

Who requested it?

Why was it requested?

What is its importance?

Who was/were your client(s)?

Who are the members of your team?

What were their roles?

What was the role of the client(s)? $I.e., did they supervise only, or did they participate in doing development$

2 Requirements Document

2.1 Old Requirement Document

Abstract

The end goal of this project is to end up with a research paper. The paper should outline the problem that the project is trying to solve, the steps taken to solve the problem and how successfully we were at solving the problem. The paper should also allow the project to be continued if someone chooses to. The whole process should be explained in detail allowing whoever wants to continue the project to continue without any problem. The main body of this research paper will be about the program that we develop to predict a person's core body temperature. The program should first be able to extract data from a thermal image. The data of the image should come from the top half, focusing on the head. It will then interpret the data to create a mathematical model that uses the temperature of a person's skin as data and analyzes that information and predicts what their core body temperature is. A high accuracy rate is not strictly required as that is not the point of the project, the goal of the is to determine if this method will be effective to detect whether a person is symptomatic with Ebola. A high accuracy rate is a good indicator that a mathematical model is a good way to predict, where a low accuracy rate indicates that we should look for an alternative method.

Introduction

Purpose

In the medical industry it is known that one of the first signs of illness is an elevated core body temperature. Currently, the only way to get this temperature is with contact sensors which put health care workers at risk of infection if a patient is ill. This project will aim to solve this problem by creating a device that will be able to quickly take a person's core body temperature from a distance using only stand-off sensors.

Scope

In Africa throughout the years there have been many different Ebola outbreaks. This project is part of a larger project that is trying to use robotics and automation to fight against Ebola. The National Institutes of Health are funding this project and we are also working with Medecins Sans Frontieres (Doctors Without Borders) in Brussels, Belgium who lead the response to the 2015-2017 Ebola outbreak. This project, if successful, will be part of an automated system that will sort people into two groups (symptomatic and asymptomatic) as they come in for care. This will lower the risk of health care workers and asymptomatic patients being exposed to Ebola.

Definitions, Acronyms and Abbreviations

OpenCV: an open source computer vision library.

Overview

We will be creating this automated system by using pictures from thermal imaging camera and analyzing them to estimate core body temperature. We are going to attempt to do this by taking an average temperature of the head of the person in the thermal image, taking their core body temperature with an ear thermometer and comparing the average temperature from the image to their actual core body temperature to build a mathematical model that will estimate core temperature from the thermal image temperature. To get the average temperature of the person's head from the thermal image we plan to isolate the head from the thermal image, make the pixels of the image into data points on a graph, get rid of the outliers, and take the average of the rest of the data points. We will also be collaborating with the mechanical engineering team to create this automated system. The mechanical team will be providing us with a thermal imaging camera and the physical structure that will hold the camera.

Overall Description

Product Perspective

Our program will be working with a thermal camera for the data collection. Our program will be working with a thermal camera for the data collection. We will also be using OpenCV for the image processing. Our program will also have a user interface that allows for easy data input. This user interface should be friendly enough where someone without an engineering background can easily use our program.

Product Functions

Processing Thermal Images

We will create a program that will process data taken from a thermal camera image. The program will then isolate the person in the picture and only take data from the upper body of the person. In order to isolate the person, we will use a specific background or a door frame to cut only the person person from the image. Once the image is fully processed the pixels will be used as data points. The temperature value will depend on the color of the pixels and the program will analyze the data from all the pixels and process it.

Processing Data from Pixels

After isolating the head of a person from the image the pixels will be used as data points for our program. The temperature of each pixel will graphed, to create a histogram which we can analyze. The extreme outliers in the histogram will not be counted towards the analysis as those data points could come from the background. We will then analyze the data to get an estimation of the person's temperature.

Collecting Data

To collect data we will use a thermal camera to take the temperature of their skin, mainly focusing on their head and neck. Every person that has their picture taken will also have their real core temperature checked through the ear. The two pieces of data will then be stored for the program to analyze.

Analysing Data from Thermal Image

Once we've collected enough data, we will statistically analyze it different ways. We will then look for the best statistical analysis that best connects the collected ear temperature to the predicted processed temperature from the thermal image. The statistical analysis will then be used as a baseline for a mathematical model to predict a person's core body temperature.

Mathematical Model

The data collected from the analysis will then be used to feed into a the mathematical model that predicts core body temperature. The accuracy of the model will improve over time as more data is fed into it.

User Characteristics

Our program should require no background knowledge in engineering to use. The usage of our program should be as simple as walking through a doorway and having a simple yes or no output to a screen. The hope of this project is to have it be deployed in areas where there is an Ebola outbreak. For this to be effectively deployed an expert should not be required to use it.

Constraints

Working with the Mechanical Team

For this project, we will be partnering up with a mechanical engineering team. Our project progress would be highly affected by the team's input. The mechanical team will be responsible for working with the camera and building the sensors set up. Our progress might be affected by the other team's progress.

Getting People to be Checked

Getting data will involve participants. We will need to check people for their temperatures. Participants can be people from our class but not other people. This constraint might make it harder to get sufficient data collection to create a comprehensive model.

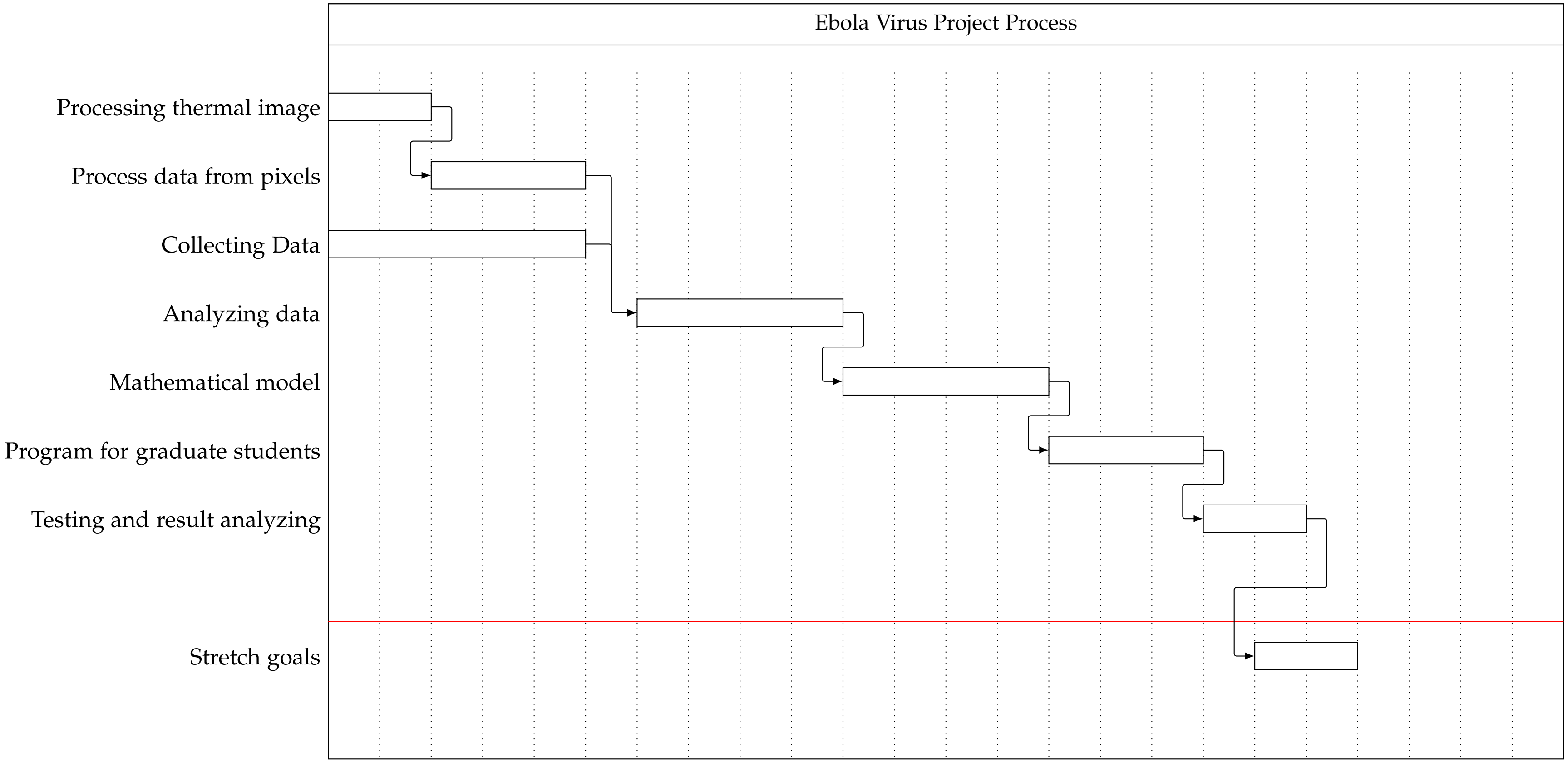
Specific Data Needed

In order for this project to work, we need to collect data from people. The data would be a set of ear temperature and a thermal image of a person.For the project to work, we need enough data from all spectrums. This means that we need to get data from people with different core body temperatures. We need to collect data from people with high and low core body temperatures. This mean people will either have to have a fevers or they would have to elevate their body temperature by exercise. This constraint will make it harder to collect large sets of data.

Limitations of the Camera

The accuracy of the camera will reflect on the research's results. If the camera is not accurate, the model will not be accurate. We might end up spending a lot of time creating an insufficient model.

Assumptions and Dependencies


Ebola Virus Project Process
Processing thermal image
Process data from pixels
Collecting Data
Analyzing data
Mathematical model
Program for graduate students
Testing and result analyzing
Stretch goals

Specific requirements

Our program will be able to take a thermal image and perform pixels selection. After that, the program will analyze the pixels in the image and produce an average temperature of the skin.The program will then run the observed skin temperature calculated from the camera through a mathematical model that will produce an estimated core body temperature.

The whole process will take less than one minute and will no have more than more than 40% of false negative results. In the end of the process, the program should provide a simple yes or no to whether or not the person's core body temperature is elevated.

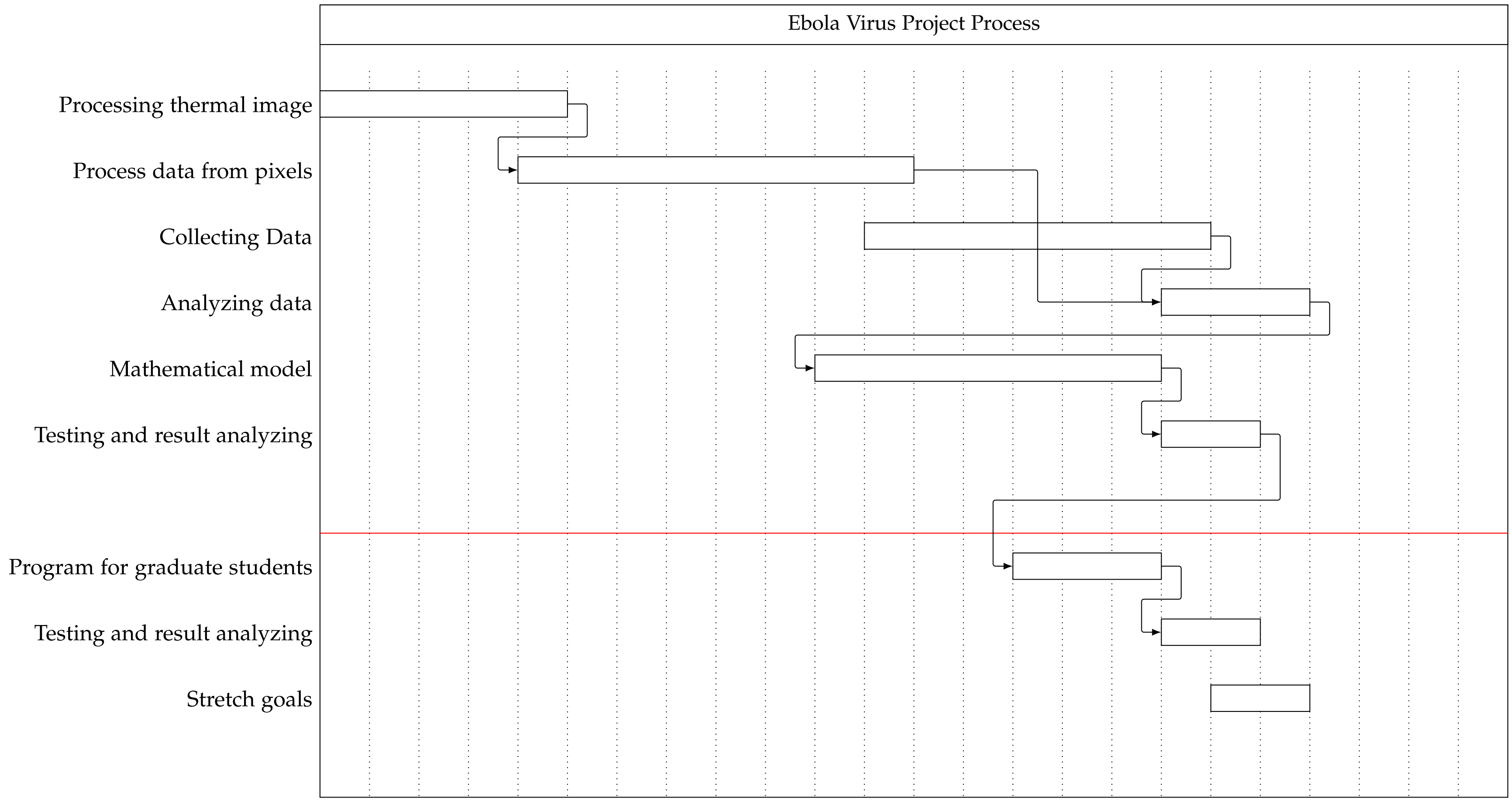
Stretch Goals

We will go out and collect data to feed into the model, and make it able to predict more accurately.If we will achieve our baseline of 40% accuracy[1] early, we will work to lower the percentage of the false negative.

2.2 New Requirements

There were two main things that we changed in the requirement documents. The first was the 40 percent success rate. From the initial research that we did, we found that some previous attempts of this approach produced about 40 percent success rate. When we started the project we thought that we could achieve that success rate. However, as we'll discuss in this report, we ran into many problems with the camera. The camera wasn't calibrated right and it returned bad values. We tried many things to fix it even though it is not in our requirements to try and fix the camera. When first writing this document, we assumed the camera would be working correctly. Since that wasn't the case, we had a hard time collecting good data. That's why we took out the 40 percent success rate goal. Our client, Bill Smart, agreed with this change. Another change that we made was producing a report and not a research paper as a final product with the code. Our client agreed that there is no need for a research paper. We just needed to produce a report that will help his graduate students in continuing this project.

2.3 Final Gantt Chart


Ebola Virus Project Process
Processing thermal image
Process data from pixels
Collecting Data
Analyzing data
Mathematical model
Testing and result analyzing
Program for graduate students
Testing and result analyzing
Stretch goals

3 Design Document

3.1 Original Design Document

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.

Introduction

There are four major parts to this project, image processing, machine learning, production and evaluation. The image processing portion will import the thermal images from the camera, format the images, select necessary pixels, and summarize those pixels into a single temperature value. The machine learning portion of the project will create a mathematical model to represent the relationship between skin temperature and core body temperature and then test this model on different sets of data. The production portion will use the model that has been created to produce a estimated core body temperature from the estimated skin temperature and then produce a true or false output as to weather or not that subject has a fever. The evaluation portion will evaluate how accurate the true or false output of the production portion is. The machine learning, production and evaluation portions will have user interfaces. The first piece of the project that will be developed is be the image processing portion since both the learning, and production code rely on processing the thermal image to get a skin temperature. The second piece that will be worked on is the machine learning portion since the model needs to be created and tested before it goes into production. The third piece that will be done is the production portion and the forth piece that will be done is the evaluation portion.

Importing Images and Image 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.[2]

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.

Formatting Images

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-dimensional 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.[2]

Pixel selection

In order to get a good temperature estimation from the image, only the pixels from the warm parts of the subject's head are needed. The program will go through the pixel values in a two-dimensional 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. If they are not represented as degrees the pixel values that represent 98.8 and 105 degrees will be found and those values will be used instead. This approach will work because by only selecting these values it will, in theory, isolate only the warm parts of the subjects's head and it will leave out the cooler parts of the head as well as the background behind the subject.

Since this is a primarily a research project this method will tried first, but if is fails a different approach to selecting the pixels may be taken. Another approach may be to isolate the subject's head from the picture before getting the pixel values. This way the background would be removed prior to selecting the warmest parts of the face which would in theory eliminate the chance of the background being included in the data.

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.[3]

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 nth 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. [3]

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. [3]

Production Mode

The production mode is the third piece of the overall project. The production mode will use the model produced by the learning portion of the project to analyze the image. The output of the production mode will be the estimated core body temperature produced by the model as well as a true (fever) or false (no fever) statement. This portion of code 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. The production mode will use the image processing portion of the project to get the data from the image and then it will use the model created by the learning portion of the project to estimate the core body temperature of the subject.

The production mode should be accurate as possible since it will be used to predict a patient's core body temperature. This would mean keeping the false negative rate as low as possible. However, since this is primarily a research project it is not realistic to have a perfect program in such a small time frame. So the goal that has been established is a 40 percent false negative rate. [4]

Evaluation

After the production model is created it needs to be tested. The accuracy can be evaluated by examining either false positives or false negatives. In this case false negatives are critical to keep as low as possible since placing a sick person in a group of healthy people could result in spreading disease to the masses. False positives are still very important but not as important since having a healthy patient among unhealthy patients will not spread disease to the masses.

To gather data for the false negative and false positive rates, the binary output of the program will be collected into a text file. In another text file there will be the corresponding known binary values. A program will be created to parse the text files, count the number of false negatives and false positives and output the percentage of both false negatives and false positives.

User Interface

Once the project gets started a user interface is going to be one of the first things constructed. The initial user interface will have a command line user input and print out the results to the screen. The user interface will be created using python's file input and print statements. It will be up to the user to properly parse the input files and properly input the relevant information. This simple interface will be used for most of the project, if not the whole project. This will depend on how accurate the model becomes and how much time is left. If the model is very accurate and there is time left a nicer user interface will be created. As the project becomes more and more complete the user interface should also improve with the project. The user interface will slowly develop as the project continues to progress since more inputs and outputs may become necessary.

If there is time to improve the user interface a GUI (graphical user interface) will be created. Instead of using command line for input and output the user interface will allow the user to select the input file using a mouse and keyboard and then allow the user to choose where the output file should go. The output of the machine learning portion should output a text file that contains the model. The output of the production mode should output a temperature that is predicted from the model as well as a true or false statement to the screen.[4]

Conclusion

The goal of this project is to increase the safety of health care workers as well as the people coming to those workers for help. By taking a thermal image of the patient and using that data to predict their core body temperature, health care workers and healthy patients will not have to be within a close proximity to sick patients. To accomplish this task, thermal images will be taken, the images will be processed, a mathematical model will be made, the model will be tested, a production mode will be created and the production mode will be evaluated.

If successful, this project will create something that will be usable in the field for doctors without borders. If the accuracy is not high enough for use in the field, it will be handed over to graduate students to continue the project. Ultimately, if this project is successful and has accurate prediction rates it could reduce the spread of illnesses.

3.2 New Design Changes

The main change that we did in the design document is the same as the requirement change. We changed the final product from a research paper to a detailed report. Our client agreed that there is no need for a research paper. Instead, we produced a detailed report that can assist someone in continuing our project in the future.

4 Tech Reviews

4.1 Brian's Tech Review

Abstract

The goal of this project is to be able to determine a person's core body temperature just by taking a thermal image of them, and analyzing the data take from that picture. This requires many parts, but the three I am tasked with is the Production code, Statistical Analysis, and the User Interface. To create these piece of the project I need to choose a pieces of technology to create them. For the production code I chose to use python, and analysis I chose to use R, a language specifically designed for statistical computing and analysis. For the user interface I am choosing to use command line as it is something easy to implement through print statements and command line arguments. However the choice for the user interface may change in the future depending on how far the project gets. The most important feature about all my choices is that each option is open source meaning there will be no licensing issues.

Introduction

The goal of this project is to create a program that uses a thermal camera to measure a person's skin temperature, then use that data to predict their core temperature. An elevated core temperature is an indicator that the person is symptomatic with the Ebola virus. The model and the analysis will be what predicts the core body temperature given a skin temperature.

For this project I am tasked with creating the Production code, User Interface and the Statistical Analysis. The production code and the analysis are mostly done towards the end of the project, while the user interface can be done at almost anytime during the project. The user interface should be very simple and not require much experience to use it. For most of the project a command line user interface will be sufficient.

Production Model

Overview

The production code is the second step, and what would be the final product of this project. As inputs it will take in a model, then take in a picture to then analyze the picture using the model provided. The output of this will just output a core body temperature.

Critera

The production code should try and be as accurate as possible, but should try to keep the false negative rate as low as possible. False negative should kept as low as possible as it can be extremely detrimental to misdiagnose someone that is symptomatic and putting them in the healthy group, as opposed to misdiagnosing someone that doesn't have it and putting them the unhealthy group.

We have found other projects that are similar to our project. They took a thermal image of a person's ear and head to derive a person's core body temperature. In this paper they achieved a positive predicted value of 40

Potential Choices

For this aspect of the project we have chosen three different possibilities to create a model. They are python, MATLAB and R.

First Choice

Python is a versatile language that we planned to use from the very beginning. It is easy to use, and has very many useful functions that makes data manipulation easier. It also has many plugins and libraries that can be used for model creation. We also plan to use python for our image analysis, so it will be easier to use the same language for both of these portions to make it easy to incorporate it.

Second Choice

MATLAB is a computing language created by MathWorks which is primarily used for numerical computing. This makes it an obvious potential choice to analyze the data from the image. MATLAB has many functions built into it so we mostly do not write any of the functions ourself for the analysis. The main difference between Python and R is that MATLAB requires a licence.

Third Choice

R is language used for statistical analysis and graphing. It can easily analyze the data taken from the image. R also includes many packages, so we can simply add in any feature that we need in the future. One small potential problem that we may run into with R is to have it run with our image processing program. We would need to set up a properly formatted text document as the output of our image processing and then run R with the document as the input.

Discussion

The first and most important distinction to make between the three choices is that Python and R are both open source, while MATLAB is not. This in itself makes MATLAB not a good option for us. Python is more of a general purpose language, while R has a specific use. For inputting data R is much better than Python because it is simply a few simple commands, while in python we will most likely have to write our own parser. Analyzing the data will most likely be easier in R as its intended use is to analyze data, while python is just general purpose.

Conclusion

Matlab requires a licence while the other two choices are free, so we most likely will not use MATLAB. We will most likely use python as our image handling program will also most likely be written in python. This allows us to just combine the program rather than handling the input and output of two different languages. However R will also be very useful for analyzing the data from the image, but incorporating the pre made model could be difficult.

Statistical Analysis

Overview

The statistical analysis is the intermediate step between the mathematical model and the data processing of the thermal images. We will use different statistical analysis to determine which is best for the creation of the mathematical model.

Critera

The analysis should show some kind of trend in the data. It should be easy to tell what different factors affect a person's core body temperature.

Potential Choices

For this aspect of our project we would use similar tools, if not the same tool for the mathematical model. Our choices for this piece are MATLAB, R, and Microsoft Excel.

First Choice

MATLAB can easily process huge amounts of data and create multiple different analysis of the data. It has many built in functions so the different analysis could simply be changed by changing a name, or a number of a function.

Second Choice

R is an obvious choice for as the whole point of R is statistical analysis. Just like MATLAB it can easily process huge amounts of data and create many different analysis. R also has many built in functions that do different statistical analysis to make the whole process easier. Data input and output would also be extremely simple with R as it simply uses a function call or two for data input and then another to process. Data output can also be setup to output to either a text file or an excel document.

Third Choice

Excel is another choice we had as it is a tool to create spreadsheets and analyze data. Data input is simple, and most of the analysis is simple. Output is also simple as it is all done through the Excel environment. No code has to be written and it is all done through the user interface of the program. The statistical analysis most likely would need to be done by hand, where we would need to set up equations using the cells as the data.

Discussion

In terms of power, Excel is lacking when compared to MATLAB and R. MATLAB and R are both stronger in than excel in the analysis portion as they both can import different libraries and packages if the built in tools are not enough. However both MATLAB and Excel require licences while R is open source. The versatility and the whole purpose of R make it the a clear choice for the statistical analysis. With easy input and output, being open source and its whole purpose being to analyze data.

Conclusion

R is the clear choice for this piece of the project as it is the most powerful and it is also free. Both MATLAB and Excel require licences that could become an issue in the future. Using an open source software makes it easier for us, and easier for anyone that intends to work on this in the future.

User Interface

Overview

The user interface is what we will be using to input our data and display the output. In the end we need to create something simple that can be used by someone without engineering or computer experience. However for most of the project a extremely simple UI can be used. A nicer UI is only required if our project gets into its end stage, where it is almost ready to be deployed in the field. If we do not get to that stage, a graduate student may take over the project and for them I command line UI will be sufficient.

Critera

For our end product we need something simple and clear to use. The output of the program should be displayed with a simple yes or no. For the graduate student version, it should be simple to input data and have a clear output showing what it is doing. The usage of should be quick and clean so data input can be fast.

Potential Choices

For this piece of the project depending on where we are, we could use command line. However for the simpler version we chose to use either Java or Python. Command line is a good option because it can be written in almost any language and does not require any extra libraries or packages to install. Java and Python are also good options for a nicer looking UI because they have so many libraries and packages available to them.

First Choice

Command line is an easy choice for someone with a background in computer science. If a graduate student were to take over this project in the end a simple command line UI should be sufficient. A command line UI is easy to create, compatible with almost any language and easy to use, with a slight learning curve. Inputting data can be done with some command line arguments, while outputting can just be done with a couple of print statements.

Second Choice

Python is a good choice for UI design as it is easy and simple to use and there are many different libraries and toolkits available for python User Interface design. Python is also a good option for us in particular as we have planned to create the rest of the project in Python. This means that there should be no miscommunication between the languages when putting all the pieces together.

Third Choice

Java, just like Python has many libraries and toolkits built just for creating an user interface. One problem we may run into with using Java is the communication between the other parts of our project. We intend to create the other pieces using Python so creating a Java UI requires us have the two languages communicate which could cause issues.

Discussion

The command line interface will be easy to create as it can be created inside the main program.The other choices will have some learning required as the we would need research which toolkit we would want to use, and how to use the toolkit. However if we were to use one of the tool kits the user interface would be more robust and friendlier to the input and output of data.

Conclusion

Depending on where the project ends we can choose what kind of UI we want to make. For most of the project we will most likely be using a command line UI, where all the input and output is done from the program itself. In the end we will most likely create the UI using Python, as it will most likely be the easiest because we have already planned on using python.

4.2 Claude's Tech Review

Abstract

Currently, to check for Ebola, doctors must use contact sensors such as thermometers to test for an elevated core body temperature in patients. This process is slow and can potentially lead to infection of staff members as well as other patients. This project will aim to create a device that will be able to quickly estimate a persons core body temperature from a distance using a thermal camera. To do this, we will process the images and then run the data through a model that will return an estimated core body temperature.

My Role in The Project

I am responsible for the image processing part of the project. For our project, we are going to take pictures of people using a FLIR thermal camera. We will use the images to determine peoples core body temperature. For my part, I will import the images from the camera, isolating the head of a person from the image, and process the thermal information out of the images pixels.

What are you trying to accomplish

We are trying to create a device that will minimize the exposure of staff and patients to the Ebola virus. To do so, we are using thermal images to predict peoples core body temperature. People that arrive at the health center will walk through the device which will screen their core body temperature. We will use a FLIR thermal image for this project. A thermal camera can pick the approximate skin temperature, but there are many different factors that could increase a person's skin temperature. We will process images from the thermal camera and get an approximation of a persons skin temperature. We will create a computational model that will use the data from the camera to determine a persons core body temperature.

IMPORTING THE IMAGE

Overview

The first step for image processing is importing the image from the camera. Its important that the image will stay in the right format. The image needs to be in a format which maintains the temperature data.

Criteria

The tool needs to work with our provided camera and sensors. It should be fast and should create the right image format.

Potential Choices

Flir Tools

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 our FLIR A315 camera. Other than importing images the tool can be used to create PDF reports, add headers logos to images and sort the image folder by specific variables

FLIR Tools+ Reporting Software

The FLIR Tools+ software provides all the features FLIR Tools offers combined with extra features. It offers controls for generating more comprehensive thermal imaging examinations and allows for better research reports. FLIR Tools+ also allows for recording and playing of radiometric videos. It costs $295 and offers all image import capabilities described in the previous tool. It has an easy to use interface and is easy to install.

Copy Paste from driver

The third option is to copy paste the images from the images from the camera drive to our personal computer.[4] This option would be easy to do because weve all done that before. We will just open the camera drive on our computer and copy the images to a specified folder on our computer. This approach might be slower because we will have to transfer the images manually. Another downsize is that its easy to miss file when copying and pasting. The order of the images might also change using this method. This option is free since it doesnt require additional programs to import the images.

Discussion

All three options would work with our camera. They are all straightforward and are easy to use. For price comparison, FLIR Tools+ is much more expensive than the other two options, since they are offered for free. Both FLIR Tools and FLIR Tools+ offer more capabilities and better interfaces than the copy paste option; however, FLIR Tools offers all the features we need to transfer images from the camera. The FLIR Tools+ extra features are not an interest for our use case.

Conclusion

We will use the FLIR Tools software to transfer files from our FLIR A315 camera to our computer. This is the best option for us because its free and it offers all the features we need for this process. Even though FLIR Tools+ offers more features, we dont need them. By using the free option, we will make the project cheaper and still get all the capabilities we need to transfer the files.

MANIPULATING THE IMAGE

Overview

The image imported from the camera would be an image of a person's full body. For our analyzing, we will only need the persons head part of the image. We will manipulate the image to cut the person from the background and cut the head of the person from the image as well. The final product should be a head-image of a person with no background to it.

Critera

The tool should be free or cheap to purchase. It should be well documented and easy to learn. The bigger the community of users for the software, the better. It will be easier to use a tool that other people use.

Potential Choices

OpenCV with Python

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[5]. OpenCVs library has more than 2500 algorithms which offer many features. Some of those features are facial recognition, gesture recognition, motion understanding, biomedical analysis and more. The software is used all around the world and has a strong user community and offers technical support. OpenCV is also known by our client so we will have access to people that know how to work with it.

VLFeat

The VLFeat is a free open source library for vision algorithms.[6] It offers algorithms for image processing in a simple and portable package. It includes implementations for common building blocks such as feature detectors, k-means clustering, and super-pixelization. VLFeat is fully documentations and offers usages examples to many algorithms. It has a high-quality implementation and is good for computer vision researchers and students. It has no external software dependencies and is accessible by a MATLAB interface. There is also a command line interface offered. The library is written in C and is free to download.

MATLAB Image Processing Toolbox

MATLAB is a high-level language for technical computing. It integrates computation with an easy to use interface. It offers libraries for mathematics, algorithm development, graphics and more. Most importantly for our use case, it offers an Image Processing Toolbox. The toolbox offers algorithms and workflow for image processing, imaging analyzing and algorithm development. It is free and easy to use.

Discussion

All three options offer the capabilities we need to manipulate our images. MATLAB would be the easiest to use with an easy programing language and easy to use interface which will lead to easy debugging.[9] However, MATLAB is much slower than the VLfeat and OpenCV.[10] In fact, OpenCV is also faster than VLfeat but for our implementation, both will be fast enough as this the difference will not be obvious in our implementation. MATLAB commercial use (not needed now but might be needed in the future) 1500 euros. It is much more expensive than the other two options as those options are free. Both also have a lot of users, so it will be easy to find solutions or assistance if we run into problems. Moreover, if we choose to use OpenCV, our client and his TAs would be able to assist us as they have previous knowledge and experience with OpenCV.

Conclusion

For this part of the project, we will use OpenCV. It is free and has a large users community. It has shown to have best performances and it has many easy to understand tutorials available. Moreover, our client suggested that we will use it so he or his TAs could help us with the process.

ANALYZING IMAGES PIXELS

Overview

The last step for image processing would be extracting the images pixels. In the thermal image from the thermal camera each pixel is holding a temperature value. We need to extract that value for each pixel in the image.

Criteria

The program should be able to extract all pixels information from the image and should work with our image type. It should should be as cheap as possible.

Potential Choices

FLIR ResearchIR Software

ResearchIR is a thermal analysis software by FLIR. It is easy and free to use. The software can be connected to a FLIR thermal camera and process images in real-time. It offers image analysis tools, image file explorer, high-resolution image scaling and temporal plots. Its easy to extract pixels data from an image using ResearchIR.

MATLAB

MATLAB, as described previously, is a high-level programming language and an easy to use interface. It can be used for thermal analysis of FLIR cameras images. MATLAB can be used for interfacing with thermal imaging devices, analyzing thermal images and objects detection. It is free to use but has a high cost for commercial use.

FLIR WebViewer

FLIRWeb Viewer is an online tool for analyzing thermal images. It works with our camera and has an easy interface. It can be used for uploading images, manipulate measurement tool, get camera properties and more.

Discussion

All three programs can be used for our purpose. ResearchIR is the fastest option out of the three. MATLAB offers more features and analyzing algorithms than the other two options; however, ResearchIR offers all the features that we need to process the image's pixels. WebView offers a remote connection to the camera but we dont need that feature for our project.

Conclusion

For analyzing images pixels, ResearchIR would be the easiest and most straightforward solution. Its free and connects directly to the camera. It has support from FLIR costumer services and it offers all the features that our project requires.

4.3 Bianca's Tech Review

Abstract

There is currently no way to take a persons temperature without being within a close proximity to their bodies, putting health care workers at a great risk of infection. The purpose of this project is to reduce this risk by creating a device that will be able to quickly take a persons core body temperature from a distance using a thermal camera. In order to do this the thermal image must be processed and a model needs to be created. To process the image the pixel data must first be extracted and sorted, then the outliers will be removed. To create the model, summary statistics will be preformed to consolidate the data into a single value. That value will be used to model the relationship between skin temperature and core body temperature. Then the model will be tested by calculating the error of the model. This document will discuss some of the possible options and come to a conclusion best option for implementing the summary statistics, model, and model test. The best option for the summary statistics was determined to be the mean because it is the most accurate option and it is the simplest to find. The best option for the model was determined to be the linear regression because it is the simplest type of model and using it is a good starting place to see if it works of if a more complex model is needed. The best option for testing the model is absolute error because it provides a simple evaluation of if the model is accurate or not.

Summary Statistics

Overview

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. There are three good options to consider for this purpose. The first is taking the median of the data, the second is to find the variance of the data and the third is to find the mean of the data.

Criteria

The option chosen has to be the most accurate representation of the data. This is important because this value will represent the skin temperature and will be used in relation to core body temperature. The model that creates the relationship is the most important part of this project. Therefore, the value produced by this step must be a good representation of the skin temperature.

Potential Choices

Median

The median is the value that separates the upper and lower halves of the data. The median is found by sorting all of the data points from least to greatest or from greatest to least and then finding the value that is in the middle of this set of data. If the number of data points is odd then a value will exist in the middle of the data set but if the number of data points is even the two data points that are closest to the middle are averaged. One advantage of using the median is that it minimizes the impact of outliers in its representation of the data set.[5]

Variance

The variance measures how far a set of random values from the data set are from the average value of the data set. Variance is calculated by squaring the standard deviation of the set of data. The variance is typically used to identify the causes of variability in a data set.[6]

Mean

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. [7]

Discussion

The key characteristic of the median is that it minimizes the impact of the outliers in that data set. This is not very important in this case because by the time the data is ready have the summary statistics taken, the outliers have already been removed. The key characteristic of the mean is that it is the best representation of every value in the data set. This is very important because the value produced by the summary statistics will be used to represent the the data set in the model. The key characteristic of the variance is identifying the causes of variability. This is not very useful in this case because the value from the summary statistics will be used to represent the data and not analyze it.

Conclusion

The mean is the best choice for the summary statistics because it is the most accurate representation of all the data points and it is simple to find. 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.

Modeling

Overview

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.

Criteria

The model must be able to represent the relationship as accurately as possible. This means that the best option for the model 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.

Potential Choices

Linear Regression

A linear regression 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.[8]

Polynomial Regression

A polynomial regression model is used for modeling the relationship between dependant and explanatory variables as an nth 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.[9]

Ridge Regression

Ridge regression 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.[10]

Discussion

The linear regression is the least complex of the models since it is finding a simple line to represent the relationship between core and skin temperatures. The polynomial regression is mildly complex since it is used for modeling nonlinear relationships. The ridge regression is more complex since it is similar to the linear regression but considers independent variables that are correlated and takes the error into account.

Conclusion

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.

Model Testing

Overview

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.

Criteria

The calculation used to find the error needs to be the best representation of the accuracy of the model. To do this it needs to 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.

Potential Choices

Absolute Error

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.[11]

Squared Error

Squared error measures the differences between the calculated and measured values by putting a larger emphasis on values that are more consistent throughout the data set. This is done by subtracting a measured quantity from a calculated quantity and squaring 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.[11]

Root Mean Square Error

Root mean square error represents the spread of the calculated and measured values from each other. This is done by subtracting a measured quantity from a calculated quantity and squaring the result, repeating this for all sets of measured and calculated values, adding all of these values together, dividing the result by the number of sets, and then taking the square root of the result.[12][13]

Discussion

The absolute error represents the size of the error of all the data sets and puts the same emphasis on all data sets. Unlike the absolute error, the squared error puts more emphasis on the errors that occur more often in the data sets. The root mean square error measures the magnitude of the error much like the absolute error but it puts a larger emphasis on larger errors.

Conclusion

The absolute error is the best error calculation for finding the error of this particular model. Any error at all is detrimental in this case. Putting emphasis on larger errors or on errors that occur more often won't offer any extra benefit overall. They do offer benefits for fine tuning the model because they offer extra information but are not needed to know the overall error.

5 Weekly Blog Posts

5.1 Brian

Fall

Week1

Finished picking my projects.

Week2

I met with my group mates this week and exchanged contact information. We decided on a time to meet with our client, however our meeting time will be after the problem statement is due, so we may need to talk to our professors about this.

Week3

We met with our client and TA this week. After listening to the client and telling the TA about our meeting he voiced his concerns that this project may have too many components for us to do and it could potentially be too difficult for us to do. I worked and finished my rough draft problem statement after meeting with the client to add more details to my initial draft.

Week4

Met with the client to talk about our concerns about the project. The project was changed to something more research oriented so a high accuracy rating was not absolutely required. OpenVL was suggested for image handling which was also one of our concerns. We still have no camera, and still have no idea where the data is coming from.

Week5

We now have more guidance and more of an idea as to what to expect for the future of the project. We now have a better idea as to what is going on and how we will be doing the analysis. However we still do not have a camera and are still unsure about where the data will be coming from

Week6

We had trouble finishing the requirements document as it was still unclear to the group what the final product will be. The client did not show up for the meeting, so we couldn't double check during the meeting about the requirements. We have sent an email, and are currently waiting for a response.

Week7

Not a lot happened this week. No meeting this week because of the holiday on Friday. All that has happened this week is that I looked at the tech review assignment.

Week8

The requirements for the project changed again, so we had to change some of our tech review document to fit these changes. This also meant that we would have to go back and change our requirements document as well.

Week9

It's Thanksgiving break, nothing got done.

Week10

This week we had trouble getting the design document finished as we could not complete a portion of it. We could not complete it because we had absolutely no idea what was going on with that part. We are allowed to take an incomplete, but I would prefer if we didn't. During this weeks meeting we have our TA attend with us to help us clarify what was actually asked of us. After that meeting we continued to talk with the TA to make what was discussed in the previous meeting more clear. It was pretty helpful overall and it gave us a better idea on what was being asked, and how to fill out the tech review.

Winter

Week1

Nothing much happened this week. This week was mostly getting things together and preparing to create the project. We have a camera, some images and csv files so we are ready to start on the project. We also need to schedule a time where we can meet with Bill and Andrew.

Week2

We are slowly working on the project by creating the image processing piece by piece. Claude created the bulk of the program which analyzes the pixels in the images. I created something that creates a histogram out of the csv file, as well as a short program that detects the type of file. The plan is to put all these pieces together to run on both csv and tiff files.

Week3

We have the image analysis mostly complete. I created a fake data generator for any portion in the future that needs data. I have also created a reader to go with that data to make it easy to use in python. For the evaluation portion I have decided to look into something called Receiver operating characteristic. This is something that can minimize false positives, so it might also be able to minimize false negatives.

Week4

Not much progress on the project was made this week. This week most mostly a lot of planning on what we need to do, and what we needed from the mechanical team. We made some progress on the evaluation portion of the project, and found the a receiver operating characteristic curve would be helpful in evaluating the model.

We also worked with the camera a little bit. We found that the temperatures that it was producing seemed to be very off, so we need some sort of constant temperature to let the camera have something to compare to.

Week5

This week we were mostly working with the camera to see if we could fix the issue with the recorded temperature being too high. We tried multiple different reference temperature, and all of the failed to work. We however found that there was a folder supposed to be used for calibration, and found that when we used that folder the camera was only off my a couple of degrees. We plan on taking more pictures to homely have the camera calibrate correctly.

Week6

This week was mostly dedicated to working on the progress report. Claude and I recorded our parts together. Sill not very much progress with the Camera.

Week7

Not much was done this week besides testing the camera with the coffee. We were testing if the camera was adjusting to the hottest thing in the room, and the results of the test was that it does adjust. This makes calibrating the camera difficult, as it can only detect things in the range of temperature around the hottest thing it detects.

Week8

This week was super busy for me, so not much got accomplished this week. Claude and I met with Chris' to look at his code again and to see how it works.

Week9

This week we met with Chris again to go over his code. He wrote his code assuming that we have a Linux machine, so our task is to get one to take data with his code. I am going to try and get a Linux installed on my laptop so we can finally collect data with the camera. Chris' code seems to have better calibration than the window version, so we are more comfortable taking images with it.

Week10

We were finally able to gather some data from the camera. To do this we used Chris' code to grab some frames from the camera while it took some video. To do this we needed a linux computer, so I installed Ubuntu on my laptop to have a computer that could use the code. I also wanted to start piecing all the piece of our project together so I can start working on the production mode, and the user interface. However I am still missing the model code so I can't create it yet. I looked at the requirements for the report and the presentation. I plan on working on the report this weekend, and the presentation on Tuesday.

Spring

Week1

Nothing really happened this week. Our group just discussed times we could meet and what we need to accomplish before expo.

Week2

We mostly worked on the poster this week. We moved some things around and also wrote out some more stuff to put on the poster. I also did some more testing for the model portion of this project. It seems like the library that we were using just does a Least Squares Regression line. I wrote a small script that does that, so we don't have to import the library in the future if we are to give this project to someone else to work on.

Week3

Nothing really happened this week. The only thing that is important was during the meeting with Bill he said that there is a graduate student that is going to help us collect data. This would help greatly in creating the model in the future.

Week4

We were able to finish the poster. We decided to put a picture of the mechanical teams rig on the poster as it gives some context as to what we are doing. We also decided to leave the model as is as it works, but it is just highly dependent on data. Bill has also given us some code to use, but a lot of the function he uses inside the library are outdated and throws a lot of errors.

Week5

We talked with Andrew about normalizing the data to try and get better results with the image processing. It is something we can try later, but I feel like that will throw errors in the future because if there is no person in the background, and it gets normalized it would detect a person begin there.

I also worked on the production mode a little bit this week. I tweaked the camera code a little bit so it takes more pictures. I need to look at the image processing code more thoroughly as I don't think I am getting the correct outputs.

Week6

I finished working on the production mode and then put the finishing touches on it with Claude. I made it so the camera would take 40 frames, but discard the first 10 frames in case the camera is still warming up. Then the program averages the temperature data from all of the 30 remaining frames, and then takes the average of the 30 averages. This is then used as the output as the predicted core body temperature.

We also cleaned up the GitHub repository this week, as well as adding readmes to all of the necessary files.

Week7

This week we had expo, so everything was finished. Nothing much was done other than expo.

Week8

The final report was assigned this week. I didn't do much other than write one of my sections in the final report.

Week9

This week was dedicated to working on the final report. I finished up my sections and started to work on some other sections as well to get the report done faster. The document should be done soon, just need to convert into latex later and figure out how to add images/code to the latex document.

We also need to start on our presentation soon.

Week10

PENDING

5.2 Claude

Fall

Week1

This week I read through all of the project proposals and chose my top choices.

Week2

This week we received our project topic. We emailed our client to introduce ourselves and set up a time to meet with him. I also worked on the problem statement assignment.

Week3

I finished my individual problem statement. My group and I met with our client on Friday for the first time. We also met with our TA on Friday.

Week4

This week my team and I worked on the final problem statement. We met with our client and redefined our final solution. We understood that this is a proof of concept project and the solution might not work. We are going to test this approach and do our best to figure out if we can estimate the core body temperature from the images.

Week5

This week we focused on our project's requirements. We started writing them down but we noticed that we didn't have enough information. When we met with our client, he helped us with defining more requirements for the project. We discussed the camera the mechanical team is going to use and the tools we should use to learn how to process the images from the camera.

Week6

This week we worked on our requirement document. It was really helpful to focus on that because it gave us a better understanding on the project. Now it is clearer what the final product needs to be. We still need some more clarification from the client but overall we understand our project better now.

Week7

We've divided the project to three parts and each person it responsible to write on one. I will be responsible to working on the image processing part. I will need to cut the person from the image, and process the pixels out. I've been researching different technology options to do this.

Week8

This week I finished the tech review document. It was helpful to look into the different technologies and learn more about image processing. It showed me that there were so many resources for image processing.

Week9

This week was Thanksgiving week so I there wasn't a lot of time to get stuff done. I worked on fixing and finishing up my tech review document. I also started working on my design document draft.

Week10

This week we met with Bill and Andrew to discuss some of our problems with the project. We weren't sure if we could meet the reequipments for the project in time so we had to redefine it.

Winter

Week1

First week of Winter term. We finished all of our documents from Fall term and started processing thermal images from the camera.

Week2

We met with the mechanical team to talk about where we are in the project and what they've been up to. They will provide us a distance of which the camera will be positioned at. This way we will know how to take our images for now. We also decided that we need to start working on the AI part of the project.

Week3

This week Brian and I worked on the image processing code. We are almost done with most of it. If you don't count the camera problems.We need to start working on the model soon. I hope that we will start working on it this weekend or at the beginning of next week. We have a lot more to do for the project.

Week4

We are trying to figure out how to use the camera in the right way. It gives weird temperature readings. We need to see if the difference in the temperature is always "wrong" in the same way. If not, we will have to find a better way to calibrate the camera. Other than that, we are stuck on the model part. We are waiting to hear from Andrew.

Week5

Started working on the report and the poster. Got some progress with the camera, finally got some good readings. Met with Andrew to talk about our progress and about the model. Bill gave us some ideas for the model part. Over all we're making progress.

Week6

We worked on the report and the poster this week. We need to do a lot more work on the poster. We also had our in-class workshop for expo, it helped us understand the best way to talk about our project in expo. This week was a little stressful because we had many assignments to submit. We met with Bill, he said that he will take care of the camera problem by the weekend, we haven't heard from him yet but we hope that it will be done by Monday.

Week7

We didn't meet with Bill this week because we didn't have anything new. We met with his graduate student, Chris who helped us with importing the images from the camera. We still need to work with them on calibrating the camera. On Friday, Brian and I took the camera and tried what Bill suggested. We had a cup of coffee on the table and we took a picture every 5 minutes. The results were bad because the temperature of the table changed when the coffee's temperature changes, this is exactly what we didn't want to happen.

Week8

Brian and I met with Chris on Tuesday. We worked on the code to make the camera take an image and import it into the code and get the temperature values and enter them into a CSV. We got it to work but it returns values that are off by 3-4 Celsius . We emailed Bill about it and he said that this will work. Next we need to start collecting data. We need to go take pictures of people and take their body temperature at the same time and enter those data points into our code.

Week9

This week was really busy for all of us so we didn't make much progress.

Week10

This week we worked on taking new images to get data for the model. Unfortunately, after taking the images, we notices that the csv values are still off. They are higher than they are supposed to be. I emailed Chris, the graduate student, to try to work on this but I don't know if we we'll get it done by the end of the term.

Spring

Week1

I worked on the image processing code to try and help with the calibration of the camera. I think the image processing is done. Brian and I met to work on the poster, we made a lot of progress on the poster.

Week2

This week we divided the responsibilities that we have left. I am going to go over the documents and see what we need to change. Bianca will try to finish the poster. Brian will do the production mode.

Week3

We met with Bill this week. We talked to him about making some changes to some of our documents. He said the document changes are fine and that the poster is fine. He said there will be a graduate student that will try to collect some data for us.

Week4

This week we worked on finishing the poster. We had to make many small corrections to make it look good. We added the logos and a picture of the mechanical team structure. We also tool the team picture. I submitted the poster for printing.

Week5

This week we met with Andrew and we talked about trying to normalize the data. He suggested an approach to normalize the data but our client didn't want us to use it because he didn't think that normalizing the data will help with the calibration problem. Our client Bill write some code for the model and gave it to us as an example of a possible model. He also gave us a list of a few different models that would be good to try on our data.

Week6

This week we worked on making sure our code is ready for the code freeze. We organized the GitHub repository and worked on our production mode for Expo.

Week7

We spent the week preparing for expo. We saw that we didn't get the space that we asked for so we decided to just put the camera on the table at expo. In expo there were too many people so we could run our program (it need only one person in the frame). We ended up just running the camera software and explaining people about our project. People seemed to really like it. It was fun explaining it to people, especially the kids.

Week8

This week I started working on the final report for our client. We need to give him a report that will explain everything that we've done so his graduate student could continue the work.

Week9

This week I worked on the report for Bill and the final report for the class. For the final report for the class I started by putting all of the required documents together in one big LaTex file.

Week10

This week we worked on finishing the report and the presentation for the class.

5.3 Bianca

Fall

Week1

test

Week2

test

Week3

test

Week4

test

Week5

test

Week6

Week7

Week8

Week9

Week10

Winter

Week1

Week2

Week3

Week4

test

Week5

Week6

Week7

test

Week8

Week9

Week10

Spring

Week1

Week2

Week3

test

Week4

Week5

Week6

test

Week7

Week8

Week9

Week10

test

6 Final Poster

Our final poster as it was presented at the Undergraduate Engineering Expo.

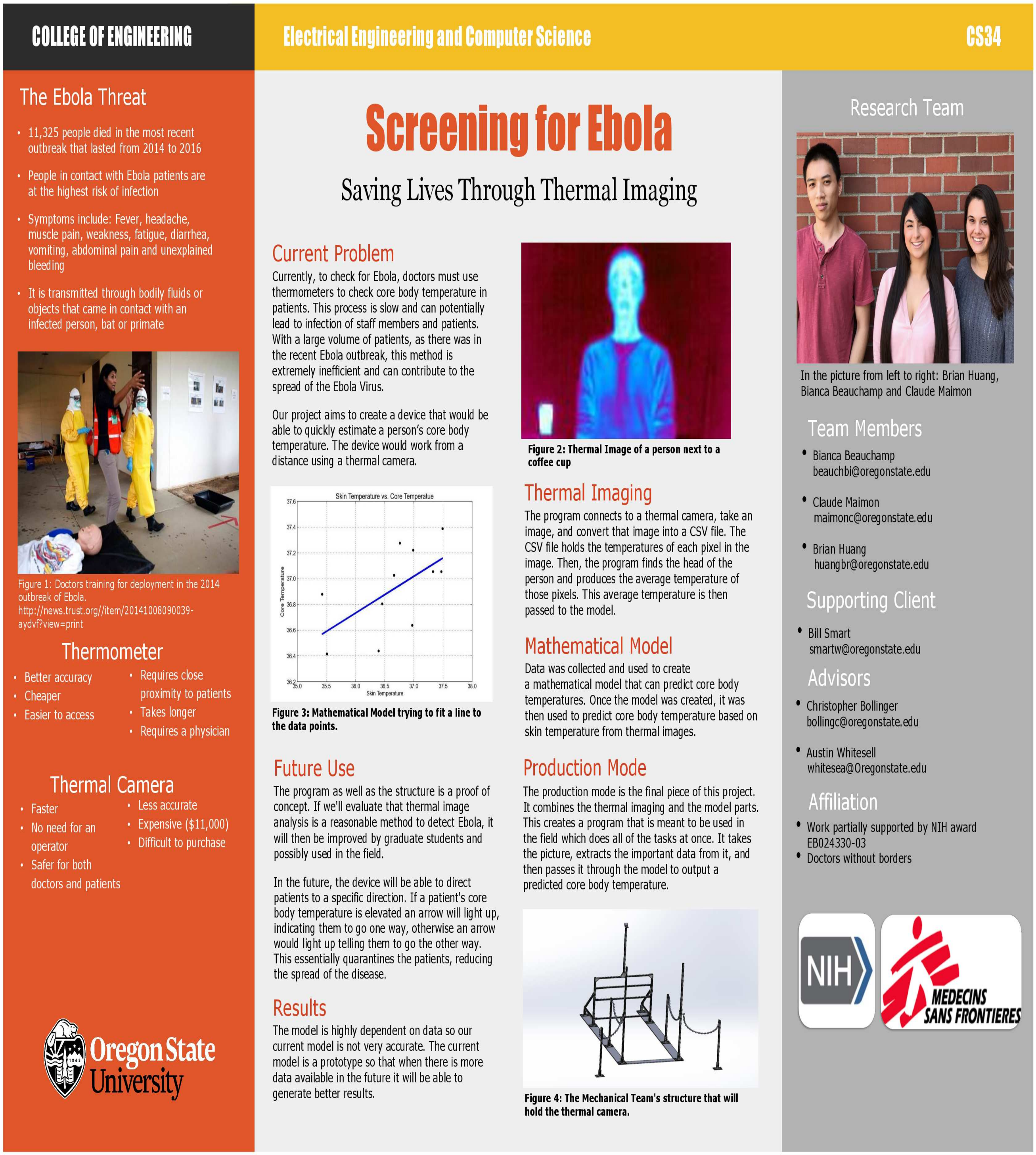


Fig. 1. Our final poster

7 Project Documentation

Code

Our code can be found at https://github.com/maimonc/Ebola-Virus-Project

Overview Of The Code

Our code has two main modes. One is a learning mode and one is a production mode. In the learning mode, the images processing code and the model code are used separately. In the production mode, the separate pieces of code are used together to output an estimated temperature.

In the learning mode, the image processing code processes a folder of CSV files that were taken in advance. This mode is to be used when trying to collect data to train the model. While using this mode, the user should collect the actual temperature of the people that are being photographed. After running the image processing code on the CSV folder, the thermometer temperatures need to be added to the file that the code produces. Then, that file (that holds both the actual and estimated temperature) needs to be sent into the model to create a new model.

In the production mode, the whole process happens at the same time. The code can only run while the camera is connected to the computer and can only run on Linux environment. When running the code, the camera will take a few pictures in a row, our code will process them and produce an estimated core body temperature.

Image Processing

The image processing code is used to process an image CSV file and produce a temperature mean value. That value can later be sent to the model code to create a model. The code finds the head of the person in the image, calculates the mean of the head pixels and returns that value. This code doesn't include the process of connecting to the camera and producing a CSV from the image. That code is available under the production mode.

Running the Program

In order for the program to run, a folder with CSV files needs to be specified. That folder should hold the CSV files that were produced by the camera. There are two ways to include the folder. The first is via a command line argument. If a folder path is specified at runtime (python images2.py ¡folder path¿) the program will use the files in that folder. However, if a folder was not specified, the program will use a built-in folder path that is specified in the MYPATH variable. It is recommended to use the built-in option when running this code in production more since the folder will not change. However, while training the model, it will be best to use the command line option.

Images.py line 124:

```python
if (len(sys.argv) == 1):
MYPATH = '<INSERT PATH TO FOLDER>'
elif (len(sys.argv)==2):
MYPATH = str(sys.argv[1])

```

Listing 1. Python example

Correcting Values

In the problems and solution section of this report, we will discuss some problems that we experienced with the camera. Our main problem was calibrating the camera. Since we couldn't find a good way to calibrate the camera completely, we tried to use our code to "fix" the values a little.

It's important to state that this is not a good solution. We couldn't find a way to fix the problem so we tried this to help a little. We found that at some times the background values are higher than usual. It looked as if subtracting the background value from all values helps to get the head values into a better range. We couldn't rationalize why this is happening and we don't think this approach is right. We only used it so we could start collecting data and proceed with the project.

The following function, correcting_values, demonstrate how we tried to help with the calibration problem. This function is simple, it finds the min value in the temperature Data list (that's that list that holds the CSV values), and then it substructs that value from all the values in the list. Since it's a "two-dimensional" list, the code finds the minimum value of each inner list and compares it to the absolute minimum variable min_value. Once the code finds the absolute minimum value, it iterates over the list and subtracts that value from every element of the list. Then, the code simply returns the modified list.

```
def correcting_values(temperatureData):
#setting initial values so everything would be smaller
min_value = 1000
#goes through the list to find the absolute minimum
for elements in temperatureData:
  temp_min_value = min(elements)
  if temp_min_value < min_value:
    min_value = temp_min_value
# min_value = min(temperatureData)
if DEBUG:
  print min_value
#subtract the minimum value from all values
for row_counter, elements in enumerate(temperatureData):
  for column_counter, element in enumerate(elements):
    element = element - min_value
    temperatureData[row_counter][column_counter] = element
    # [float(i)-min_value for i in elements]
    # print temperatureData[row][column]
#returns the new list with changed values
return temperatureData
```

Listing 2. Python example

Finding the Person's Head

The following function, starting_point, receives the temperatureData list (that's that list that holds the CSV values) and returns a list. The list either holds two values, x and y, that represent the coordinates where the head begins, or it can be empty if the function failed. Y is the inner list number (the row in the CSV) and X is the element number inside that list (the column in the CSV). If the image is bad or if our code had an error, it is possible that the code won't find a starting point. In this case, the returned list would be empty.

The method to find the starting point is the following: the code goes through every inner list (every row) until it reaches an element that is in the right range (¿ 30 and ¡45). Then, the code checks if the next four elements in the list are also in the right range. If there are five values that are in the right range in a row, our code assumes that where the head begins.

```python
  def starting_point(temperatureData):
    for row_counter, elements in enumerate(temperatureData):
      for column_counter, element in enumerate(elements):
        if element < 45 and element > 30:
          good = []
          for iterator in range(1, 5):
            # print len(elements)
            if (column_counter + iterator) < len(elements):
              if elements[column_counter + iterator] < 45 and elements[column_counter + iterator] > 30:
                # print "added to good"
                good.append(iterator)
          # print len(good)
          if len(good) == 4:
            # print "in"
            coordinates = []
            coordinates.append(row_counter)
            coordinates.append(column_counter)
            return coordinates
    return []

```

Listing 3. Python example

Getting the Mean

In order to get the mean temperature value for the head, we had to come up with a method to decide where the head is. Bill allowed us to assume that the head will usually be about the same size. After finding the coordinate where the head “begins”, we assume the head is going to be around 40 columns wide and 30 rows long. These numbers are base on an image of a person who is standing 2 meters away from the camera.

The code uses those values (40, 30) in the following way: It uses the starting point (from the last function) as the top of the head. However, since the head of a person is wider in the center of the face (as shown by the temperatures in the CSV file), we assumed that the top of the head is about the center of the head horizontally. This is why the code goes from y to y+Height vertical but from (x - width/2) to (x + width/2).

There might be better ways to extract the head from the picture. We used this method for simplicity and because we’re only using pixels that are in a specific range. That is, if we go too far from the head it doesn’t matter because those values will probably be too cold and won’t be counted towards the mean.

```python
def get_mean(x, y, WIDTH, HEIGHT, temperatureData):
  if ((y + HEIGHT) > 120 or ((x + (WIDTH / 2)) > 160)):
    print "PERSON OUT OF RANGE"
    return 0
  for row in range(y, (y + HEIGHT)):
    # y is the row it begins, h is the width of the length of the head
```

```
      for column in range(x - (WIDTH / 2), x + (WIDTH / 2)):
        if (temperatureData[row][column] > 43 or temperatureData[row][column] < 35):
          continue
        else:
          good_range.append(temperatureData[row][column])
    if (len(good_range) == 0):
      return 0
    range_mean = sum(good_range) / float(len(good_range))
    return range_mean

```

Listing 4. Python example

Output

In the training state (not in the production mode) the program outputs a CSV file. In the file, there are two entries in every row, one for the calculated mean and one for the name of the processed file. Outputting the name of the files helps with matching the calculated mean value with the thermometer temperature. After processing the CSV folder, the user should add the temperatures taken with a thermometer to this file. After matching the calculated value with the thermometer value, the file name column should be deleted. At that point, the file would be ready to be sent into the model code.

Model

Our Model

Our current model is created by using linear regression, specifically, Least Squares Linear Regression. This method works well for the most part and can be made more robust with larger datasets. However, it does not account for other variables that could alter a person's skin temperature or their core body temperature. This could potentially prove to be troublesome in the long run, so different models should be tested to try and solve those problems.

Other possibility

One possibility for another model to test is a multivariable regression line, where the slope is dependant on multiple different factors. This way factors like weather, ambient temperature, humidity, if the picture is taken indoor or outdoor, windchill, and activity before the picture was taken can all be taken into account. However, this method requires a little bit more thought behind it and may require more user input as they need to input those variables everytime the program is run.

Production Mode

The production mode is the culmination of all the parts of this project. It combines the image processing and the model into one program. The code has two parts, one to create the model and one to take the picture and process it. Creating the model is very simple, it creates a Least Squares Regression Line with the data provided. It is run with the same program, but with a different command. We currently only have a linear regression, but more can be added in at a later time.

The actual production mode where a picture is taken and an estimated core body temperature is produced is ran with a different command. It is run with two different functions within the program. The first function creates a folder called ‘Images’ in the current directory which will be where all the images are created, stored and analyzed as to not clutter up the current directory. Then it takes 40 frames worth of images, but only the last 30 are kept. We do this to prevent any frames where the camera is warming up where it produces garbage frames of data. However, there is a chance that it can produce 40 frames of garbage, so it is up to the user to watch to make sure that doesn’t happen. However there is error checking in the image processing so an error will be thrown.

```
def takeThermalImage(configFile):
  path = "./Images"
  name = "image"
  with IrCamera(configFile) as ir_cam:

    for i in range(40):
      data_t, data_p = ir_cam.get_frame()
      cv2.imshow('visual data', data_p)
      print '--------------------'
      print data_t
      print '--------------------'
      cv2.waitKey(5)

      n_rows, n_cols = data_t.shape
      # print data_t[0]

      if (i > 10):
        filename = str(name + str(i) + ".csv")
        with open(os.path.join(path, filename), 'w') as csvfile:
          thermalwriter = csv.writer(csvfile)
          for row in range(n_rows):
            thermalwriter.writerow(data_t[row])

```

Listing 5. Prodiction Mode

After the first 10 frames, the code will then start to save the data as csv files in the ‘Images’ folder. The second function that the code is the processImage() function in the image processing module.

```
config = sys.argv[1]
modelData = sys.argv[2]

# Taking the image using the first config file.
takeThermalImage(config)

# Getting the means of all the files.
imageMeans = images2.processImages()

if (0 in imageMeans):
  print "An error has occurred. Try again."
else:
  # Getting the mean of all the means
  totalMean = mean(imageMeans)

  # Running the total mean through the model.
```

```
    print "Raw Processed Temperature: ", totalMean
    print "Predicted Core Body Temperature: ", modelPredict(totalMean, modelData)

    print "DONE"

```

Listing 6. Prodiction Mode

The program depends on two command line arguments, the first is the configuration file for the camera and the second is the model. The function that runs the camera requires the configuration file, while the prediction function requires the model information. The piece of code above is the main portion of the program. It runs the function that takes the pictures, then runs the image analysis code we wrote on all the csv files that were produced. The analysis code analyzes the images and produces a mean temperature of the persons head. The variable imageMeans is a list of means created by the code.

Within the image analysis code there is error checking to check for if the person is out of frame, garbage frames or irregular temperature spikes produced by the camera. If any of these are found it produces a 0 and immediately returns. The production mode code then check if there is a 0 within the list of means, and if there is an error is thrown. Otherwise it takes the average of the list of averages and inputs it into the least squares model. The output of the model is then output onto the screen as the predicted core body temperature.

```
def modelPredict(rawTemp, modelDataName):
    modelData = reader.commaReader(modelDataName)
    predictedTemp = (rawTemp * float(modelData[0])) + float(modelData[1])

```

Listing 7. Prodiction Mode

Since we are using a linear regression line the model is a simple linear line. The model data contains a slope and intercept

Command Line

The production mode has two modes. The first being model creation and the second being taking the pictures and analyzing it. These two commands are ran based on the command line arguments passed to the program. Creating the model is ran by simply adding the '-c' tag and the training data. Where training data is a list of number pairs where the first is the skin temperature produced by the camera and the second is the real core body temperature. These two values should be separated by a space. The command should look like this:

Python productionMode.py -c trainingData.txt

To take the picture there is a slightly longer command as there are more arguments that need to be passed. The command should look like this:

Python productionMode.py config.xml model.txt

The first command argument is the configuration file for the camera and the second is the model information for predicting core body temperatures. For our current model we used a least squares regression line, so the model file is simply slope and intercept. More model types can be added in later if needed but we only have a linear regression right now.

Testing

While running the code we ran into some edge cases where errors will occur. These included things like the person not being in frame, the person being too tall or too short, or camera errors such as garbage frames or strange temperature spikes. We tried to account for all of this in the image analysis portion by adding in error handling where we would output an error to the user. There may be even more edge cases we haven't run into yet, but for best results try and have the person's head in the center of the frame.

Problems and Solutions

Camera

Our projects rely on data from the camera. In order to create a good model and even a good image processing process, we needed the camera to be reliable. We ran into many problems with the camera, mainly due to bad calibration. In the beginning, we used the camera software to test the camera. While using the software it was easy to see that the camera returned bad values. In the beginning, it returned about 60 degrees Celsius as the temperature of a person. It also returned about 100 degrees Celsius for a cup of coffee even after it set for a while. We tried a few methods to test the camera, one of them was to test the change of temperature of a cup of coffee on a table. We put a hot coffee cup on a table and took a picture of the table every few minutes.As times passed and the coffee's temperature went down so did the table temperature. That showed us that the camera is easily affected by its surroundings. This little experiment showed us that the camera's problems are worst than we thought.

We also tried to change a lot of the variables in the software such as ambient temperature and adding a reference temperature. None of these helped. There is a folder of a few images with temperature values in the camera folder. It's called Samples. We used that folder as a calibration folder for the camera and it made the camera work better, but it was still a little off. In the production mode, where we don't use the camera's software to take the pictures, the code just uses the default calibration file.

Data

This kind of project relies heavily on data. We ran into many problems with collecting data. First, as mentioned before, the camera returned bad values. We still collected some data but it's not reliable. It was hard to predict how the camera will behave when taking pictures. Another problem, even bigger than the camera problem is the nature of the data. In order to create a model that predicts high fever we needed to have pictures of people with a fever or high body temperatures. This kind of data is hard to collect. As part of this class we weren't allowed to use random people to collect data from. We ended up using only the three of us for collecting data. Bill's graduate student helped us by providing some extra data but it was still of people with normal temperatures.

Results

We were able to create a program which takes a picture at and produces a estimated core body temperature. However our current version may not be accurate as the model may be weak due to the lack of data. As a proof of concept were were successful in making something that could work, if provided with a large data set, and a fully functioning calibrated camera. The current methods for image processing and model creation are somewhat simple, and could be improved upon at a later time. The model is a least squares regression line so it does not account for many other variables that could affect a person's skin temperature and core body temperature. The image processing code is a simple search through a csv to return a batch of values roughly in the area where the person's head is. This works for the most part, but does not account for things like long hair or beards.

Overall we create a simple proof of concept to estimate core body temperature through a thermal image. The overall accuracy of this project is dependant on the data set used, and the complexity of the model. With a larger data set the program could be fairly accurate.

Recommendations for Future

Model

We are currently using a least squares regression line for our model creation. This creates a simple linear line which produces a slope and intercept to correlate a person's skin temperature to their estimated core body temperature. This simple method could be the best method, but we did not much data to create a more robust model. Multiple linear regression is the next logical step to test for the model. It can account for more variables that could possibly change someone's core body temperature. Some of these variables could be the weather, the ambient temperature, and the humidity.

Camera

The camera must be calibrated correctly in order for this project to work. Currently, the code uses the default calibration. We were stuck on this problem through the whole project. We couldn't find a way to get the camera to work correctly. It also wasn't in our requirements to calibrate it. We think that a possible fix to the problem might be to take more pictures and videos with the camera, add to them their temperatures, and use those as calibration files. We weren't able to do this, but it might help increasing the accuracy of the camera.

Data collection

While presenting our project at the undergraduate engineering Expo, we've noticed many factors that could change the camera readings. It might be helpful to pay attention to these factors while collecting data to train the model.

- The distance from the camera: the closest to the camera, the better the reading. People that were standing farther than 2.5 meters away seemed colder in the image.
- The camera's range: When standing 2 meters away from the camera, a lot of people were either too short or too high to fit in the frame. We had to change the camera's angle many times to get the people in the frame.
- Glasses: glasses are really cold on the camera. It might be best to make people take their glasses off for the picture. It might make the mean smaller.
- Hair: hair is colder than the skin. Therefore it might be problematic to run the code on people with a big beard or mustache. It might be helpful to train the model on people with such characteristics.
- Some people are just way hotter than others. It was clear that even though people were in the same conditions, some were really hot even though they didn't have a fever and didn't work out. When collecting data, it might be helpful to collect data on people that are usually colder and people that are usually hotter.

Code Modifications

We are currently using somewhat rudimentary methods for both the model and finding the person's head in the image. Our model is a simple least squares regression line so it simply creates a best fit line based on a person skin temperature and core body temperature. This can be thrown off my many different factors, like if it is cold outside skin temperature will be lower while their core stays the same, or if the person has just exercised their core body temperature will be higher than normal. If a different model is used it should be more robust, and take into account different factors such as the weather, and ambient temperature.

Our image processing code simply scans the csv file, looking for temperatures within skin temperature range then outputs an average of those temperatures. It does some error checking in the production mode and it also tries to isolate the head of the person. However, it does not account for things like long hair, beards, glasses or piercings. These things are colder than human skin and could potentially produce errors with our code. Our code does a decent job, but it could be improved upon by using something smarter like some kind of facial recognition where hair, beards and glasses won't throw off the data gathering.

Conclusions

This project relies heavily on data. As mentioned in this report, we weren't able to collect good data. Both the camera problems and the nature of the needed data made it hard to achieve that goal. We believe that once the camera is fixed and enough data is collected, our code can be used to estimate a core body temperature of a person using a thermal image. There are some code modifications that need to be done. Also, new prediction models need to be tested. But more importantly, a lot of data needs to be collected for this project to succeed.

8 Recommended Technical Resources

9 Conclusions and Reflections

10 Appendix 1: Essential Code Listings

11 Appendix 2

References

[1] C. M., L. P., L. L., and C. H., "Mass screening of suspected febrile patients with remote-sensing infrared thermography: Alarm temperature and optimal distance," http://www.sciencedirect.com/science/article/pii/S0929664609600176, 2008, (Accessed on 11/03/2017).

[2] 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, 2017.

[3] B. Beauchamp, "Bianca's technical review," https://github.com/maimonc/Ebola-Virus-Project/blob/master/Documents/TechReview/Bianca_Beauchamp_Tech_Review/Technology_Review.pdf, 2018.

[4] B. Haung, "Brian's technical review," https://github.com/maimonc/Ebola-Virus-Project/blob/master/Documents/TechReview/Brian_Huang_Tech_Review/Brian_Huang_Tech_Review.pdf, 2017.

[5] "Median," https://en.wikipedia.org/wiki/Median, (Accessed on 1/10/2018).

[6] "Variance," https://en.wikipedia.org/wiki/Variance, (Accessed on 1/10/2018).

[7] "Mean," https://en.wikipedia.org/wiki/Mean, (Accessed on 1/10/2018).

[8] "Linear regression," https://en.wikipedia.org/wiki/Linear_regression#Applications_of_linear_regression, (Accessed on 1/10/2018).

[9] "Polynomial regression," https://en.wikipedia.org/wiki/Polynomial_regression#Interpretation, (Accessed on 1/10/2018).

[10] S. Ray, "7 types of regression techniques you should know!" https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/, August 2015, (Accessed on 1/10/2018).

[11] "Squared or absolute? how different error can be." http://archive.is/0hHpF, September 2013, (Accessed on 1/10/2018).

[12] S. Holmes, "Rms error," http://statweb.stanford.edu/~susan/courses/s60/split/node60.html, November 2000, (Accessed on 1/10/2018).

[13] "Mae and rmse-which metric is better?" https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d, March 2016, (Accessed on 1/10/2018).