Portable Retina Eye Scanning Device

Senior Project Report
Electrical Engineering Program
Sonoma State University

Sumbmitted by:
Cristin Faria and Diego A. Espinosa

Faculty Advisor:
Sudhir Shrestha, PHD

Industry Advisor:
Ben Valvodinos

Client:
North Bay Vision Center

LinkedIn:
https://www.linkedin.com/in/cristin-faria-710010112/
https://www.linkedin.com/in/diego-a-espinosa-687812132/

Website:
https://diabeticretinopathyssu.weebly.com

May 18, 2018

Abstract

Throughout the world there are millions of people diagnosed with diabetes every year, one of them is diabetic retinopathy, a disease of the eye affecting the retina, which can lead to blindness or distorted vision. The issue surrounding diabetic retinopathy is in the diagnosis, there are two common problems, for one, patients generally don't notice a small, steady decline in their vision over time or they simply do not have access to preventative health care.

Multiple companies have proposed or have previously marketed medical devices used to scan for diabetic retinopathy. Our concept improves upon previous as well as current devices because our concept works absent the need for an Internet connection. Current devices must be connected to the Internet in order to implement the program to validate the devices results, because they are run over applications and have to have Internet in order to connect to the online images and algorithms they use.

Within this document, we will discuss the hardware and software models of our proposed Portable Retina Eye Scanning Device, abbreviated as PRESD, its ability to use machine learning and image processing to scan images of retinas with the goal of determining whether or not the imaged retina is healthy.

The PRESD is made up of different components including the Raspberry Pi 3 Model B, an embedded platform that is used to apply the machine learning and image processing concepts into effect, a retinal camera used for taking images of a patient's retinas, an LED box light to display the resulting determination, and battery pack to power the system. The goals of this project are to create a low-cost device which functions independently from the Internet.

Within this documentation, we also discuss key performance factors of our project, including our engineering and marketing requirements. We conclude this paper with our supporting diagrams and display the functionality of our system, and future work.

Table of Contents

Table of Contents
List of Figures 4
List of Tables 5
1. Introduction.......... 6
1.1. Literature Review 7
1.2. Problem Statement 7
1.3. Methodology 8
1.4. Challenges.......... 8
1.5. Marketing Requirements 9
1.6. Engineering Requirements.......... 9
1.7. Customer Survey Results.......... 10
2. Implementation 11
2.1. High Level Architecture 11
2.2. Hardware Architecture.......... 12
2.3. Software Architecture.......... 12
3. Testing 13
3.1. Machine Learning Test.......... 13
3.2. Classifier Test.......... 16
3.3. Accuracy Test.......... 20
3.4. External Camera Test.......... 22
3.5. Outside Image Test.......... 23
3.6. Camera Image Test.......... 24
3.7. Algorithm Accuracy Test.......... 25
3.8. Automation Test.......... 27
3.9. ArduCAM Test.......... 28
3.10. Raspian Test.......... 30
3.11. MATLAB and LED Test.......... 31
3.12. LED Test.......... 32
4. Engineering Ethics and Our Project.......... 33
5. Future Work.......... 33
6. Special Thanks..........34
Appendix A – Detailed Budget..........35
Appendix B – Schedule..........36
Appendix C – Customer Survey..........37
References..........38

List of Figures

List of Figures

Figure 1 - High Level Diagram. 11
Figure 2 – Hardware Diagram. 12
Figure 3 – Software Flowchart. 12
Figure 4 - Sample Images 14
Figure 6 - Bag of Features. 14
Figure 7 - Classifier Training 15
Figure 8 – Prediction. 15
Figure 9 – Confusion Matrix 1. 19
Figure 10 – Confusion Matrix 2. 19
Figure 11 - Webcam 1 22
Figure 12 - Outside Image. 23
Figure 13 - Webcam 2. 24
Figure 14 - Image Classification. 24
Figure 15 - Algorithm Bar Graph. 26
Figure 16 - Test Image. 29
Figure 17 - Health Indicator. 26

List of Tables

List of Tables

Table 1 – Classifier Accuracy Table 1 17
Table 2 – Classifier Accuracy Table 2 18
Table 3 – Classifier Accuracy Table 3 18
Table 1 – Detailed Budget 35
Table 2 – Project Schedule 36

1 Introduction

In the world today, there are millions of people with diabetes who do not have access to medical professionals in the ophthalmology arena or are unable to afford the diagnostic tests that they need to address health concerns, specifically, eye diseases. It is our goal as electrical engineers to build a device that allows physicians and health practitioners to scan a patient's eye in order to detect the presence of a specific eye disease. Successful implementation of our device will enable health practitioners specialized in treating patients with an eye disease to have access to an advanced diagnostic tool that is less expensive, is mobile, and can work independently from the Internet. In terms of eye illness, the project's primary focus will be on Diabetic Retinopathy: a condition caused by damage to the blood vessels in the retina. Uncontrolled high blood sugar levels is the main cause of this form of eye illness, it prevents blood from passing through the blood vessels of the eye, leading to swelling and leakage from the blood vessels into the retina which ultimately causes distorted vision and cell loss to an individual.

The most important aspect of this project is that the proposed device must be accurate enough to determine the current status of a patient's eye. Our goal is to create a device that is portable, and can be used by physicians and health practitioners to practice in rural areas or third world nations where access to diagnostic tools are limited or not available. We intend to make this diagnostic device inexpensive so that more doctors in hospitals and clinics as well as other health practitioners throughout the world can afford to purchase this product and use it to diagnose their diabetic patients.To achieve our aim of making the device available to the developing world, it is crucial that we create a device that is Internet-independent. Due to socio-economic conditions in the poorest nations and rural areas, many places in the world are isolated from Internet connectivity or have little to no access to Wi-Fi, Ethernet, or other Internet capabilities, so our device has to be able to run offline with no issues. Our intent is to ensure the device is simple to use after a short training session. The device must also maintain reasonable power levels, and processing speeds in accordance with today's industry standard for electrical devices. Success in the project means the device successfully diagnoses a diseased eye with a 90 percent accuracy rate. In the next section, we will review some current products that scan the retina of an eye to detect eye illness or help determine the health state of an eye.

1.1 Literature Review

There have already been similar devices to ours that have been invented to scan the retina of an eye. However, in a previously published article, [1] the author proposes a device that uses machine learning and basic algorithms to get positive results. This machine has been made more compact and at a lower price. However, one flaw that this device has is that it must be connected to an Internet-based network to run its algorithms properly. A review of article [2] demonstrates a machine and method that discusses earlier detection of eye disease and how it can affect patients. The article states that many diabetic patients are unaware that their vision is getting worse until it is too late and so it states the desirability of a machine that can track the early signs of the disease in order to save their patients' vision. The article suggests a device that can scan, take pictures quickly and identify the type of disease hence meaning that a more proper medical device should be able to help patients sooner rather than later. In another article [3], a machine is used to look at 50 different images at a time to determine whether or not an eye is unhealthy, healthy or in the process of developing the disease; thus, this device can be used to determine multiple diseases. The main issue present in this device is it does not check enough images at a time while additional problems include are that it is more expensive and lacks the ability to run while offline. The high cost of the device makes it impossible for poorer communities to afford the technology. In yet another article [4], a device is demonstrated that tracks eye disease by tracing signals in the eyes through imaging them. This is similar to our design in a way because ours also takes images of the eye, but varies because it looks at the images differently. This device tracks blood pulses in the vessels rather than the vessels themselves.

1.2 Problem Statement

Modern technology has been able to develop cameras that can scan the retina of an eye and take high-resolution images of it, however, there has not been substantial development towards devices that would do the type of functionality that we are proposing in this design project. Our intent, as discussed earlier, is to allow this diagnostic technology to be easily accessible to physicians and health practitioners to use regions that are far out of reach from a modern health institution or have limited access to electricity or health care. As mentioned before, certain doctors and physicians who work in clinics and hospitals currently have the equipment and testing procedures to scan the retina of a patient's eyes to inspect for any signs of eye disease. However, not every place in the world contains health institutions with this type of technological equipment that can help all patients determine the health state of their vision. Having a portable device that scans for Diabetic Retinopathy will help raise self-awareness to individuals who don't have access to health care that would help them determine this. The main objective and functionality of the Portable Retina Eye Scanning Device is to scan a patient's eye to determine whether or not the eye is diagnosed with the disease Diabetic Retinopathy, thereby enabling physicians to quickly determine the health state of their patient's eyes related to the disease. The main problem in this project is to determine how to create a device that is less expensive and can be used regardless of Internet connectivity. For the people who suffer from Diabetic Retinopathy, the main problem is that the disease is hard to initially diagnose and manifests itself often without the patient being aware. Usually it is only after the patient's health starts to deteriorate is when the disease is recognized. Unfortunately, the most obvious symptom is a gradual decline in vision, which can ultimately lead to blindness. Due to a lack of awareness, this disease causes millions of people to go blind every year article [8]. The patients who are most likely to suffer this disease are often the patients who simply do not have the access to the current technology or preventative health care. It is our firm belief that people should not go blind because they don't have access to health care, live too far away, or are poor. We believe the creation of this device will be a step closer in enabling those patients who desperately need preventative treatment access to the technology and care they deserve. If successful, we will market this device towards physicians and health practitioners who are specialized in diagnosing eye illness. Additionally, the device will also be marketed towards firms who are involved in supplying cheap, portable devices to hospitals and clinics that reside in rural regions around the world, such as villages along the coast of South Africa.

1.3 Methodology

Our idea is to implement machine learning on this project by using the Raspberry Pi 3 Model B graphical user interface with a retinal camera that will scan images of patients' retinas, and use the presence of damage in the blood vessels to determine whether or not the patient has Diabetic Retinopathy. A major aspect of this project will be connecting certain peripherals to the Raspberry Pi platform, including an LED box light to display whether the eye is healthy or unhealthy, which will be determined by the image comparison algorithm done through the MATLAB Machine Learning Algorithm. The image and subsequent result will then be updated into the database of the MATLAB algorithm and by applying machine learning to the system, additional images and results will increase the accuracy of our device, as it will run a more complete algorithm. The final aspect to consider in order to complete this project is to determine how much power the Raspberry Pi needs to run on its own, so we can determine the battery levels needed in order to maximize battery life and to make our device portable over an extended period of time for the user.

1.4 Challenges

Our first challenge towards implementing our project was learning how to work with the Raspberry Pi 3 Model B. As a function of understanding the Raspberry Pi we developed our machine-learning algorithm. Initially, we began development using the Ubuntu Operating System, and MATLAB, however, we learned through operational testing the Ubuntu Operating System was unable to support the camera and MATLAB. After analyzing the results and doing more research concluded the need to switch to the Raspian Operating System.

Another challenge that we faced was getting clear images of retinas and being able to discern the subtle differences between healthy and unhealthy retinas. The microscopic damage to the blood vessels, which includes bubbling and splintering in the vessels within the retinas, can be extremely difficult to identify. To combat this issue our device will have to be highly accurate at detecting differences in images so that a physician or health practitioner would be able to get the correct results.

Power consumption was an issue because of the volume of data that the processor was required to analyze significantly drained the battery life. Our initial idea to overcome this obstacle was to develop a method to allow the device to go directly into sleep mode right after the algorithm fully executed. Our solution was to add code into MATLAB which produced the required action, thus preventing unnecessary expenditure of power, and increasing battery life.

1.5 Marketing Requirements

MR.1: The device will be reliable in determining the health status of a picture of a patient's retina.

MR.2: The device will be powered by a battery pack.

MR.3: The device will not weigh more than 2.5 pounds. The computer involved is not included in this measurement.

MR.4: The device will be neatly presented with all electrical components protected from the user.

MR.5: The device does not need to be fully automated, but will be simple to use.

MR.6: The device will have significantly low power consumption.

MR.7: The device will capture pictures of a retinal fundus images with high resolution.

MR.8: The device contains enough storage space to maintain all programs and peripherals used by the Raspberry Pi 3 Model B. It will not store the images it takes on the Raspberry Pi so it does not take up necessary storage space.

MR.9: The device will be marketed toward trained physicians and health practitioners involved in treating patients living in rural regions with limited access to ophthalmology testing technology throughout the world.

MR.10: Users will have option to extract photos taken by the camera from the Raspberry Pi 3 Model B.

1.6 Engineering Requirements

ER.1: The accuracy of the scanning must be over 90 percent in order for the device to be reliable in determining the status of the picture.

ER.2: A battery pack will power the device for a minimum of 6-7 hours of continuous use.

ER.3: The device will consist of four components: Raspberry Pi 3 Model B, ArduCAM (camera), Battery Pack, and LED box light.

ER.4: The project will have four to five visible components, including: computer, Raspberry Pi, ArduCAM, Battery Pack, and LED box light.

ER.5: The Raspberry Pi is the main component consuming power, using 5 V and a minimum current value of 2 mA. The maximum current is 2.5 mA.

ER.6: The camera used will have a resolution of 1280x720.

ER.7: The Raspberry Pi 3 Model B has a maximum RAM of 1GigaByte.

ER.8: Functionality of the device will be a precursor to taking a scan of a patient's retina, a practice done by licensed individuals.

ER. 9: Photos will be saved in folders on the computer.

1.7 Customer Survey Results

We interviewed five different Optometrists and concluded many of their patients discover during their yearly routine eye exams they have diabetic retinopathy. Due to a lack of diagnostic screening many patients did not know about the disease or how the disease impacts their vision. Optometrists we interviewed agreed they are in need of better technology that is able to check for this disease and get results to them in a faster and more reliable manner. Opthamologists who commonly diagnose the disease also agreed in more rural regions in the world, there is a need for improved technology, which can run offline without an Internet connection. A common complaint from the doctors was the expense involved in testing and the fact that people in lower income areas are not able to afford these expensive tests. A final concern brought to our attention was the need for portability and ease of use.

2 Implementation

2.1 High Level Architecture

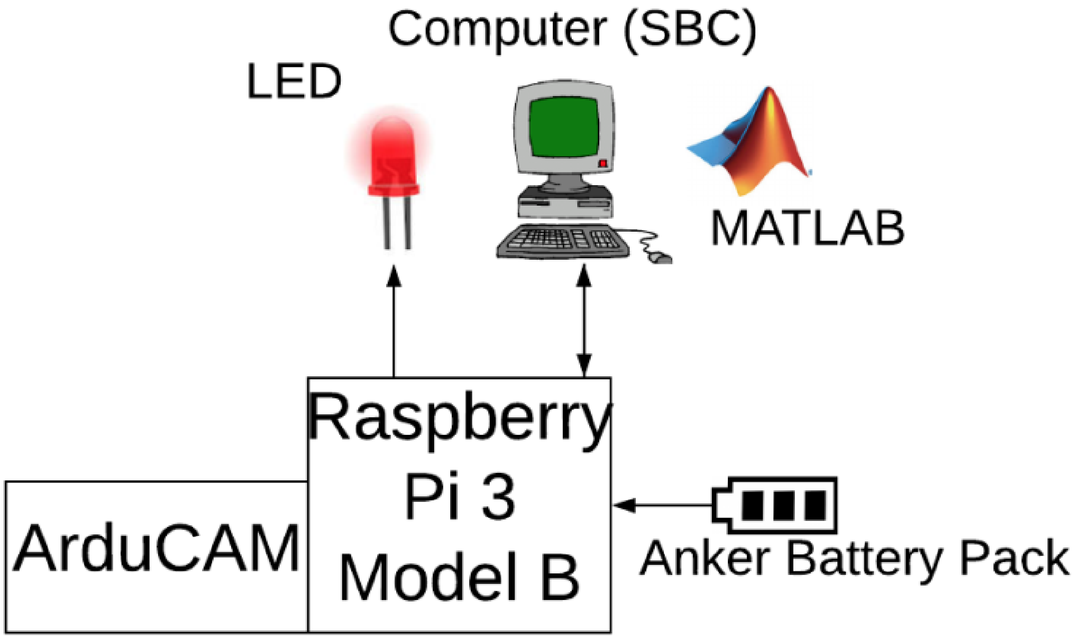


Figure 1: High-level Diagram

Figure 1, is our high-level diagram which gives a basic overview of how our project is able to function. It first sees a retinal fundus image with the camera and takes a picture using the MATLAB algorithm, and based on the result either displays healthy or diabetic retinopathy on the computer screen, and also gives a result by either turning on or off the LED to give a fast indication of the health state of the image.

2.2 Hardware Architecture

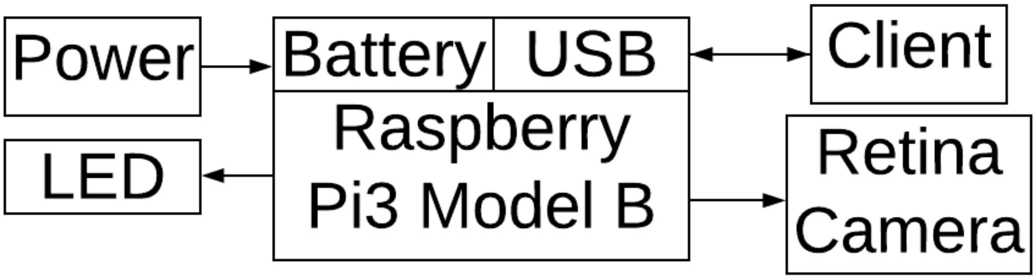


Figure 2: Hardware Diagram

Figure 2, is a more in depth version of our hardware, It states how the hardware components are connected and gives a more in depth version of how they interact with each other. It also shows how the information is sent between the different project components.

2.3 Software Architecture

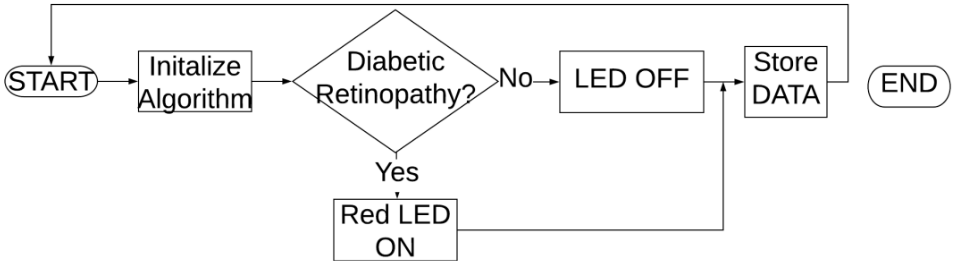


Figure 3: Software Flowchart

Figure 3 is where we explain the functionality of the software in our system. To start we have to initialize the algorithm in order to utilize our machine learning functionality. Then the program checks the image to see if it has diabetic retinopathy. If it does it displays Diabetic retinopathy and turns the health indicator on. If not, it displays healthy and turns the health indicator off. It then stores the result on the computer, and starts the program over again without the need to initialize the algorithm.

3 Testing

3.1 Machine Learning Test

Completed Test: January 25, 2018 Machine Learning Test (MR. 1 and ER. 5):

We were able to use MATLAB to develop our first machine learning algorithm.

Challenges:
We initially faced a large challenge due to the large sets of images which were 60 images in total. The large volume caused a significant time delay of six hours due to the extra data running through the algorithm and features the algorithm looked for in each individual image. In order to overcome the challenges we did the following:

1. We first saved all of our files and images inside of our P.R.E.S.D. folder, and inside the folder we had another folder called RetinaImages that contained the folders with images of Healthy, DiabeticRetinopathy, and Glaucoma eye scans.
2. We labeled our image set which is a 1x3 matrix, and initially contained 45 images from three different class.
3. We created a bag-of-features that selected 80 percent of the strongest features from the photos. Formed them into various clusters, and initialized and completed the clusters.
4. We encoded diagrams, which had a visual representation of each retina type, and paired it with a graph that displayed the visual occurrences and frequency of these occurrences.
5. We trained the image classifier Linear Support Vector Machine for each category , and then evaluated it in order to test the classifier.
6. We evaluated our image classifier and found that our average accuracy was at 67 percent.
7. We ran the prediction test, tested a random image in our files. The result came out correctly and occurred in less than a minute.

Results:

From this test we concluded that we will have to try different classifier types to see which one gives the most accurate result. Also, we will need to increase the amount of pictures in our algorithm in order to make it more reliable.

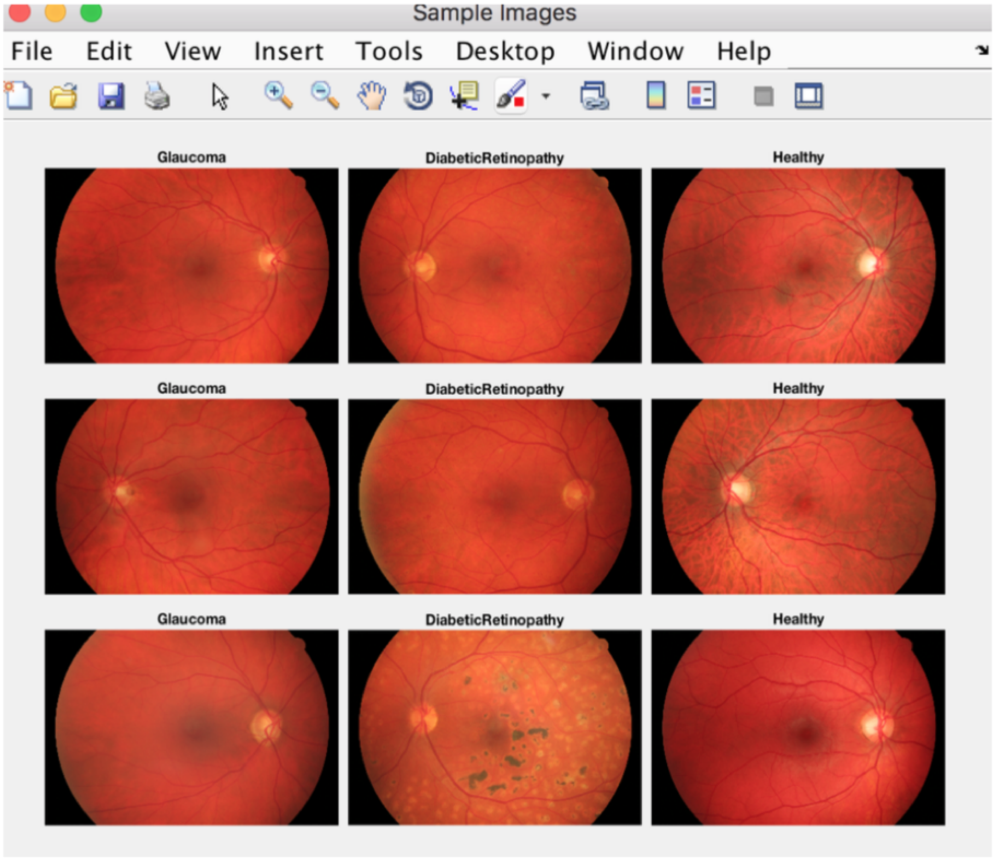


Figure 4: This window displays a group of sample images and groups them based off of the file they are stored in.

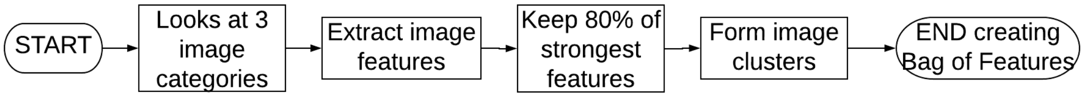

START
Looks at 3 image categories
Extract image features
Keep 80% of strongest features
Form image clusters
END creating Bag of Features

Figure 5: Flowchart: Bag of Features

Figure 5, this flowchart explains the functionality of the Bag of Features, and gives an idea of how machine learning is used in order to categorize the images, and display the results based off of the original groups. It extracts the most important features in order to get the best results, and it is part of how we were able to increase the accuracy of our algorithm and later how we received the results needed to produce an accurate result.

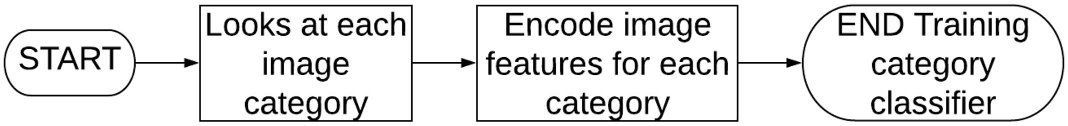
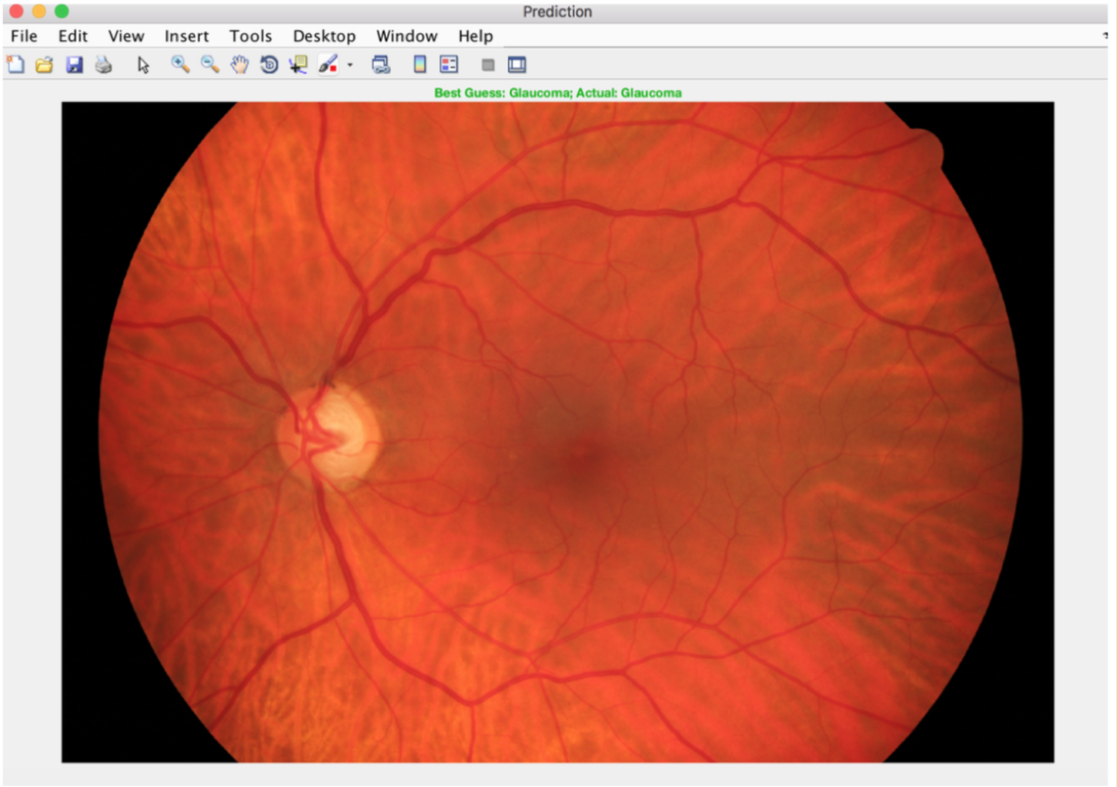

START
Looks at each image category
Encode image features for each category
END Training category classifier

Figure 6: Flowchart: Classifier Training

Figure 6, is a flowchart that demonstrates how the algorithm trains classifiers in order to receive correct results from the algorithm. In order to do this, it much look at each image category in order to find specific features in each one and to have the algorithm check for similar features. It then encodes these features in the algorithm, which causes the algorithm to get smarter because it looks at very specific parts of different images in order to differentiate between them. Finally, the classifier is trained based off of the different features and makes it so the algorithm can check for these features and give more conclusive results based off of the training it received.



Best Guess: Glaucoma; Actual: Glaucoma

Figure 7: Prediction Image

Figure 7, is the window we received that predicted the category in which the image would be placed. It gave us its best guess which was correct, and showed us that the algorithm and code did indeed work the way we wanted it to.

3.2 Classifier Test

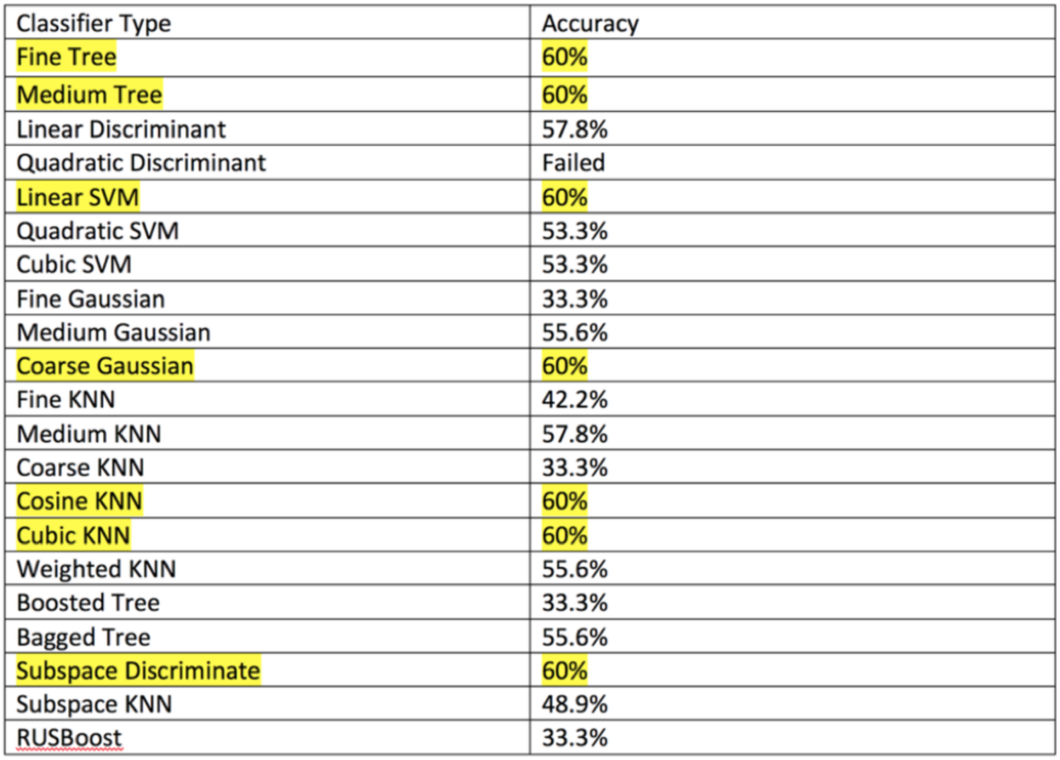
Completed Test: February 1, 2018 Classifier Test (MR. 1, ER. 1 and ER 5):

We tested different types of classifiers to determine which classifier gave us the highest accuracy, used minimal time, and performed the best in all three categories we tested.

Challenges:
The challenge we faced during this part of the project was using the Classification Learner, a classifier application of MATLAB, because we had never used this application previously, and it was hard to understand what needed to be tested initially. In order to overcome the challenges, we did the following:

1. First, we had to run our machine learning algorithm, set up the bag of features and initialize the files we would be utilizing in the Classification Learner.
2. We ran the Classification Learner application in the program and made a new session by picking the correct file from the workspace.
3. We started to train the classifiers by selecting all, and putting each classifier into their own individual Confusion Matrix. It does this by running individual tests on images it knows the results for and is able to give you the accuracy of each individual classifier.
4. Once each classifier was trained we were able to see how accurate each classifier was and put them into a table to compare their accuracy rates. The highest accuracy we received was 60 percent.
5. Eight different classifiers had this level of accuracy so we had to do more comparison tests.
6. We then looked at the Positive Predictive Values and the False Discovery Rates, to find the percentage of false positives.
7. Then, we looked at the True Positive Rates and False Negative Rates, in order to see the summary of true values in each class.
8. From all of these different tests we were able to conclude that the Linear SVM Classifier was best suited for our project.

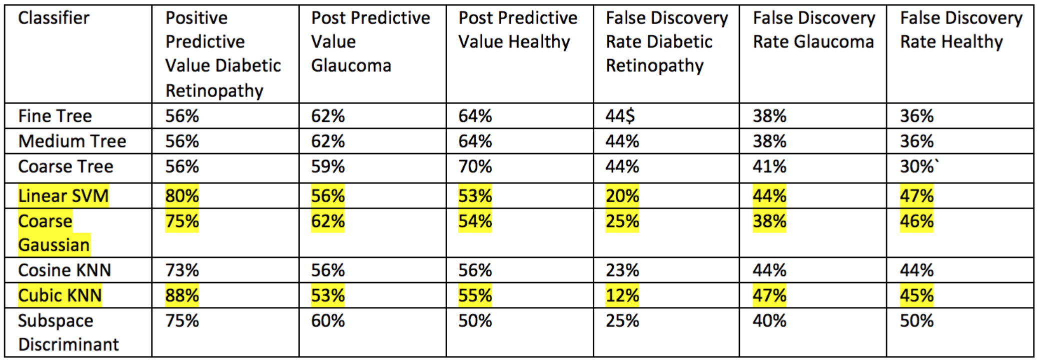
Results:



Classifier Type	Accuracy
Fine Tree	60%
Medium Tree	60%
Linear Discriminant	57.8%
Quadratic Discriminant	Failed
Linear SVM	60%
Quadratic SVM	53.3%
Cubic SVM	53.3%
Fine Gaussian	33.3%
Medium Gaussian	55.6%
Coarse Gaussian	60%
Fine KNN	42.2%
Medium KNN	57.8%
Coarse KNN	33.3%
Cosine KNN	60%
Cubic KNN	60%
Weighted KNN	55.6%
Boosted Tree	33.3%
Bagged Tree	55.6%
Subspace Discriminate	60%
Subspace KNN	48.9%
RUSBoost	33.3%

Table 1: Classifier Accuracy Table

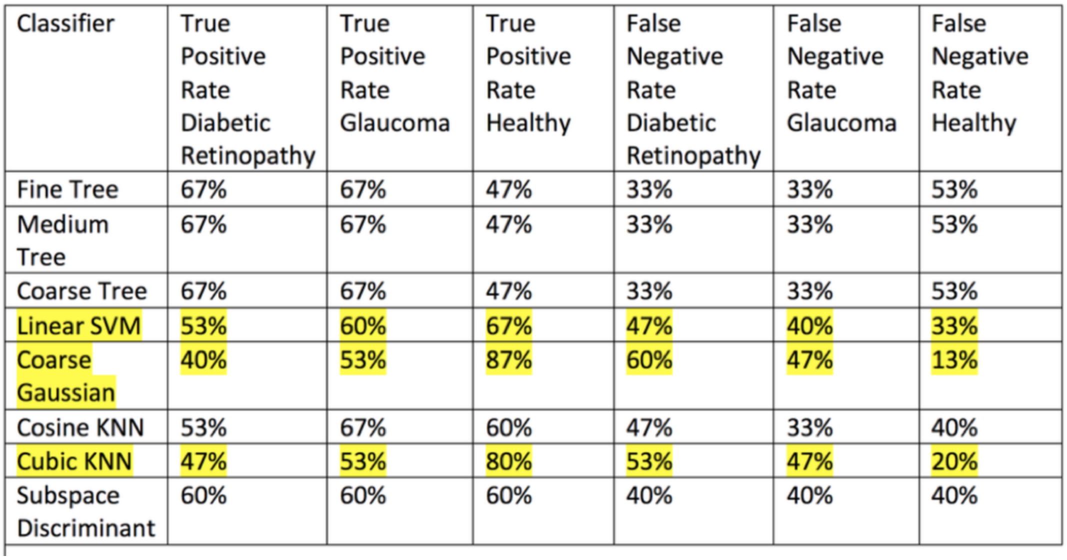
Table 1, was where we looked at the confusion matrix of each classifier which gave us an average accuracy of the correct values the classifier was able to produce. This table looked at each classifier tested, which was twenty-one different classifiers, and the ones highlighted were the seven which gave us the highest overall average accuracy.



Classifier	Positive Predictive Value Diabetic Retinopathy	Post Predictive Value Glaucoma	Post Predictive Value Healthy	False Discovery Rate Diabetic Retinopathy	False Discovery Rate Glaucoma	False Discovery Rate Healthy
Fine Tree	56%	62%	64%	44$	38%	36%
Medium Tree	56%	62%	64%	44%	38%	36%
Coarse Tree	56%	59%	70%	44%	41%	30%`
Linear SVM	80%	56%	53%	20%	44%	47%
Coarse Gaussian	75%	62%	54%	25%	38%	46%
Cosine KNN	73%	56%	56%	23%	44%	44%
Cubic KNN	88%	53%	55%	12%	47%	45%
Subspace Discriminant	75%	60%	50%	25%	40%	50%

Table 2: Post Predictive Value and False Discovery Rate Table

Table 2, was used in order to check the post predictive values and false discovery rates of the classifiers with the highest accuracies. Linear Support Vector Machine, Coarse Gaussian and Cubic K-nearest Neighbor were the top three because all three post predictive values were above 50 percent.



Classifier	True Positive Rate Diabetic Retinopathy	True Positive Rate Glaucoma	True Positive Rate Healthy	False Negative Rate Diabetic Retinopathy	False Negative Rate Glaucoma	False Negative Rate Healthy
Fine Tree	67%	67%	47%	33%	33%	53%
Medium Tree	67%	67%	47%	33%	33%	53%
Coarse Tree	67%	67%	47%	33%	33%	53%
Linear SVM	53%	60%	67%	47%	40%	33%
Coarse Gaussian	40%	53%	87%	60%	47%	13%
Cosine KNN	53%	67%	60%	47%	33%	40%
Cubic KNN	47%	53%	80%	53%	47%	20%
Subspace Discriminant	60%	60%	60%	40%	40%	40%

Table 3: True Positive Rate and False Negative Rate Table

Table 3, was used in order to check the True Positive Rate and False Negative rate of the top seven classifiers, and concluded that from the top three stated above, Linear Support Vector Machine would be the best option because all of the False Discovery Rates and False Negative rates were below 50 percent, and it had the high average accuracy.

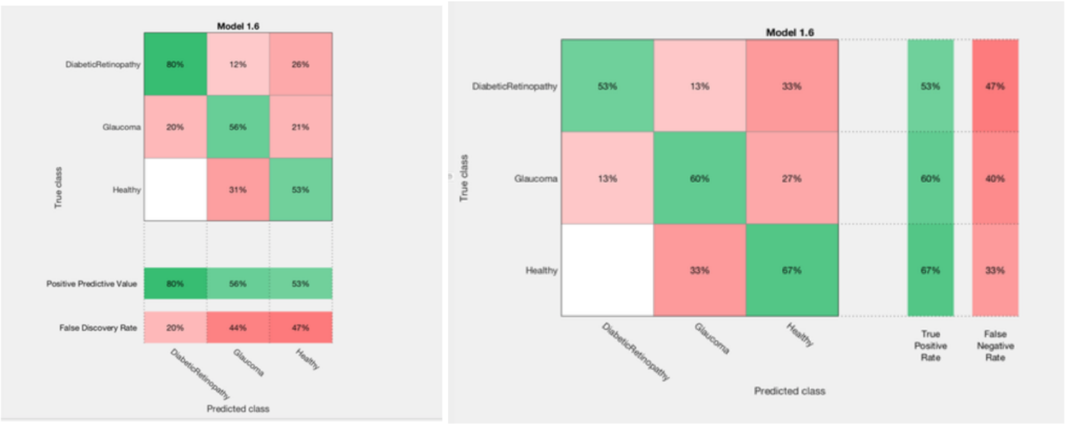


Figure 8: Linear Support Vector Machine Confusion Matrices

Figure 8, these two images are used in order to explain in more detail and show how the confusion matrices work, and how the Post Predictive Values, False Discovery Rates, True Positive Rates, and False Negative rates are visualized in the program. The Post Predictive Rate, and False Discovery Rate are displayed below the confusion matrix. Whereas the True Positive and False Negative Rates are displayed to the right of the confusion matrix. The values which were most significant to our project were the True Positive and False Negative Rates, because we needed our true positive rates to be much higher than the false rates, in order to make a reliable system which would not give us untrue values.

3.3 Accuracy Test

Completed Test: February 8, 2018 (MR. 1, ER. 1 and ER. 5):

In this test we eliminated the glaucoma group and increased the number of images to be applied to the algorithm in order to increase our accuracy.

Challenge:
The main challenge we faced during this test was getting our accuracy up to over 90 percent and making sure we get the most accurate results possible. In order to overcome the challenge, we did the following:

1. First we removed the glaucoma part of our program in order to increase our accuracy. Unfortunately, it only increased to around 80 percent, thus we were not at our target goal.
2. Next we included more images to the DiabeticRetinopathy folder to increase the accuracy to over 90 percent.
3. After increasing the amount of photos to 45 in the folder, we found that we could get our Linear SVM accuracy to 91.7 percent.

Results:

The results we received were that Linear Support Vector Machine gave us a 91.7 percent accuracy rate, and had a prediction speed of 480 objects per seconds, and it took .58022 seconds to train the classifier in order to get these results.

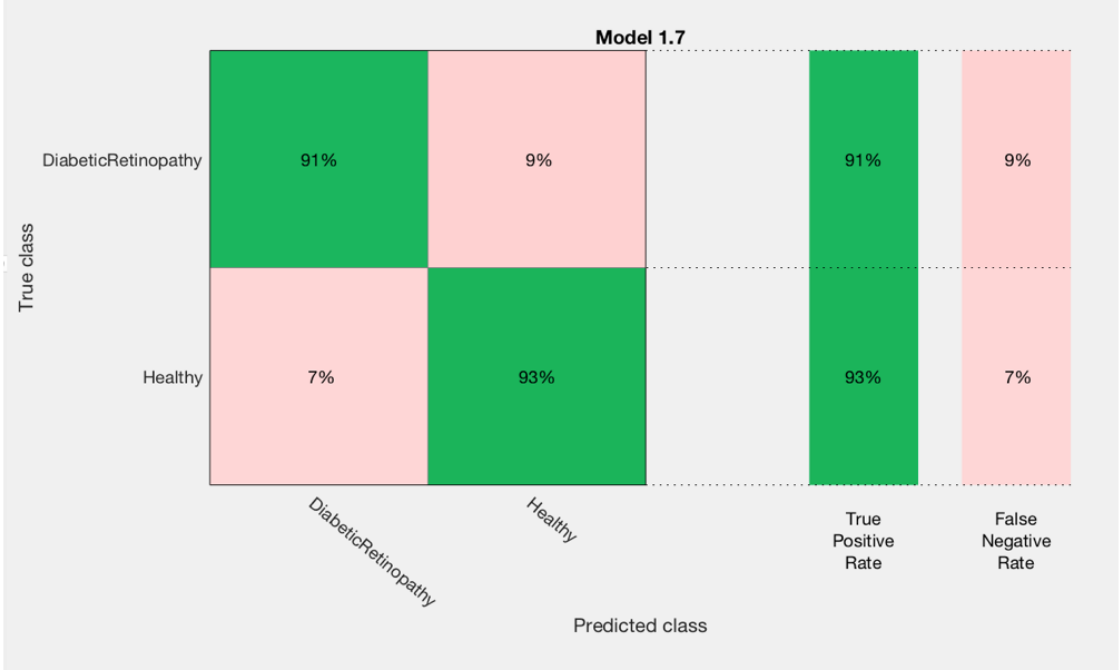


Figure 9: Linear Support Vector Machine Confusion Matrix

Figure 9, in this confusion matrix we were also able to look at the True Positive and False Negative Rates for both categories, and were able to conclude both True Positive Rates were over 90 percent, and their False Negative Rates were below 10 percent, which means overall our accuracy was above 90 percent, and gave us the product accuracy we were looking for, and helped meet our engineering requirements.

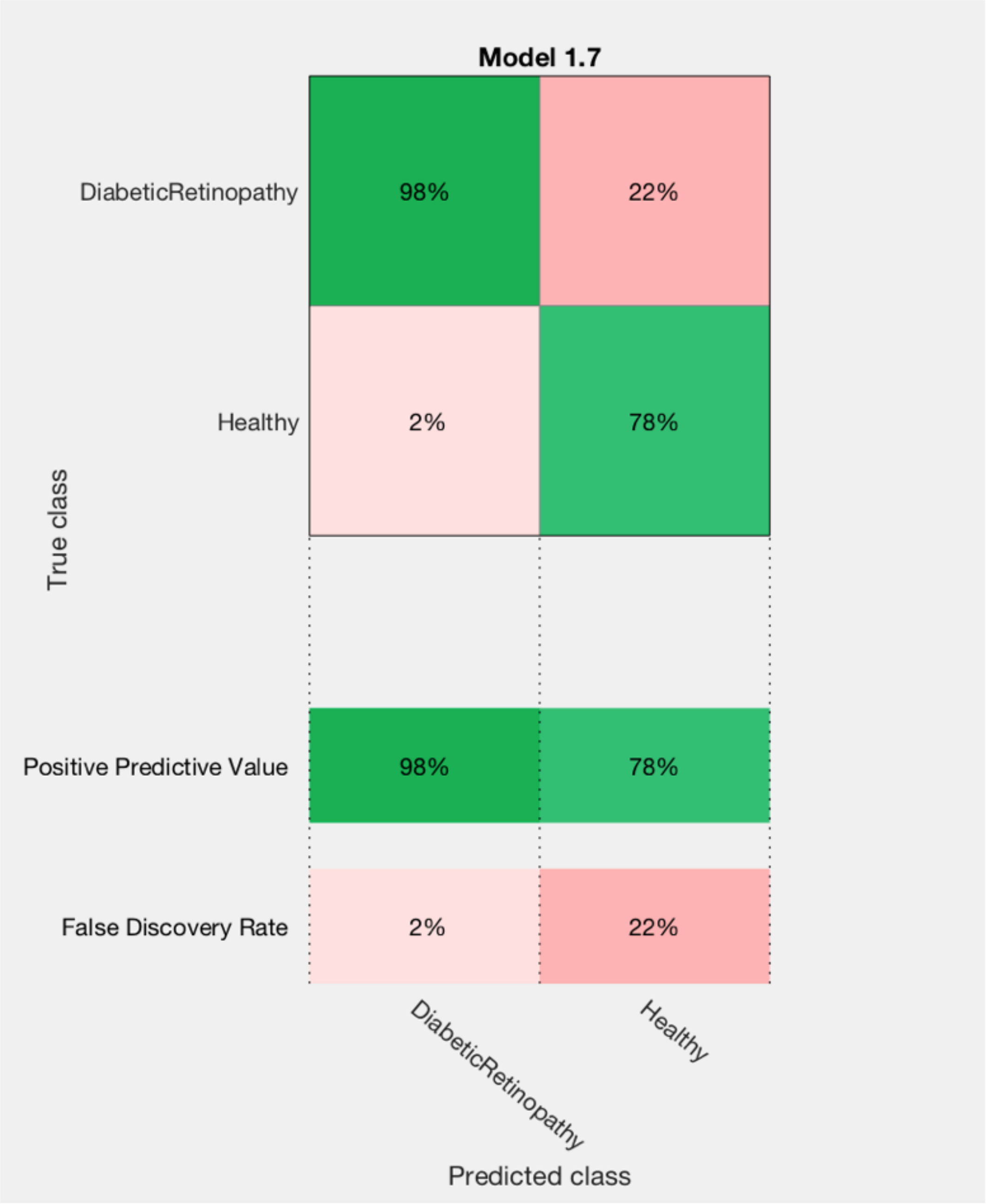


Figure 10: Linear Support Vector Machine Confusion Matrix

Figure 10, in this confusion matrix we looked at the Post Predictive Values and False Discovery Rates of the Linear Support Vector Machine. From this we received a very high percentage for diabetic retinopathy, which is very important because we do not want to tell someone they are healthy when they are not, whereas healthy was far lower.

3.4 External Camera Test

Completed Test: February 15, 2018 (MR. 7, ER. 7, ER. 8, and ER. 10):

In this test we took an image using an external camera and sent it to a folder.

Challenge:

We had a few different challenges during this test. The first one was getting the correct toolbox so that we could access the webcam using MATLAB. The next issue we had was using the snapshot command and getting the camera to focus, we realized we would have to take more than one image in order to get the results we wanted. The next was saving the image and putting it in the correct folder took some time, but once completed was a simple task. In order to overcome the challenges, we did the following:

1. First we utilized the webcam list command to find out what cameras are connected to the device running the program.
2. Next we utilized the webcam command and set it to the camera we wanted to use to take our images.
3. After that we checked for the available resolution of the camera so we could see what type of image we would be able to take with the camera connected.
4. We then created a for loop.
5. Inside the for loop we used the snapshot command to take pictures using the camera we selected.
6. Next, we used the imshow command so that we could see what image the camera was taking and decide what is usable.
7. Finally we used the imsave command to choose a location to save the image and what to label it as.

Results:

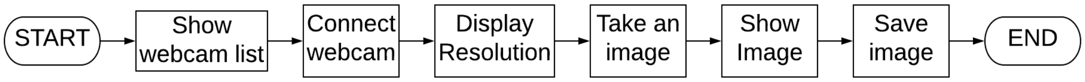


Figure 11: Flowchart: Webcam

The result from the completion of this flowchart was that it displayed the camera we used, found the resolution of the camera being utilized, and told us the available resolution of the camera. It then took an image an told us the image resolution. This allowed us to know the information of the camera and the amount of pixels found in the images. It next took an image using the camera we selected, showed us what the image looked like on the screen, and then saved the image in the folder we selected, and finally ended the program.

3.5 Outside Image Test

Completed Test: February 21, 2018 (MR. 7, ER. 8 and ER. 10):

In this test we took an outside image, test it for diabetic retinopathy, and get the correct result.

Challenge:

The biggest challenge we had during this test was setting up the correct image path in order to access the image we wanted to test. Then the other issue was making sure we used the correct classifier in order to get an accurate result.

1. First, we had to use the imread function in order to select the image we want the program to read.
2. We next used the predict function in order to utilize our classifier and classify the image we acquired in our program.
3. We then used the label command in order to display the label given and to determine whether or not we were given the correct output from our algorithm.
4. In our case we gave it an image that we knew had diabetic retinopathy and it gave us the correct result at the output.

Results:

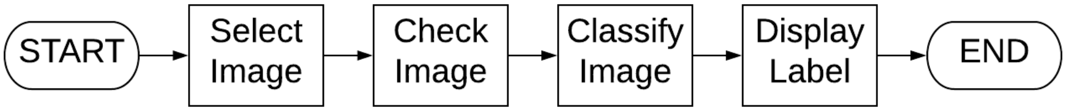


Figure 12: Flowchart: Outside Image

Figure 12, In this test we selected an image from a specific file, and checked the image for diabetic retinopathy. The image was then classified, and the result was displayed in the MATLAB window. We checked an image for Diabetic Retinopathy we knew had the disease. It ended up displaying the result Diabetic Retinopathy which was the correct answer and proved our algorithm and code worked, and was able to give us the correct result using the algorithm and code given.

3.6 Camera Image Test

Completed Test: March 1, 2018 (MR. 7, ER. 7, ER. 8, ER. 10):

In this test we were able to take a photo of a retinal fundus image, take a picture of it with the computer, store the image, and then run it through our code and get accurate results.

Challenge:

The big challenge we faced during this test was making sure the camera was taking a straight and clear shot of the image in order to get the most accurate result. There wasn't a big delay between photos so you have to be accurate and quick when taking images. In order to overcome the challenge, we did the following:

1. So, we first took an image to test and make sure our camera is functioning properly.
2. We took a second image of the images we have which are test pictures that we know have diabetic retinopathy.
3. We then stored the image in a file.
4. We then ran the image file in the code to test for diabetic retinopathy.

Results:

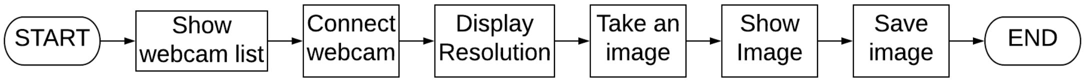


Figure 13: Flowchart: Webcam

Figure 13, we started this program by looking at the list of webcams connected to the device, and selecting the webcam we wanted and using MATLAB to connect to it, we then checked the display resolution in order to make sure it had the correct resolution. Then we took an image and displayed it using the MATLAB window and saved the image to a file we selected.

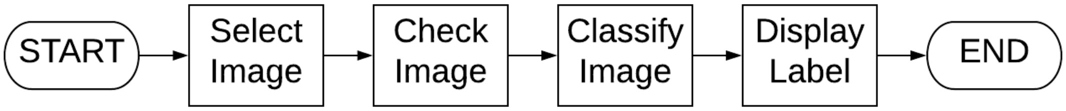


Figure 14: Flowchart: Image Classification

Figure 14, from this flowchart we were able to explain how the program classified images. It did so by first selecting the image it wanted to test, it then checked the image to see which category it best fit into. After doing this it classifies the image and displays the category it belongs to in the MATLAB window.From this thid we were able to receive an answer which stated that our image we tested had Diabetic Retinopathy. This test was important because it proved that our camera function worked as long as it was given a clear enough image and was able to give us an accurate result.

3.7 Algorithm Accuracy Test

Coompleted Test: March 8, 2018 (ER. 1 and ER. 9)

In this test we tested 60 different images. 30 were images with diabetic retinopathy and 30 were images of healthy retinas. We then took our results, calculated the accuracy of both types and found the average accuracy between the two groups.

Challenge:

The main challenge we faced during this experiment was taking clear images, and making sure that we received an accurate result from the image taken with the camera. In order to overcome the challenge, we did the following:

1. Initially we had issues taking clear images with our camera, we overcame this by changing the distance and angle of the picture to make it clearer.
2. The next thing we had to do was make sure we had our images properly labeled so we knew which images had diabetic retinopathy and which were healthy to insure we were be given the correct results.
3. Another thing we did in order to overcome camera issues was put the camera function in a for loop in order to make sure we were able to take a proper image. 4. Finally, in order to get best images possible we checked each image in the display window and adjusted the brightness in the room in order to reduce the amount of glare on each image.

Results:

1. We ran 30 images of healthy retinas through our algorithm, which resulted in 25 correct results and 5 incorrect results.
2. We then found the percentage of correct results which was 83 percent.
3. We next ran 30 images of retinas effected by diabetic retinopathy through our algorithm, which resulted in 29 correct results and 1 incorrect result.
4. We then found the percentage of correct results was 97 percent.
5. We then added up all of the correct results in both groups, and found that we had 54 correct answers and 6 incorrect answers.
6. We then found the average percentage of correct results was 90 percent.

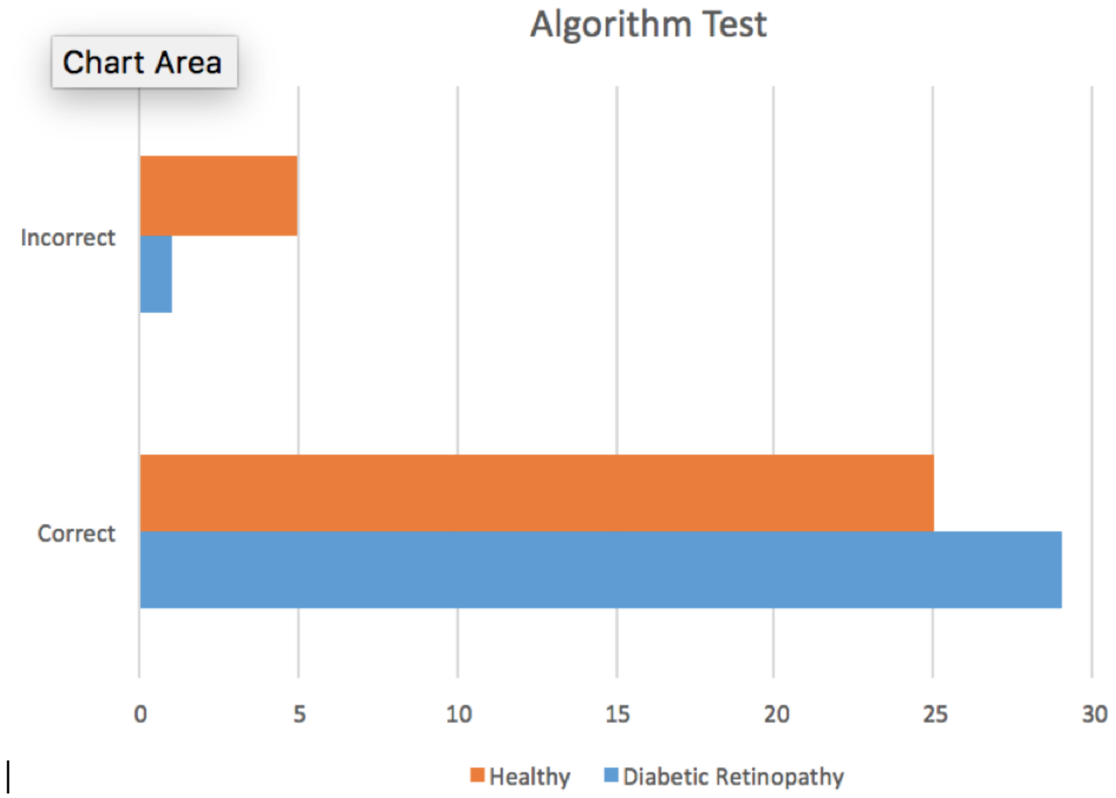


Figure 15: Algorithm Bar Graph

This bar graph shows us the total number of correct images versus the number of incorrect images for our two groups. Healthy, had 25 correct images and 5 incorrect images, and diabetic retinopathy had 29 correct images and one incorrect images. When you add the two categories together you receive an average accuracy of 90 percent which was the goal accuracy to receive from our system.

3.8 Automation Test

Completed Test: March 28, 2018 (MR. 5 and ER. 10)

In this test we automated taking pictures, saving them and testing them with our algorithm. So we were able to take images using an external camera in MATLAB, choose a name, file type and location to save the image. Then used the name and location of the image to test to see whether or not the image has diabetic retinopathy.

Challenges:

The first challenge that we faced was making sure we chose the right camera, file location and file name. Without these being carefully addressed we would be unable to get the picture to be taken. The next phase we had to figure out was how to get the for loop to end so that we could disengage the camera after it had been used. In order to overcome these challenges, we did the following:

1. Set a counter outside of my for loop, this was used in order to count the number of iterations and was set equal to zero initially.
2. The saveas is implemented in the foe loop to choose the name that you want to save it as and location.
3. The counter is increased by one.
4. An if statement is implemented that sets the count as less than two, continue otherwise it ends.
5. After the if statement use clear function to clear camera which allows us to clear the camera that was just used so we can use it again if needed.
6. The imread function selects the folder read from and the file name.
7. The predict function predicts whether the image is Healthy of has Diabetic Retinopathy.
8. Finally, the Label of the image is displayed in the MATLAB window.

Results:

So, from this test we were able to conclude that we could take an image, automatically save it, test it, and display the results in a matter of seconds.

This section of our system was edited in order to get our program to take the images. We had to add in an if statement, and utilize a counter to get the code to run properly and produce an accurate result. After the if statement we were able to select a file to test, had our algorithm make the prediction and show the label it gave the image in the display window. So it was able to automatically take an image, store it, and display the result. The program then ended itself and cleared the camera in case the program needed to be run yet again.

3.9 ArduCAM Test

Completed Test: April 1, 2018 (MR. 7, ER. 3, ER. 4, and ER. 7)

This test was to use the ArduCAM to take an image with the Raspberry Pi3 Model B. This test will help us in the next aspect of the project where we use MATLAB on the raspberry pi in order to take an image of a retinal fundus image using the ArduCAM.

Challenges:

One challenge to tackle in the hardware aspect was to connect our camera to the Pi in order to use it to take pictures of retina fundus images, and be sent to the Raspberry Pi Model 3B for image processing. After correctly connecting the camera, Raspberry Pi, and using the right Terminal commands to access it, an error arose saying that the camera was not detected thus another challenge was to troubleshoot this error in order for the camera to work properly. To overcome these challenges, we did the following:

Initially we interfaced the ArduCAM, the type of camera used, to our Raspberry Pi with the operating system of Ubuntu Mate.

1. Connected the camera by attaching the camera's wires to the ISP of the Raspberry Pi.
2. Boot the Raspberry Pi into it's Operating System, Ubuntu Mate.
3. Open the system's settings usingsudo raspi-config.
4. An error occurred where Terminal said it didn't recognize the command. The issue with this is that we needed to install configuration settings of Raspberry Pi for the operating system. The command to do this wassudo apt-get install raspi-config rpi-update.
5. Now from using the sudo raspi-config, the Pi's configuration menu opens up listing options for the user to configure.
6. Select Enable Camera option, select yes for enabling camera, and then reboot.
7. Another error occurred stating the camera was not detected. After searching for some intel on the net, we realized that certain changes must be made on the Pi's boot/config.txt file.
8. From using the command raspistill-o image.jpg, the camera had successfully taken a photo.

Results:

The result of this is that the camera is now interfaced with Raspberry Pi and the users, as we can now take photos using the terminal command.

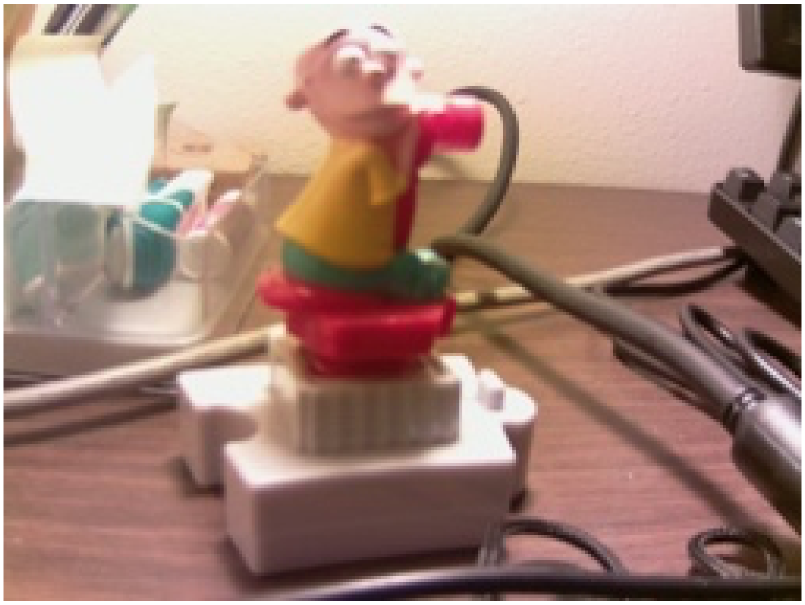


Figure 16: This is an image that was taken with the ArduCAM on the raspberry pi, proving that it is possible to use it to take an image with an external camera.

3.10 Raspian Test

Completed Test: April 6, 2018 (MR. 8, MR. 10, ER. 2, ER. 5, ER. 6 and ER.8)

The reason for the test was so we could work with an Operating System that has better compatibility with the MATLAB support packages for the Raspberry Pi. It's also the default Operating System of the Pi, thus made it easier for us to attain assistance from the Internet and other peers.

Challenges:

At first our operating system was Ubuntu Mate, but after having issues in installing the MATLAB support packages for out Raspberry Pi we decided to switch our operating system, to Raspbian since it is the default operating system for our Pi and MATLAB. The major challenge with this part was installing and working with an operating system for the Raspberry Pi that we'd never used before. To overcome these challenges, we did the following:

1. To install Raspbian, a new SD (secure disk)Card must be used in order to flash the image onto it. Our card was a SanDisk 16G SD Card.
2. The application used to flash the Raspbian is Etcher thus the first step was to download the application.
3. The zip file for Raspbian from the Raspberry Pi's official website had to be downloaded.
4. We used extraction application, 7-zip, to extract the image file out of the folder.
5. After inserting the SD card into the PC without formatting it, Etcher was then used to flash the Raspbian image into the hard drive. This process took 20 minutes.
6. When the process was completed, the SD card was inserted into the Raspberry Pi and after manually connecting the required components to power the device, our new operating system was able to boot.
7. With the new operating system, the camera software was installed again which was done through the same procedures used while doing the process in Ubuntu Mate.
8. A major challenge in using Raspbian as our operating system was preventing the SD card from being corrupted after shutting down each time. This was fixed after downgrading the resolution of the operating system, and changed from the default size to CEA Mode 2 720x480 60Hz 4.3.

Results:

The result of this is now Raspbian Operating System and SD Card drive are now stable and free from corruption. With it, we can now boot our Raspberry Pi and the Operating System smoothly without issues.

3.11 MATLAB Support Package Test

Test April 10 2018 (MR. 2, MR. 3, MR. 10 and ER. 5)

After the installation of Raspbian on the Raspberry Pi and connecting and using the camera by installing the camera software, the next step forward was to get the ArduCAM connected to the Pi in order to access the microcontroller and its external camera using MATLAB. With the connection, we would be able to utilize the ArduCAM to take a picture a retinal fundus photograph which will be sent to the Raspberry Pi where MATLAB will read, process, and detect whether the picture contains patterns of Diabetic Retinopathy or not.

Challenges:

The challenge in this procedure was to know how to connect the ArduCAM to the Raspberry Pi 3 properly. This was done before when our Raspberry Pi's Operating System was Ubuntu Mate, but now we are using Raspbian.

How this procedure of the project was accomplished was through a small amount steps involving both hardware and software applications.

1. Connect a USB-Ethernet adapter to Raspberry Pi 3.
2. Connect straight-through Ethernet Cable from adapter to Raspberry Pi 3.
3. Create a connection from MATLAB to the Pi using the command : mypi = raspi.
4. Test connection using MATLAB commands such as : writeLED and system.

Result:

The computer containing MATLAB programming application now has connection with the Raspberry Pi 3 and can access the microcontroller through various commands. It's able to access the camera and enable it to take pictures through the snapshot command, it's also able to see the pictures that the camera takes and use it for it's functional purpose.

3.12 Health Indicator Test

Completed Test: April 13, 2018 (ER. 3 and ER. 4)

Originally we wanted to include an LCD in our project for displaying the health state of the retinal fundus image after it goes through image processing in MATLAB. After discussing this with our advisors, we realized that the inclusion seemed difficult and pointless due to the time restraint. Thus we used an LED as an indicator instead; the LED is attached on a breadboard and is accessed by MATLAB through the Raspberry Pi's GPIO Pins. So the LED turns on whenever the algorithm determines the retinal image it sees it has the disease, the LED does not turn on when it is given a healthy result through our algorithm.

Challenges:

The main challenge in this aspect of the project was to learn how to use MATLAB to access the GPIO pins of the Raspberry Pi 3. Hooking up the LED on a bread was plain and simple, connecting it to the pins of the Raspberry Pi required some research on the Internet to know which pins exactly were needed to be selected, while knowing how to program LEDs to flash on and off required some programming and research.

The procedure in getting this portion to work was:

1. Finding a breadboard, some jumper wires, and an LED.
2. Research which pin numbers matched to the pin values that MATLAB reads.
3. Connecting the LED on breadboard to two GPIO pins of the Raspberry Pi 3, with one pin connecting a power-based pin and the other connected to ground-based pin.
4. Use MATLAB to program with GPIO pin would supply current, if statements and commands related to the support package were used to make this work.

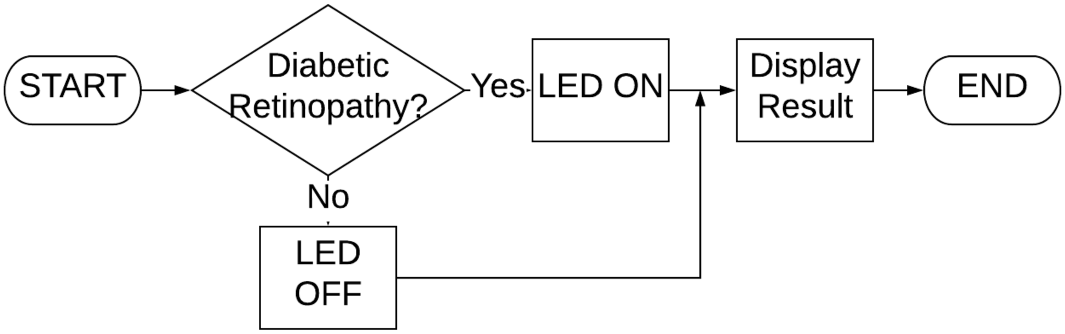


Figure 17: Health Indicator Flowchart

5. Run the simulation and the LED lights up depending of the algorithm result.

Results:

From the test, the LED is now able to light up whenever the algorithm from MATLAB detects retinal image to be Diabetic Retinopathy and that it will not turn on whenever the algorithm detects the retinal image to be Healthy. So when the output of the algorithm equals "DiabeticRetinopathy" a one/on value is sent to the GPIO pin connected to the positive terminal of the LED and supplies a voltage of 3.3V to it thus enabling it to light up. When the algorithm equals "Healthy", a zero/off value is sent to the GPIO pin connected to the LED, and no voltage is supplied.

4 Engineering Ethics and Our Project

Ethics is a an important requirement in engineering, since it requires development of specific product standards, and making sure the product is safe for the public. Engineers have to ensure the product meets the safety standards set. Our project is no different because we have a set list of marketing and engineering requirements that we must meet in order to determine the overall success of our product. The main way we have been able to achieve these goals thus far is by testing our device thoroughly. Including testing the algorithm of our product for its overall accuracy to make sure it's at a minimum of 90 percent, and our goal is to be able to repeat this result. We have been honest about our overall results so that we can ensure the safety and security of the every individual that is impacted by our product.

It is very important that throughout the project we are honest about what our product is capable of doing and also to make it known to the consumer that this product does not guarantee results, and that a medical professional should be contacted for further examination. Another important aspect to our project is that we have not tested it on a human retina, due to our own restrictions and that we are not medical professionals that can tell a patient whether or not they have diabetic retinopathy. This device is not a full diagnostic tool but rather a method to gain the information needed to make responsible health decisions. We will always advise to seek the opinion of a medical professional.

Coming to the use of the medical device, it will not be used to formally diagnose patients. We've followed the key guidelines set by our own marketing and engineering requirements, and make sure that the device follows all of the IEEE Code of Ethics. It will not be used on patients at this point of the project because right now we are unable to determine whether or not the device is safe to use directly in the eyes.

5 Future Work

Our project is now a complete diagnostic tool. However, there are things that can be added and worked on in order to complete it and make it safe to use on humans.

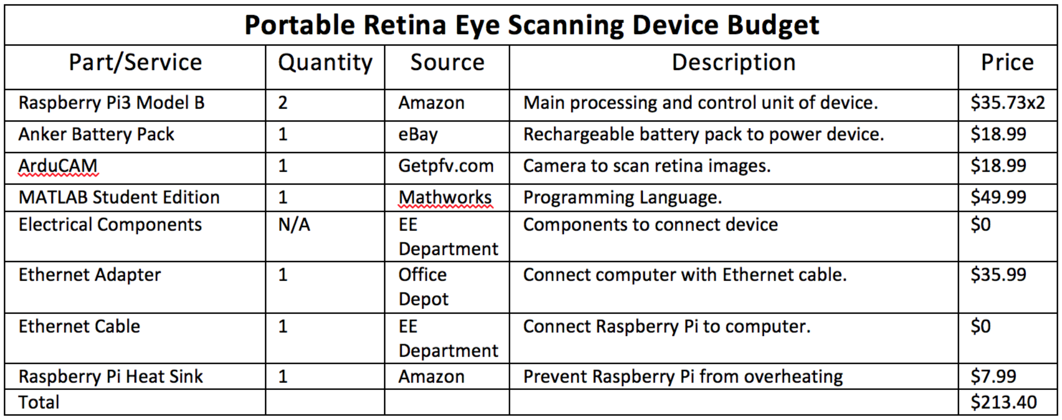
The next phase of this project will be to connect our device to a retinal fundus camera, this will enable the product to be used in order to take images of patient's retinas. After this step, the product will be ready for human use, but it will need to be tested with the new camera to make sure it is able to capture images. After the initial testing to make sure the camera works properly it would next be tested on humans to ensure that the product does not cause any eye issues such as blindness. After human testing depending on how reliable the results are the product will need to manufactured more heavily and sent to more rural regions that do not have access to well-trained professionals in order to get access to an easy to use portable machine that can make it easier test for diabetic retinopathy.

The next phase of this device will be for it to be utilized as a tool to help medical professionals not as well-trained in diagnosing diabetic retinopathy to have a device that is able to assist them in making a clearer and easier to understand diagnosis.

6 Acknowledgements

We would like to thank our Professor Farid Farahammand, Faculty Advisor Sudhir Shrestha, Industry Advisor Ben Valvodinos, and Client North Bay Vision Center for giving us feedback, and helping us with the completion of our project. We would also like to thank Shahram Marivani for providing us with parts and support throughout the project, and through our time at Sonoma State, he has been an asset to our growth during our time in the Engineering Department. We would also like to thank our family and friends, for their encouragement and support throughout our senior design project, those actions and kind words did not go unnoticed. Lastly, SSU SOURCE, for funding our project and give us more than enough money giving to make our idea a reality.

7 Materials and Costs

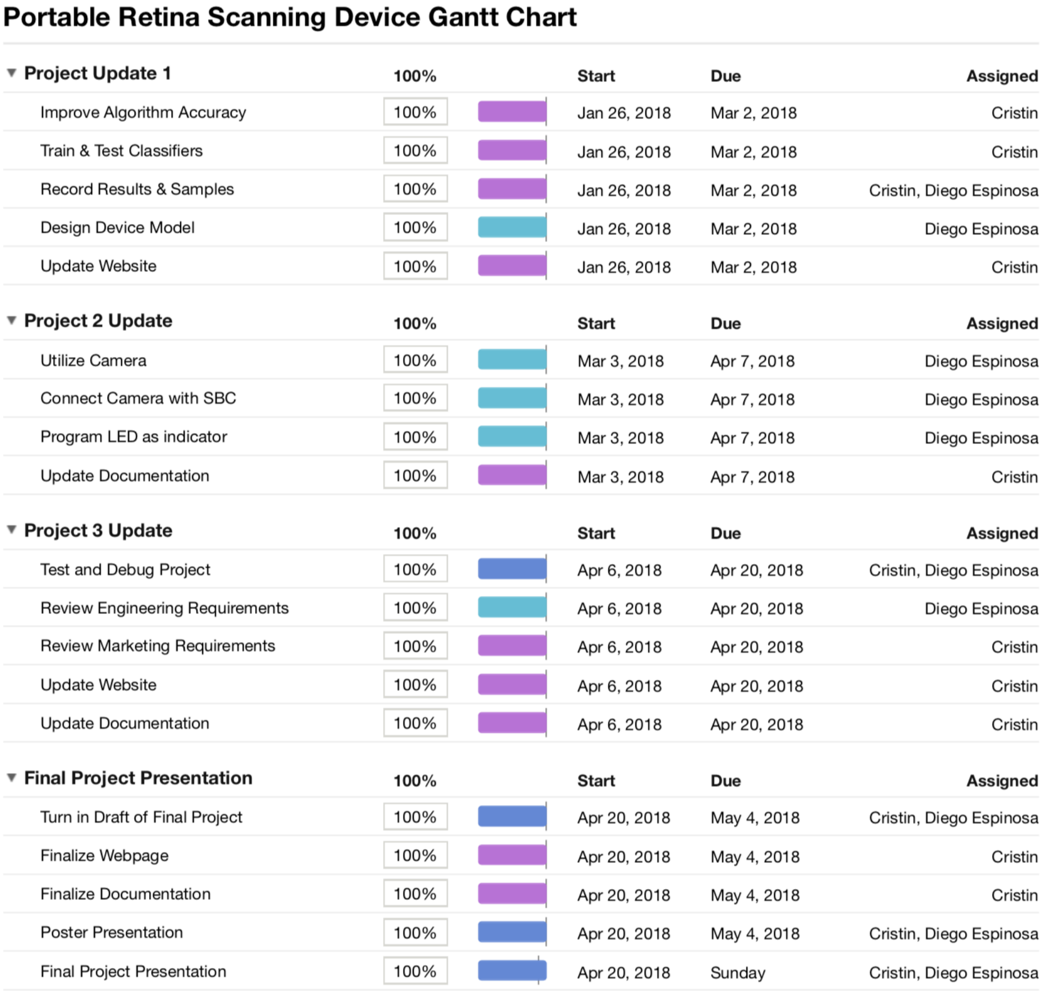


Portable Retina Eye Scanning Device Budget

Part/Service	Quantity	Source	Description	Price
Raspberry Pi3 Model B	2	Amazon	Main processing and control unit of device.	$35.73x2
Anker Battery Pack	1	eBay	Rechargeable battery pack to power device.	$18.99
ArduCAM	1	Getpfv.com	Camera to scan retina images.	$18.99
MATLAB Student Edition	1	Mathworks	Programming Language.	$49.99
Electrical Components	N/A	EE Department	Components to connect device	$0
Ethernet Adapter	1	Office Depot	Connect computer with Ethernet cable.	$35.99
Ethernet Cable	1	EE Department	Connect Raspberry Pi to computer.	$0
Raspberry Pi Heat Sink	1	Amazon	Prevent Raspberry Pi from overheating	$7.99
Total				$213.40

Table 4: This is a detailed version of our budget and labels our parts with their descriptions.

Project Timeline



Portable Retina Scanning Device Gantt Chart

		Start	Due	Assigned
▼ Project Update 1	**100%**	**Start**	**Due**	**Assigned**
Improve Algorithm Accuracy	100%	Jan 26, 2018	Mar 2, 2018	Cristin
Train & Test Classifiers	100%	Jan 26, 2018	Mar 2, 2018	Cristin
Record Results & Samples	100%	Jan 26, 2018	Mar 2, 2018	Cristin, Diego Espinosa
Design Device Model	100%	Jan 26, 2018	Mar 2, 2018	Diego Espinosa
Update Website	100%	Jan 26, 2018	Mar 2, 2018	Cristin
▼ Project 2 Update	**100%**	**Start**	**Due**	**Assigned**
Utilize Camera	100%	Mar 3, 2018	Apr 7, 2018	Diego Espinosa
Connect Camera with SBC	100%	Mar 3, 2018	Apr 7, 2018	Diego Espinosa
Program LED as indicator	100%	Mar 3, 2018	Apr 7, 2018	Diego Espinosa
Update Documentation	100%	Mar 3, 2018	Apr 7, 2018	Cristin
▼ Project 3 Update	**100%**	**Start**	**Due**	**Assigned**
Test and Debug Project	100%	Apr 6, 2018	Apr 20, 2018	Cristin, Diego Espinosa
Review Engineering Requirements	100%	Apr 6, 2018	Apr 20, 2018	Diego Espinosa
Review Marketing Requirements	100%	Apr 6, 2018	Apr 20, 2018	Cristin
Update Website	100%	Apr 6, 2018	Apr 20, 2018	Cristin
Update Documentation	100%	Apr 6, 2018	Apr 20, 2018	Cristin
▼ Final Project Presentation	**100%**	**Start**	**Due**	**Assigned**
Turn in Draft of Final Project	100%	Apr 20, 2018	May 4, 2018	Cristin, Diego Espinosa
Finalize Webpage	100%	Apr 20, 2018	May 4, 2018	Cristin
Finalize Documentation	100%	Apr 20, 2018	May 4, 2018	Cristin
Poster Presentation	100%	Apr 20, 2018	May 4, 2018	Cristin, Diego Espinosa
Final Project Presentation	100%	Apr 20, 2018	Sunday	Cristin, Diego Espinosa

Table 5: This is a detailed version of our project schedule that we have followed since January in order to ensure our project's completion. Pink color represents Cristin, Turquoise represents Diego, and Blue represents tasks both partners took part in.

Customer Survey

1. What technology do you use as an optometrist used to detect eye illness such as Diabetic Retinopathy or Glaucoma?

2. How quickly are you able to get results from the testing?

3. Do you have patients that have been diagnosed with the disease without being self-aware of their condition?

4. How important would it be to develop a portable device in scanning for Diabetic Retinopathy?

5. Would this portable device help raise self-awareness for individuals who have little to no access to health care, particularly those who live in rural regions from Third World nations? If so, why?

6. Who do you believe should be qualified to perform the test procedures from using such a portable device?

References

1. Diabetic Retinopathy
   In-text: (American Optometric Association, 2017)
   Your Bibliography: American Optometric Association. (2017). Diabetic Retinopathy. [online] Available at:
   https://www.aoa.org/patients-and-public/eye-and-vision-problems/glossary-of-eye-and-vision-conditions/diabetic-retinopathy [Accessed 7 Oct. 2017].

2. Portable Eye-Scanner to Revolutionize Detection of Diabetic Retinopathy
   In-text: (Diabetes.co.uk, 2017)
   Your Bibliography: Diabetes.co.uk. (2017). Portable Eye-Scanner to Revolutionize Detection of Diabetic Retinopathy. [online] Available at:
   http://www.diabetes.co.uk/news/2014/may/portable-eye-scanner-to-revolutionise-detection-of-diabetic-retinopathy-96133928.html [Accessed 11 Oct. 2017].

3. WHO — Diabetes cases could double in developing countries in next 30 years
   In-text: (World Health Organization, 2017)
   Your Bibliography: World Health Organization. (2017). WHO — Diabetes cases could double in developing countries in next 30 years. [online] Available at:
   http://www.who.int/mediacentre/news/releases/2003/pr86/en/ (Accessed 9 Oct. 2017).

4. What Is Machine Learning? — How It Works, Techniques and Applications - MATLAB and Simulink
   In-text: (Mathworks.com, 2017)
   Your Bibliography: Mathworks.com. (2017). What Is Machine Learning? — How It Works, Techniques and Applications - MATLAB and Simulink. [online] Available at:
   https://www.mathworks.com/discovery/machine-learning.html [Accessed 11 Oct. 2017].

5. Machine Learning — What is it and why does it matter?
   In-text: (SAS)
   Your Bilbography: SAS. Machine Learning — What is it and why does it matter?. [online] Available at:
   $https://www.sas.com/en_us/insights/analytics/machine-learning.html$

6. Introduction to Image Processing — What is image processing?
   In-text: (EngineersGarage)
   Your Bibliography: EngineersGarage. Introduction to Image Processing — What is image processing?. [online] Available at:
   https://www.engineersgarage.com/articles/image-processing-tutorial-applications

7. Epidemiology of Diabetic Retinopathy and Maculopathy in Africa — A systemic review
   In-Text: (NCBI, 2013 Apr 30)
   Your Bibliography: NCBI. (2013 Apr 30). Epidemiology of Diabetic Retinopathy and Maculopathy in Africa — A systemic review [online] Available at:
   https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4463765/

8. Diabetic Retinopathy Screening
   In-Text: (Wiley Online Library, July 2000)
   Your Bibliography: Wiley Online Library. (July 2000). Diabetic Retinopathy Screening [online] Available at:
   http://onlinelibrary.wiley.com/doi/10.1046/j.1464-5491.2000.00333.x/full