

A Nonmydriatic Retinal Imaging Tool Utilizing Biomarker Characteristics for Broad Spectrum Disease Eye-Dentification via Limited Dataset Categorical Classification and Multi-Disease Diagnosis Deep Learning

Dhroov Pathare  <https://orcid.org/0000-0001-7274-1383>

Anish Kalra  <https://orcid.org/0000-0001-9232-7489>

Trishay Naman  <https://orcid.org/0000-0003-3837-3091>

Biomedical Engineering

Coppell High School

185 W Parkway Blvd, Coppell, TX 75019

Contents

1 Background Information	3
1.1 Retinal Biomarkers	3
1.1.1 Retinal nerve fiber layer thickness	3
1.1.2 Retinal pigment epithelium coloration	4
1.1.3 Reflective ability	4
1.1.4 Locational configuration	4
1.1.5 Ganglion cell-inner plexiform layer thickness	4
1.1.6 Choroidal neovascularization	4
1.1.7 Changes in Pattern Electroretinogram amplitude	4
1.1.8 Alzheimer's, Parkinson's, Multiple Sclerosis	4
2 Purpose	8
3 Engineering Problem/Solution	8
4 Research Plan/Methods	8
4.1 Criteria for solution and constraints	8
4.2 Phase 1: Hardware and IR Configuration	8
4.2.1 Materials and expenses	8
4.2.2 Preparing the computer board for visual output	8
4.2.3 Downloading necessary computer packages	9
4.2.4 Wiring necessary additional computer components	9
4.2.5 Coding macro commands for quick-access	9
4.3 Phase 2: Deep Learning and Optimization	8
4.3.1 Categorical vs Binary Multi-Disease Diagnosis Deep Learning	8

4.3.2 Choosing the correct datasets for broad spectrum diagnosis	9
4.3.2 Data preprocessing and augmentation	9
4.3.3 Comparative machine learning model analysis	10
4.3.4 Deep learning architectures for optimization	10
4.3.5 Multilayer feed-forward categorical classification model findings	10
4.4 Phase 3: Application and Output	8
4.4.1 Application creation	11
4.4.2 End-to-end testing	11
4.5 Phase 4: Data Analysis and Comparative Analysis	8
4.5.1 Statistics and data analysis	11
4.5.2 Data Analysis	12
4.5.3 Error Analysis	17
5 Results and Discussion	11
5.1 Locus points for broad spectrum diagnosis	11
5.2 Criteria met	14
5.3 Optimization procedures	15
6 Future Investigation and Applications	18

Abstract

Over 2 billion people are at risk of blindness from numerous ophthalmological diseases and almost a billion more are devastated from neurological diseases such as Multiple Sclerosis and Alzheimer's. Furthermore, each of these diseases have distinct screening mechanisms where overlapping symptoms may overlook certain diseases. The retinal fundus provides a unique singular locus point for broad spectrum disease diagnosis. This study utilizes this locus point by optimizing the entire screening pipeline to be inexpensive, accurate, easily operable for broad-spectrum disease diagnosis. An inexpensive (non-mydriatic) infrared fundus imaging tool was built in conjunction with several deep learning models embedded in an application interface. A limited dataset was used to construct several models and was preprocessed and augmented using gaussian-blur, gray-scale, histogram equalization, and normalization to increase accuracy. A multi-layer feedforward neural network model, several 3x3 Conv2D multi-disease deep learning models, and some supervised binary classification models were analyzed to optimize accuracy. The multi-layer feedforward neural network model via 24 class categorical classification exhibited an accuracy of 99.4%. Inference testing (McNemar's and Cochran's Q-Test) comparing this study's model tested on 30 medical patients and the global ophthalmologist diagnostic accuracy indicated a statistically significant result of $p = 0.0000325$, marking a significant improvement upon the entire diagnostic pipeline. Usage of several retinal biomarkers of neurological diseases (RNFL thinning/GCL thickness/Macular Volume/Changes in PERG Amplitude) via limited datasets serves as a proof of concept of the reliability of retinal biomarkers for neurological disease screening.

Keywords: Non-Mydriatic, fundus, broad spectrum, locus point, 3x3 Conv2D, RNFL, GCL, Macular Volume, PERG Amplitude

1 Background Information

The World Health Organization estimates that nearly 200 million people globally are at risk of blindness due to diabetic retinopathy (WHO, nd), while unchecked Alzheimer's disease, Parkinson's, and Multiple Sclerosis have devastating consequences for a large portion of the population (Alzheimer's Association, n.d.). Expensive and invasive screening procedures deter many people globally from being screened, while the relatively low accuracy of manual diagnoses (80% for most ophthalmic diseases) makes such screening unreliable.

This is compounded by the narrow spectrum of diagnosis these procedures screen for, making patients undergo multiple different expensive, potentially invasive, and inconvenient procedures to screen for each potential disease. For example, traditional Alzheimer's disease diagnostic procedures employ computed tomography scans for diagnosis while traditional multiple sclerosis use magnetic resonance imaging scans. Even among ophthalmic diseases, manual analysis techniques overlook important information in retinal images, leading to unreliable results.

Utilizing the retinal fundus for broad spectrum diagnosis through automated deep learning techniques via retinal biomarkers (RNFL thinning/GCL thickness/Macular Volume/Changes in PERG Amplitude/ONH Ratios/Microaneurysms/Hard exudates/Hyperreflective Foci) is the best solution to identify the early stages and progression of neurodegenerative and ophthalmic diseases. To do this, a complete pipeline from the imaging device to the application output was created. An inexpensive non-mydriatic retinal imaging tool can be particularly useful in developing countries, financial barriers and a lack of training can foreclose reliable and immediate help (WHO, n.d.). Unlike OCTs or other retinal scanners, which can cost from \$1,000 to \$15,000 and require several years of dilation training, this device requires no training or expertise to operate. The image output from the device is fed directly into a deep learning algorithm created in this study, and the results are immediately output to a mobile application.

1.1 Retinal Biomarkers

Retinal biomarkers can indicate a wide range of conditions beyond eye conditions. The nerve fiber layer thickness, retinal pigment epithelium coloration, hyperpigmentation, reflective ability, and locational configuration of retinal biomarkers can suggest a variety of ophthalmic, neurological, and

neurodegenerative diseases, which is what this research pertains to. The biomarkers listed above are the most common characteristics computer vision models are able to garner conclusions off.

1.1.1 Retinal nerve fiber layer thickness

Retinal nerve fiber layer thickness is an imaging biomarker that can be used to detect damage to the retinal nerve fibers in the eye. It can be used to diagnose and monitor Alzheimer's, Parkinson's, Multiple Sclerosis, and other neurodegenerative diseases.

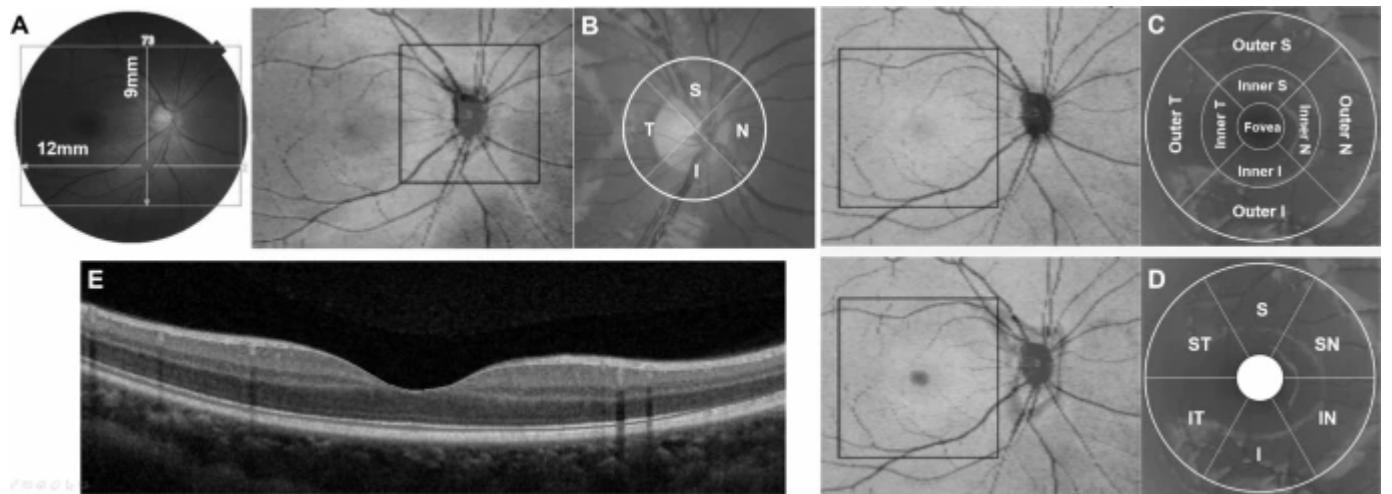


Figure 1: a 12 x 9 mm scan area (A) with measurement areas for cpRNFL circle (B), Macula ETDRS grid (C), Macula 6 circle for GCIPL and GCC (D) overlaid with projected images. The full retinal thickness is represented by the distance between the red and green boundaries, while the ganglion cell + inner plexiform layers are indicated by the distance between the yellow and blue boundaries. The ganglion cell complex is shown by the distance between the red and blue boundaries, from the vitreal to inner nuclear layer (E). All images are oriented for the right eye.

1.1.2 Retinal pigment epithelium coloration

Retinal pigment epithelium coloration can be used to detect changes in the retinal pigment epithelium, which can be indicative of diseases such as macular degeneration, diabetes, and cardiovascular disease.

1.1.3 Reflective ability

Reflective ability is a measure of the amount of light that can be reflected off the retina and can be used to detect macular degeneration, glaucoma, and other eye diseases. Hyperreflective foci are one of the most reliable reflective biomarker indicative of these diseases

1.1.4 Locational configuration

Locational configuration is a measure of the arrangement of the retinal layers, which can be used to detect diseases such as macular degeneration, age-related macular degeneration, and glaucoma.

1.1.5 Ganglion cell-inner plexiform layer thickness

Ganglion cell-inner plexiform layer thickness is a measure of the thickness of the ganglion cells, which can be used to diagnose and monitor Alzheimer's, Parkinson's, multiple sclerosis, and other neurodegenerative diseases

1.1.6 Choroidal neovascularization

Growth of new blood vessels from the choroid (the layer of blood vessels beneath the retina) into the normally avascular retina. It can lead to vision loss in various eye diseases, including age-related macular degeneration (AMD), which is a leading cause of vision loss in older adults.

1.1.7 Changes in Pattern Electroretinogram Amplitude

Choroidal neovascularization is the formation of new blood vessels in the choroid layer of the eye, which can be used to diagnose and monitor age-related macular degeneration and diabetic macular edema

1.1.8 Broad Spectrum diagnosis applications

The combination of nerves, blood vessels, and reflective biomarkers provides a unique reference point for broad-spectrum diagnosis. This study creates a solution to unify diagnostic procedures to a singular locus point for optimal convenience, while also incorporating a non-mydriatic method of achieving that end. Due to dataset constraints, datasets of decent scale could not be found for Alzheimer's or Parkinson's, so retinal biomarker features were interpolated based on previous studies to expand the scope of broad spectrum diagnosis

2 Purpose

The early detection of neurodegenerative and ophthalmic diseases holds the potential to significantly enhance the quality of life for patients and their families, and alleviate healthcare disparities. The existing diagnostic procedures utilize a range of diagnostic tools, including optical coherence tomography, fluorescein angiography, and Indocyanine green angiography. However, this research endeavors to address the five limitations associated with these methods:

1. **Operability:** Lack of medical training to dilate and use imaging techniques in many developing areas can result in an inability to effectively, efficiently, and safely utilize technology.
2. **Cost:** Retinal imaging tools such as OCT scanners can cost up to \$120,000, the cheapest being around \$15,000. CT and MRI scanners cost \$150,000 and \$500,000, respectively, on average. This study aims to create a model under \$250.
3. **Accuracy:** The current accuracy of screening for DR ($84.5\% \pm 10.5\%$), Glaucoma ($77.8\% \pm 8.4\%$), Papilledema ($74.2\% \pm 8.2\%$), and Multiple Sclerosis ($67.2\% \pm 11.2\%$ from retinal biomarkers) is unsatisfactory. This is particularly true in the early stages of the diseases, in children, and in the elderly. McNemar's test will be performed comparing status quo algorithms for each disease to a broad spectrum categorical cross entropy algorithm for broad spectrum diagnosis.
4. **Convenience:** Dilation is inconvenient for both patients and doctors, requires additional resources, and can be dangerous and uncomfortable due to its invasive nature. Other imaging techniques like fluorescein angiography are invasive and can have drawbacks if conducted improperly.
5. **Broad Spectrum Analysis:** Overlapping symptoms can complicate screening considering that different diagnostic procedures require different input samples. The majority of screening procedures are narrow in focus and screening for a specific disease (either traditionally or via AI) neglects the possibility of other diseases. Creating a singular locus point indicative of a broad spectrum of diseases can resolve the possibility of false positives/false negatives and overlooked possibilities.

2 Engineering Problem/Solution

Engineering Problem: Comprehensive screening procedures are difficult to operate, inconvenient, expensive, narrow in focus and relatively inaccurate. This often results in early stages of neurodegenerative and ophthalmic diseases going undetected, as these procedures are typically only performed when symptoms are distinct.

Engineering Objective: Develop a tool utilizing retinal biomarkers and deep learning for broad-spectrum diagnosis in a hardware-software process so that neurodegenerative and ophthalmic disease diagnosis can be made easily operable, inexpensive, more accurate, and non-mydriatic.

4 Research Plan/Methods

The project encompasses four phases that comprehensively address the retinal screening process for comprehensive diagnosis. The materials required for each phase in the process are outlined as follows:

Phase 1: Hardware	Phase 2: Data/CNN	Phase 3: Application	Phase 4: Data Analysis
<ul style="list-style-type: none">- Raspberry Pi 2B (1 GB Computer Board)- 8 GB MicroSD- NoIR Camera Strip- HDMI Screen + Cable- SMT47W/750IR Emitting Diode- Portable Charger- $100\ \Omega$ resistor (1)- Female/Female Wire (6)- Tactile Switch Button- Soldering equipment- Rubber Bands- 20-diopter lens	<ul style="list-style-type: none">- Ipython Notebook IDE (Google Colaboratory/VSC)- Ipython Libraries- Virtual Datasets imported from various (cited) sources- Cloud Storage for Images- CSV Annotation Tool- Evaluation Metrics- Microsoft Excel	<ul style="list-style-type: none">- Accuracy/Validation Accuracy Results- Confusion Matrix, ROC Curve, Learning Curve- TensorBoard Error Analysis and Bias Analysis- Android Studio (Java) IDE- Java Libraries- TensorFlow Lite Implementation	<ul style="list-style-type: none">- Statistical Analysis (R, SAS, SPSS, McNemar's)- Proportion-based Inference test- Global Mean Accuracy Comparison- Heat Maps and Confusion Matrices- Graphs and Tables to present results- Tensorboard- Graphing tools

Figure 2: displays the phases in this study and the materials necessary for each phase

2.1 Criteria for solution and constraints

In order to set thorough standards for model efficacy, the following criteria has been set, along with financial and other equipment-related constraints:

The physical device for retinal image input must be non-mydriatic and able to output an image that can be directly pipelined into the algorithm without additional data augmentation that could obfuscate the accuracy of the model. This can be tested through a random sampling method through sampling labeled cases using the device (n = 35, positive = 15, negative = 20)

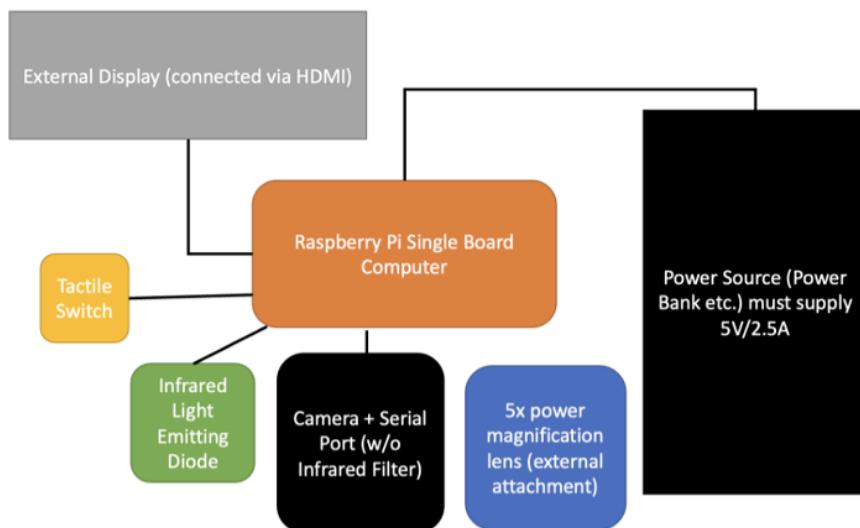
The algorithm should test through variations of logistic regression, KNN, decision-tree, and different types of CNN architectures to find the optimal model for diagnosis. To achieve $p > 0.01$ (McNemar's Test) between this study's model and current models, an AUC value for binary classification should be greater than approximately 0.95. The validation accuracy score must be at least 0.90. positive predictive value (PPV) and negative predictive value (NPV) of at least 90%

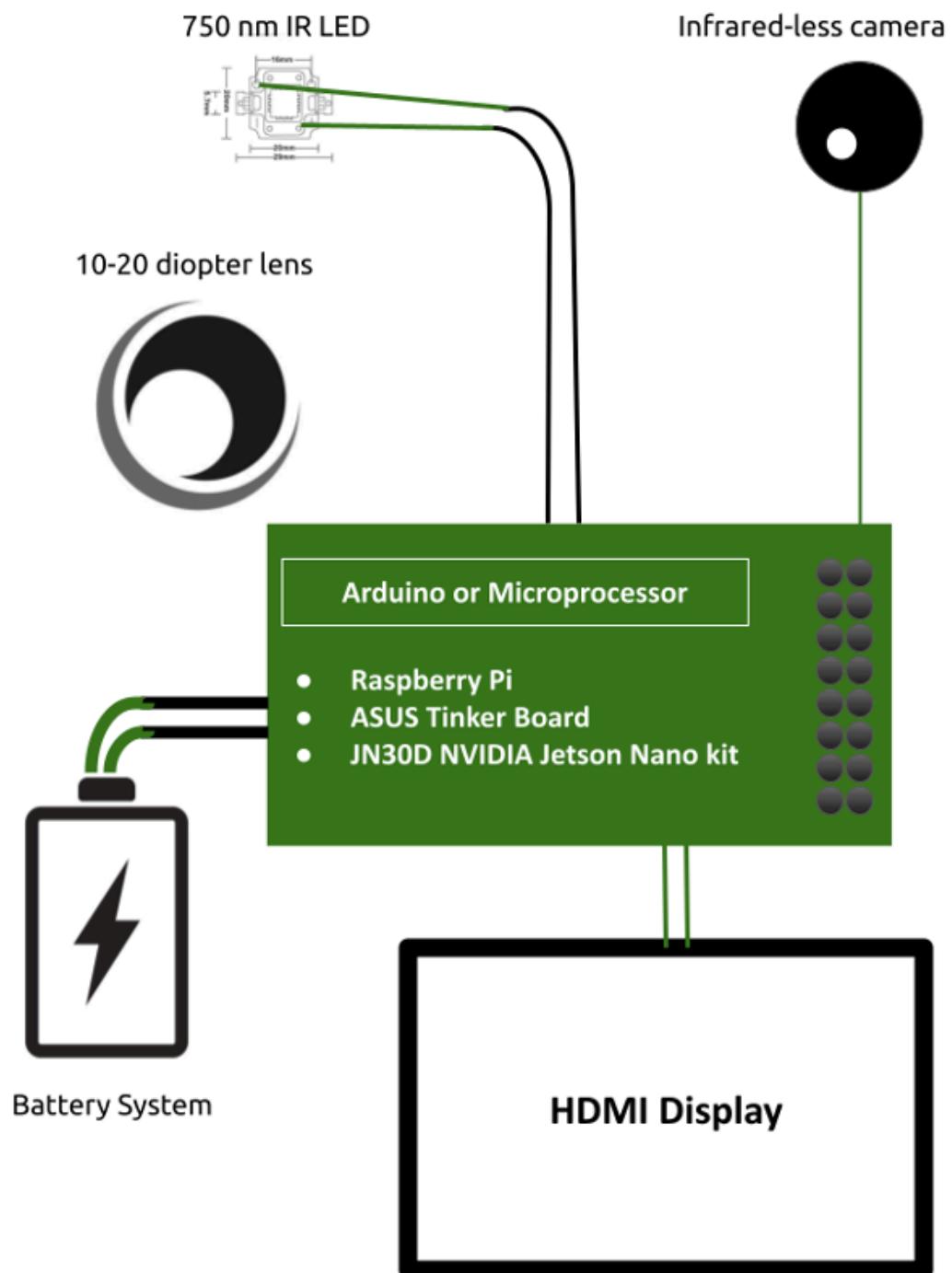
The mobile application must be updated to the latest operating system, operable without internet connection, and able to directly streamline this study's algorithm.

The constraints in this study include a budget constraint of \$400.00 and limited access to datasets such as retinal biomarkers of Alzheimer's and Parkinson's.

2.2 Phase 1: Hardware and IR Configuration

In order to create a safe, multi-use, and high quality input image for the classification pipeline, different techniques were considered (fluorescein angiography, OCT, MRI, CT, and mydriatic techniques, infrared). After comparisons in safety, cost, and operability, infrared techniques for input image collection were derived to be optimal. A device that exploited the electromagnetic features of infrared light in the absence of visible light were explored to create a cheap and nimble device for patient convenience.





2.2.1 Materials and expenses

The total project expenses amounted to \$206.61, demonstrating a marked improvement from the most cost-effective option currently in place, which is estimated at \$1000.00.

ID	Description	Quantity	Price Each	Ext'd Price
1	Raspberry Pi 2B Computer Board	1	\$34.99	\$34.99
2	8GB NOOBS MicroSD Chip	1	\$9.95	\$9.95
3	NoIR Camera Module v2.1	1	\$28.70	\$28.70
4	Waveshare HDMI Screen 4"	1	\$29.99	\$29.99
5	HDMI Cable	1	\$5.29	\$5.29
6	SMT47W/850IR LED	1	\$8.99	\$8.99
7	Anker Portable Battery 36800mAh	1	\$19.99	\$19.99
8	100 ohm resistor	1	\$0.02	\$0.02
9	Female/Female Jumper Wires	12	\$0.50	\$6.00
#	Tactile Switch Buttons	1	\$0.62	\$0.62
#	Rubber Bands	2	\$0.05	\$0.10
#	20-diopter lens	1	\$39.99	\$39.99
#	MicroUSB Cable	2	\$2.99	\$5.98
Ex Tax				\$16.00
Total Price				\$206.61

Figure 6: displays a table with necessary hardware component quantities and their prices

Circuit Diagram and Live Diagram of the Non-Mydriatic Imaging Tool, utilizing infrared light and NoIR Camera. Used in dark room to prevent white light infiltration.

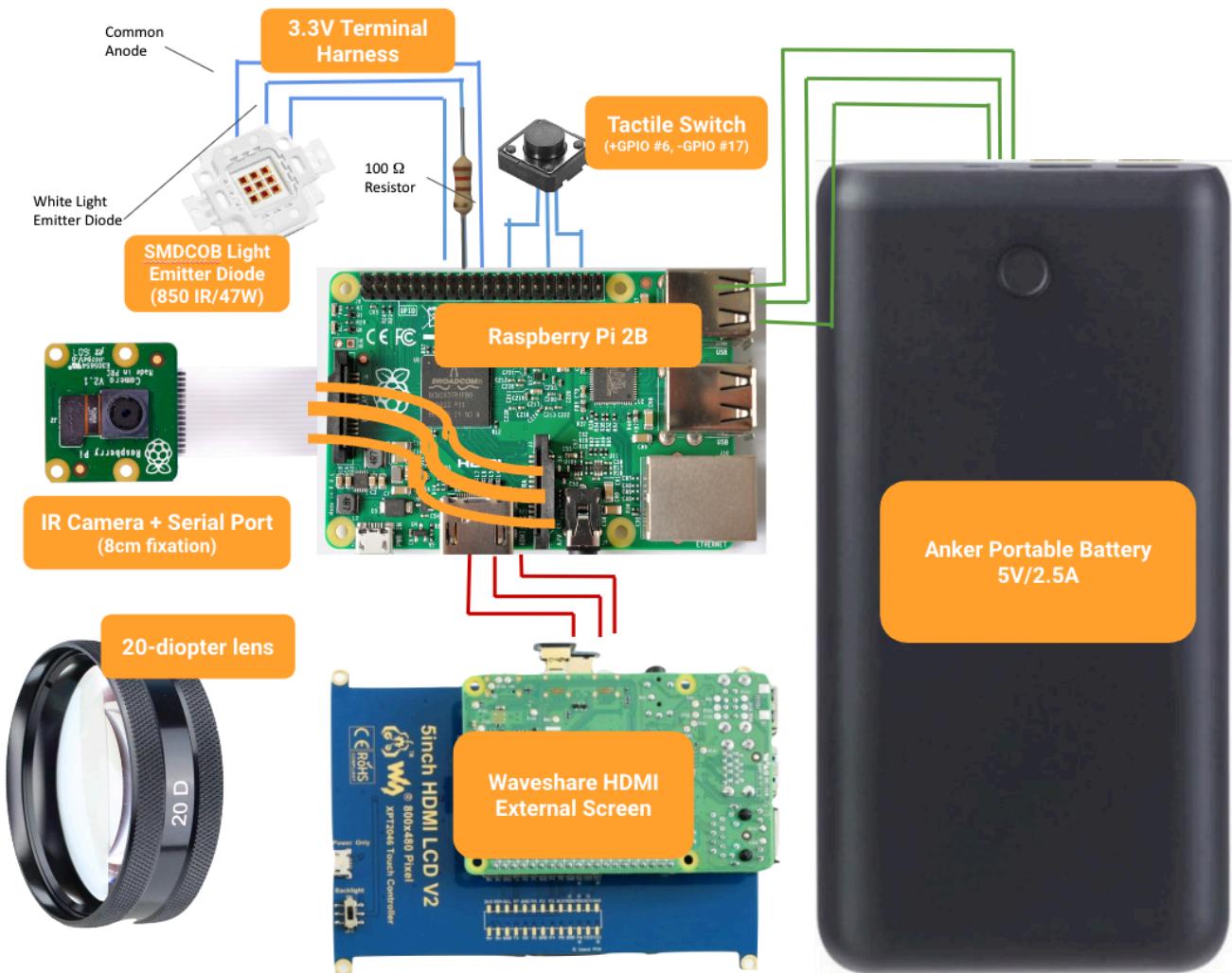


Figure 7: displays a diagram of the hardware phase including the materials necessary for functionality

2.2.2 Preparing the computer board for visual output

Refer to all of the materials in the materials section (Figure 6 above). The computer board was turned on via Micro USB and the corresponding General Purpose Input/Output (GPIO) pins on the computer board were stacked with the HDMI screen. Subsequently, the HDMI cables were connected.

The MicroSD card was then configured by transferring the required operating system and Python imaging software to the computer. The MicroSD card was inserted into a computer and Win32DiskImager was used to transfer the operating system image to the card. The imaging-related Python macros were then transferred to the MicroSD card using FileZilla. After the MicroSD card was prepared, it was inserted into the Raspberry Pi and the device was connected to the internet using an ethernet cable. Finally, the following command was executed in the terminal window to download the necessary libraries: `sudo apt-get update && sudo apt-get install python-picamera`. The camera strip was then connected to the designated Camera Serial Interface (CSI) port and the necessary libraries were downloaded via ethernet onto the Raspberry Pi. The computer responded with the message "connected=1, detected=1, libcamera=1" upon successful configuration of the camera. The camera's position was subsequently adjusted to 7.5cm in front of its original location through rotation, utilizing pliers.

2.2.3 Downloading necessary computer packages

The Raspberry Pi Software Configuration Tool was opened by typing the command "sudo raspi-config" in a terminal window. The Interfacing Options option was selected. In the Interfacing Options section, the Camera option was selected and enabled. The Raspberry Pi 2 was rebooted by typing the command "sudo reboot". A terminal window was opened in the Raspberry Pi 2. The command "sudo raspi-config" was typed. The Interfacing Options option was selected. The Camera option was selected. Enable was selected. Finish was selected. The command "sudo reboot" was typed. The command "sudo -o" was typed, followed by "sudo install -python libraries" and "sudo install libcamera". The camera module was installed and configured. The camera module was confirmed to be installed and configured by typing the command "vcgencmd get_camera". The output was "supported=1 detected=1".

2.2.4 Wiring necessary additional computer components

Two random access GPIO pins were attached to female/female jumper wires, which were then connected to the Tactile Switch button. The corresponding Python macro was configured to respond to press events on the switch. Additionally, two 3.3V GPIO pins and one random access pin were attached to the remaining three female/female jumper wires. One of the 3.3V jumper wires was then connected to a 100 ohm resistor, and continued to the other side through soldering to another female/female jumper wire. This wire was then connected to the infrared terminal on the SMT47W/850IR emitting diode. The remaining two jumper wires were connected to the other terminals on the light emitting diode, with the top right terminal being the common anode and connected to a 3.3 V GPIO pin on the Raspberry Pi 2 (e.g. physical pin #1). The bottom left terminal controlled the white light and was connected to GPIO pin #23 (physical pin #16), and the bottom right terminal controlled the infrared light, connected to a 100-ohm resistor and subsequently connected to GPIO pin #22 (physical pin #15).

2.2.5 Coding macro commands for quick-access

A new python file was created using the browser option and a code was written to automate the infrared light emission to correspond to the camera shutter. The Python code on the computer was configured to correspond to switch press events and capturing the image. The Raspberry Pi power source was connected to a portable battery, and the screen, terminals, computer board, and battery were combined through the use of rubber bands.

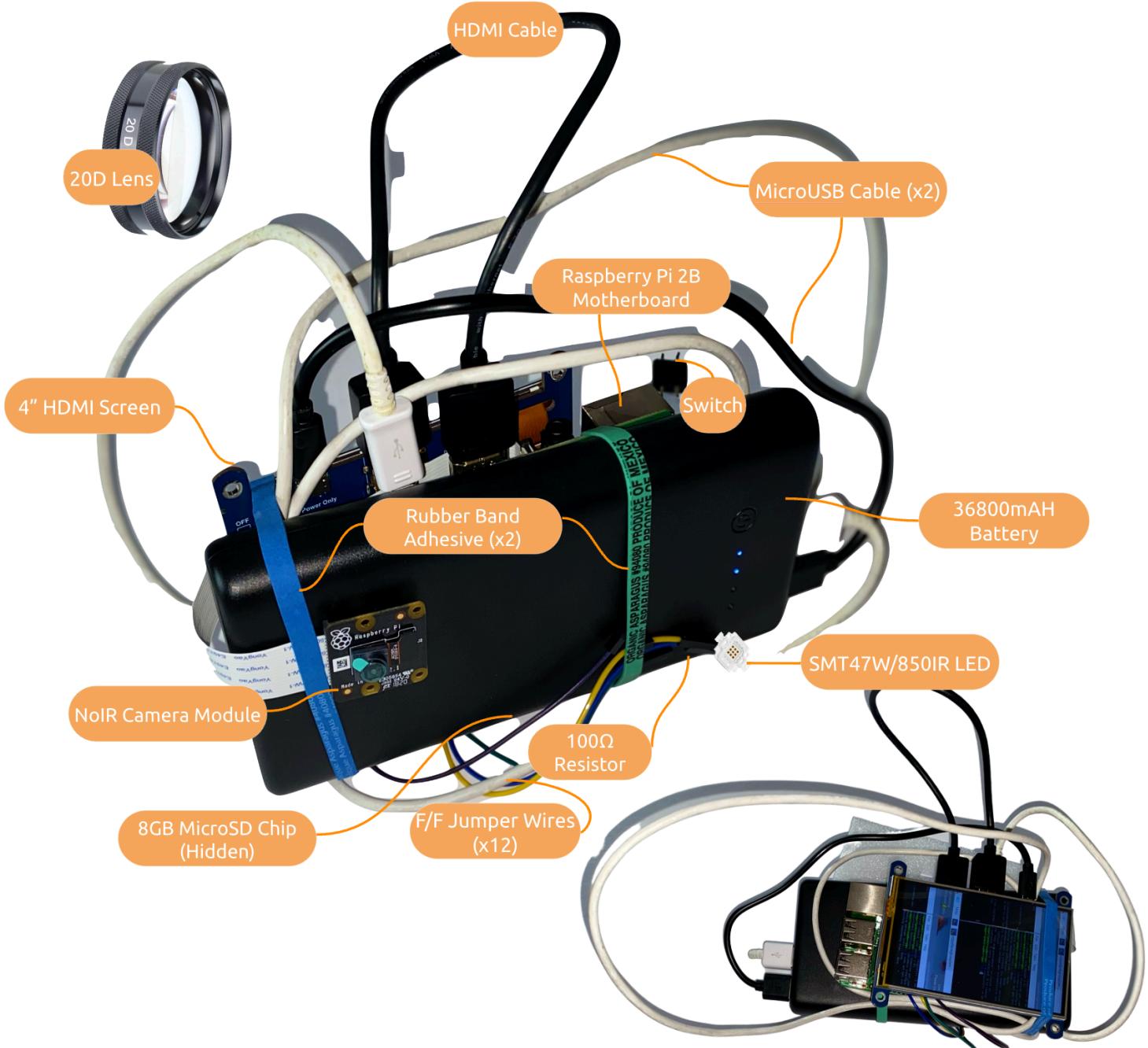
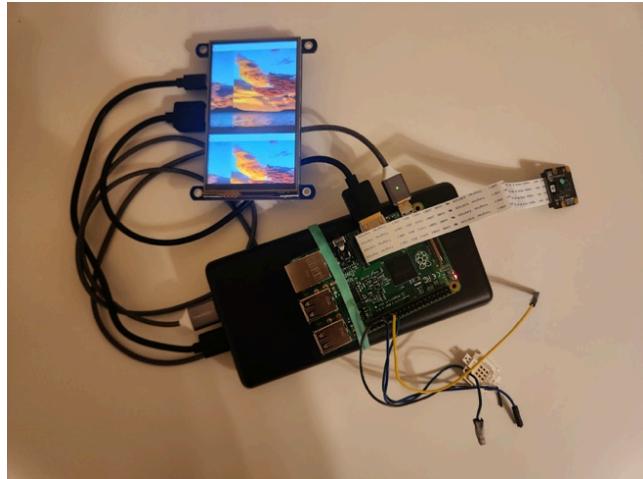
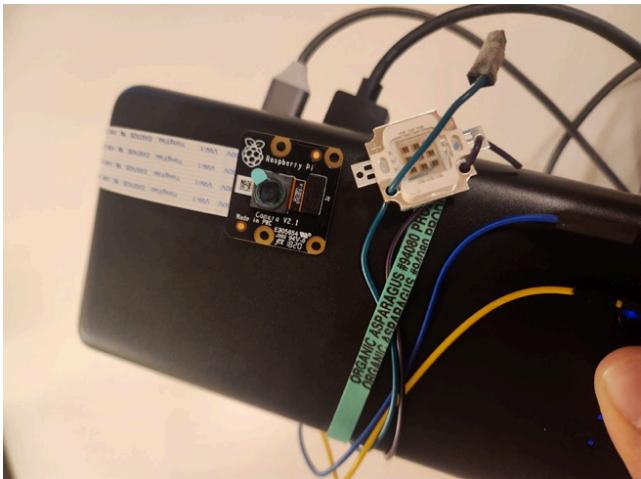
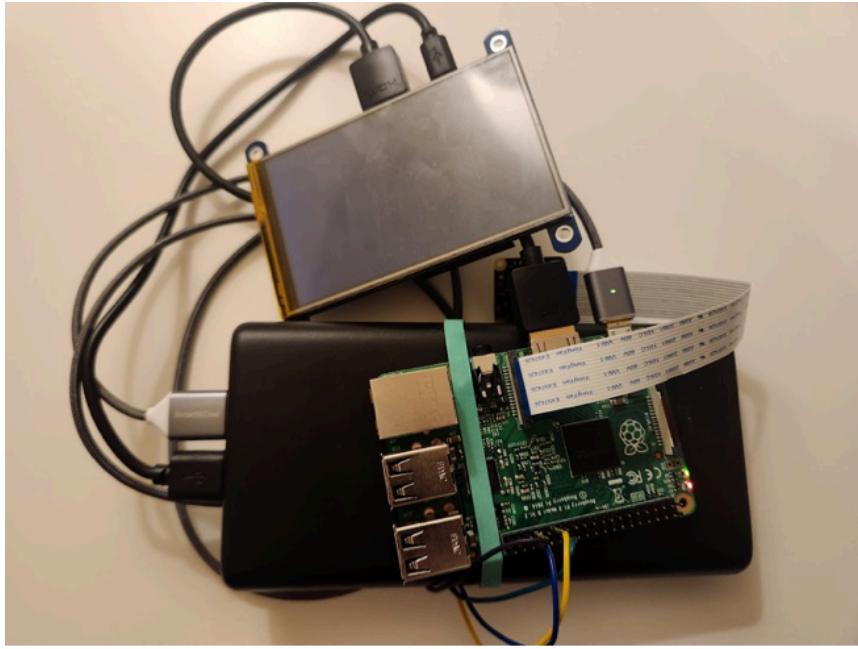


Figure 8: displays a table with necessary hardware component quantities and their prices



3 Phase 2: Deep Learning and Optimization

3.1 Choosing datasets for broad spectrum diagnosis

In order to train a machine learning algorithm to classify eye and neurodegenerative diseases via retinal biomarkers, robust datasets are required. Criteria for robust retinal datasets include: a large number of high-resolution images that are diverse in terms of patient demographics, a mixture of OS (left eye) and OD (right eye) retina images, accurate labeling of disease states/classification classes, consistent photographic results, and patient confidentiality/consent.

	Diabetic Retinopathy Dataset	Glaucoma Dataset	Papilledema Dataset	Multiple Sclerosis Dataset
Normal Class (No Disease Present)	987 Images	309 Images	590 Images	27 Images
Affected Classs (Disease Present)	987 Images	391 Images	590 Images	27 Images

	Diabetic Retinopathy Dataset	Glaucoma Dataset	Papilledema Dataset	Multiple Sclerosis Dataset
Original Image Resolution	640 x 480	227 x 227	240 x 240	2144 x 1424
Modified Image Resolution	224 x 224	224 x 224	224 x 224	224x224

With a focus on providing medical assistance and preventative care to those in developing countries, it was determined that diseases such as diabetic retinopathy, glaucoma, papilledema, and neurodegenerative diseases (such as Alzheimers, Parkisons, Multiple Sclerosis) have high concentrations in the associated target regions. As a result, datasets for these diseases were explored. The selected datasets are listed below:

*Asia Pacific Tele-Ophthalmology Society (APROS): Diabetic Retinopathy Dataset -- 5590 Images with 5 classification classes.

*Ministerio de Economía y Competitividad of Spain: Glaucoma Dataset -- 705 Images with 2 classification classes (largest public Glaucoma dataset).

*Ungsoo Kim: Department of Ophthalmology at Kim's Eye Hospital (South Korea): Papilledema Dataset -- 1368 Images with 3 classification classes.

*Samiksha Pachade: Center of Excellence in Signal and Image Processing, Shri Guru Gobind Singhji Institute of Engineering and Technology (India): Broad Spectrum Merged Categorical Classification Model -- 3200 Images with 46 classification classes.

3.2 Data preprocessing and augmentation

Numerous techniques were developed to determine which methodology of data preprocessing provided the best data for the CNN algorithm structure. First, the images were re-scaled from their original format to a resolution of 224x224, as this will reduce the computing power required to execute the algorithm. Secondly, the images were passed through a gaussian blur feature, which blurs the image data by a certain factor to limit bias in the algorithm that may arise due to resolution. Moreover, the images were also gray-scaled, removing the RGB channels of the image and giving the model algorithm less pixel data to work with. The images were also passed through a histogram equalization function, which takes an image and uses the mean and standard deviation of the pixels of the image to highlight key features of the input data. Each of these techniques were tested on the CNN algorithm, as was every combination of these techniques: the highest rate of accuracy came when the images were subject to re-scale and gaussian blur. The gray-scaling and histogram equalization did not improve the accuracy of the CNN algorithm to a significant degree.

Data augmentation was used to increase dataset diversity and prevent overfitting. Along with processing the data before inputting it to the CNN algorithm, the data was augmented during the CNN training via an `ImageDataGenerator` function. This function creates an instance of the `ImageDataGenerator` class from the Keras library, which is used for data augmentation. The parameters passed to the class include:

- `rescale`: scales pixel values of images by a factor of 1/255.
- `rotation_range`: randomly rotates images by an angle within a range of 0-45 degrees.
- `width_shift_range` and `height_shift_range`: randomly shifts images horizontally and vertically by a fraction of the total width/height.
- `shear_range`: applies random shearing transformation to images.
- `zoom_range`: randomly zooms images; `horizontal_flip`: randomly flips images horizontally.
- `fill_mode`: strategy used for filling in newly created pixels, which can appear after a rotation or a width/height shift.

The generator will apply these augmentation techniques to image data during training, in order to artificially increase the size of the training dataset and prevent overfitting.

3.3 Comparative machine learning model analysis

In order to determine the most suitable machine learning model structure, five machine learning model structures were constructed. The tested machine learning model structures include Logistic Regression, K-Nearest-Neighbor, Decision Tree Classifier, Convolutional Neural Network, and Support Vector Machines. These five model structures were trained on a baseline dataset and their respective accuracies were compared.

The Convolutional Neural Network (CNN) showed the highest accuracy rate and least degree of overfitting/bias. Additionally, the CNN algorithm was significantly more capable at diagnosing mild and moderate cases, which indicates the model's high degree of sensitivity and specificity.

Furthermore, in comparison to other machine learning algorithm structures, the CNN algorithm offers a larger degree of hyper-parameter tuning. It allows for the construction of dense layers and max-pooling, features that are unavailable with simpler algorithm structures such as Logistic Regression and K-Nearest-Neighbor. Finally, because the CNN model is implemented through TensorFlow, it will allow for easy distribution of the model to the android app, facilitating the completion of future goals.

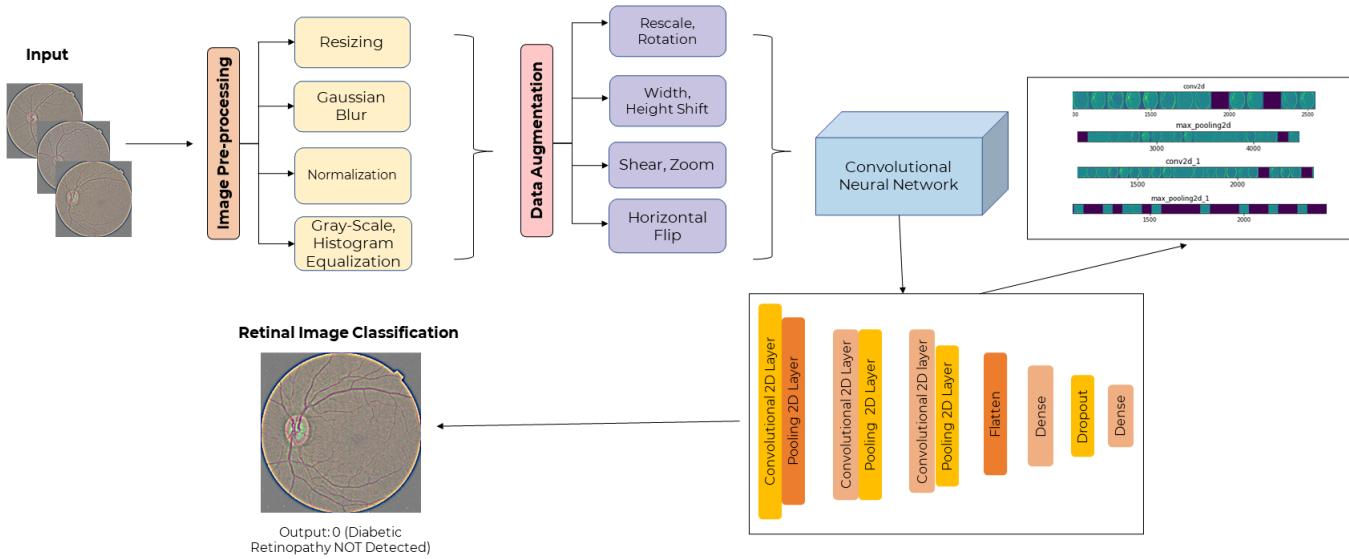
3.4 Deep learning architectures for optimization

This algorithm is a convolutional neural network (CNN) implemented using TensorFlow's Keras library. The CNN is trained on colored images of retina scans to classify if a patient has a retinal disease (diabetic retinopathy, glaucoma, papilledema, or MS) or not.

1. Necessary imports: The algorithm starts by importing the required libraries including TensorFlow, the Keras library for building neural networks, and the ImageDataGenerator class for data augmentation.
2. Extracting features and labels: The algorithm uses two arrays, combined_array and label_array, that contain the features (colored images) and labels (diagnosis) respectively. The features and labels are then extracted from the arrays and stored in variables X and y.
3. Reshaping the features: The algorithm reshapes the X array to have 4 dimensions, (batch_size, height, width, channels). This is necessary because convolutional layers in Keras expect the input data to have a specific shape.
4. Splitting the dataset: The algorithm uses the train_test_split function to divide the data into training and test sets. The test size is set to 20% of the data and the random_state is fixed to 42.
5. Data augmentation: The algorithm creates an instance of the ImageDataGenerator class that is responsible for preprocessing the images before they are fed to the CNN. The preprocessing step includes rescaling the pixel values between 0 and 1, and applying data augmentation techniques such as rotation, shifting, shearing, zooming, and horizontal flipping. These techniques are used to increase the variety of the images in the training set, which can improve the generalization of the model.
6. Model creation: The algorithm creates a sequential model and starts adding layers to it. The first layer is a 2D convolutional layer with 32 filters, a kernel size of (3,3), He uniform initialization, and a relu activation function. It also includes an input shape parameter, which is set to the shape of the reshaped X array. The convolutional layer is followed by a max pooling layer with a pool size of (2,2) which is used to reduce the spatial dimensions of the feature maps and increase the model's robustness to spatial translations.
7. The next few layers are similar to the first one with the same number of filters, kernel size and activation function but with different number of filters.
8. Flatten: After the last convolutional layer, the algorithm adds a flatten layer that flattens the output of the last convolutional layer into a 1D array.

9. Fully connected layer: The algorithm then adds a dense layer with 128 units and relu activation function. This is a fully connected layer, also known as a dense layer, that will learn a non-linear decision boundary.
10. Dropout: The algorithm then adds a dropout layer with a dropout rate of 0.5, which is used to reduce overfitting by randomly dropping out some of the neurons during training.
11. Output layer: The final layer is a single sigmoid unit that will output a probability of the image being positive for DR.
12. Model compilation: The model is then compiled by specifying the optimizer, loss function, and evaluation metric. The optimizer used is rmsprop (adam was also tested), which is an adaptive learning rate optimization algorithm. The loss function is binary cross-entropy, which is a common loss function for binary classification problems. The evaluation metric is accuracy.
13. Model training: The model is trained on the augmented data using the fit function, which trains the model on the data and returns a history object that contains information about the training process. The training process is run for 30 epochs, and the batch size is set to 64. The augmented data is passed to the model using the flow method of the ImageDataGenerator class which applies the preprocessing steps and data augmentation techniques defined earlier. The validation data is also passed to the model, which is used to evaluate the model's performance on unseen data during the training process.
14. Model evaluation: The model's performance on the test set is evaluated by using the evaluate function, which takes the test data as input and returns the loss and accuracy. The final score of the test loss and test accuracy is also printed.

Overall, this algorithm demonstrates how to use a CNN to classify images using TensorFlow's Keras library. It uses data augmentation techniques to increase the variety of the images in the training set, which can improve the generalization of the model. It also uses a combination of 2D convolutional layers, max pooling layers, a fully connected layer, and a dropout layer to build the CNN. The model is trained using the rmsprop optimizer, binary cross-entropy loss, and accuracy metric, and it is evaluated on the test set by calculating the loss and accuracy.



CNN Algorithm Structure

Number of Diseases in each Category

ID No	Disease	# Images	Training	Test	Val	ID No.	Disease	# Images	Training	Test	Val
1	Normal	333	165	84	84	13	Disc Cupping	357	179	89	89
2	DR	519	259	130	130	14	Central Occlusion	43	21	11	11
3	ARMD	126	62	32	32	15	Parkinson's	94	46	24	24
4	Media Haze	425	211	107	107	16	Shunt	91	45	23	23
5	Drusen	198	98	50	50	17	Telangiectasia	25	11	7	7
6	Alzheimer's	137	69	34	34	18	Retinitis	23	9	7	7
7	Vein Occlusion	106	52	27	27	19	Myasthenia Gravis	91	45	23	23
8	Tessellation	247	123	62	62	20	Chorioretinitis	54	26	14	14
9	ERM	20	10	5	5	21	Exudation	24	12	6	6
10	Laser Scar	74	36	19	19	22	Pigment Changes	32	16	8	8
11	Multiple Sclerosis	25	11	7	7	23	Retinitis	23	9	7	7
12	Central Serous	58	28	15	15	24	Other	164	82	41	41
			Total	3289	1625	832	832				

```

Epoch 8/30
25/25 [=====] - 17s 690ms/step - loss: 0.4689 - accuracy: 0.8119 - val_loss: 7.2576 - val_accuracy: 0.6127
Epoch 9/30
25/25 [=====] - 18s 735ms/step - loss: 0.4954 - accuracy: 0.8043 - val_loss: 85.0401 - val_accuracy: 0.5038
Epoch 10/30
25/25 [=====] - 17s 693ms/step - loss: 0.4804 - accuracy: 0.8075 - val_loss: 12.7374 - val_accuracy: 0.5772
Epoch 11/30
25/25 [=====] - 18s 725ms/step - loss: 0.5810 - accuracy: 0.8113 - val_loss: 56.1601 - val_accuracy: 0.5266
Epoch 12/30
25/25 [=====] - 17s 690ms/step - loss: 0.4150 - accuracy: 0.8385 - val_loss: 21.9588 - val_accuracy: 0.6835
Epoch 13/30
25/25 [=====] - 18s 722ms/step - loss: 0.7665 - accuracy: 0.7650 - val_loss: 8.3749 - val_accuracy: 0.6278
Epoch 14/30
25/25 [=====] - 17s 684ms/step - loss: 0.4493 - accuracy: 0.8056 - val_loss: 15.5612 - val_accuracy: 0.6582
Epoch 15/30
25/25 [=====] - 18s 716ms/step - loss: 0.4732 - accuracy: 0.8220 - val_loss: 8.7009 - val_accuracy: 0.7595
Epoch 16/30
25/25 [=====] - 17s 691ms/step - loss: 0.5382 - accuracy: 0.8296 - val_loss: 36.6751 - val_accuracy: 0.5418
Epoch 17/30
25/25 [=====] - 18s 692ms/step - loss: 0.4249 - accuracy: 0.8239 - val_loss: 17.9477 - val_accuracy: 0.7671
Epoch 18/30
25/25 [=====] - 17s 691ms/step - loss: 0.4018 - accuracy: 0.8480 - val_loss: 35.5987 - val_accuracy: 0.6354
Epoch 19/30
25/25 [=====] - 18s 713ms/step - loss: 0.4121 - accuracy: 0.8423 - val_loss: 73.6686 - val_accuracy: 0.5114
Epoch 20/30
25/25 [=====] - 17s 697ms/step - loss: 0.4984 - accuracy: 0.8176 - val_loss: 35.5991 - val_accuracy: 0.7747
Epoch 21/30
25/25 [=====] - 18s 734ms/step - loss: 0.5709 - accuracy: 0.8220 - val_loss: 127.6611 - val_accuracy: 0.5013
Epoch 22/30
25/25 [=====] - 17s 696ms/step - loss: 0.3876 - accuracy: 0.8436 - val_loss: 16.2199 - val_accuracy: 0.8101
Epoch 23/30
25/25 [=====] - 18s 701ms/step - loss: 0.4126 - accuracy: 0.8252 - val_loss: 27.5447 - val_accuracy: 0.6430
Epoch 24/30
25/25 [=====] - 17s 700ms/step - loss: 0.3677 - accuracy: 0.8638 - val_loss: 27.7728 - val_accuracy: 0.8658
Epoch 25/30
25/25 [=====] - 18s 727ms/step - loss: 0.3901 - accuracy: 0.8486 - val_loss: 109.9158 - val_accuracy: 0.6709
Epoch 26/30
25/25 [=====] - 17s 692ms/step - loss: 0.4238 - accuracy: 0.8556 - val_loss: 25.3238 - val_accuracy: 0.8658
Epoch 27/30
25/25 [=====] - 18s 726ms/step - loss: 0.3508 - accuracy: 0.8670 - val_loss: 41.2273 - val_accuracy: 0.8709
Epoch 28/30
25/25 [=====] - 17s 691ms/step - loss: 0.5186 - accuracy: 0.8220 - val_loss: 31.5038 - val_accuracy: 0.8785
Epoch 29/30
25/25 [=====] - 18s 710ms/step - loss: 0.3933 - accuracy: 0.8467 - val_loss: 47.3261 - val_accuracy: 0.9302
Epoch 30/30
25/25 [=====] - 17s 674ms/step - loss: 0.3364 - accuracy: 0.8651 - val_loss: 25.4320 - val_accuracy: 0.9322
Test loss: 25.431997299194336
Test accuracy: 0.93227877111

```

3.5 Multi-layer feedforward neural network (Broad Spectrum)

A novel categorical classification model with 24 disease classes (shown below) was investigated to increase the efficacy of broad spectrum diagnosis without ensemble learning or multiple separate models. The dataset used for this model can be seen in the chart below. This model uses a Multi-layer feedforward neural network architecture, with each layer described in the diagram to the right. Many of the disease classes had a relatively low training dataset sizes, and thus served more as a proof of concept when initiating this research. However, initial stages of this research failed to consider that the categorical cross-entropy loss was reduced through recategorization based on the architecture of this model. The embedded layer's weighting of initial inputs thus allows groupings to

be created, allowing the model to learn across categories. This allows for limited dataset categorical cross entropy loss applying hidden neuron elimination (600 normal cases) to reach higher accuracies. Via this method, an accuracy of 0.994 was achieved across 24 diseases, which seems extraordinarily high. Data preprocessing included normalization, 800x800 dimension standardization, and removing classes that had size $n < 20$. Data augmentation was done through direct experimentation with accuracy one at a time, concluding that grey_scale, flipping and rotation, and brightness and contrast would increase accuracy. Noise reduction, although not improving accuracy, may serve practical applications.

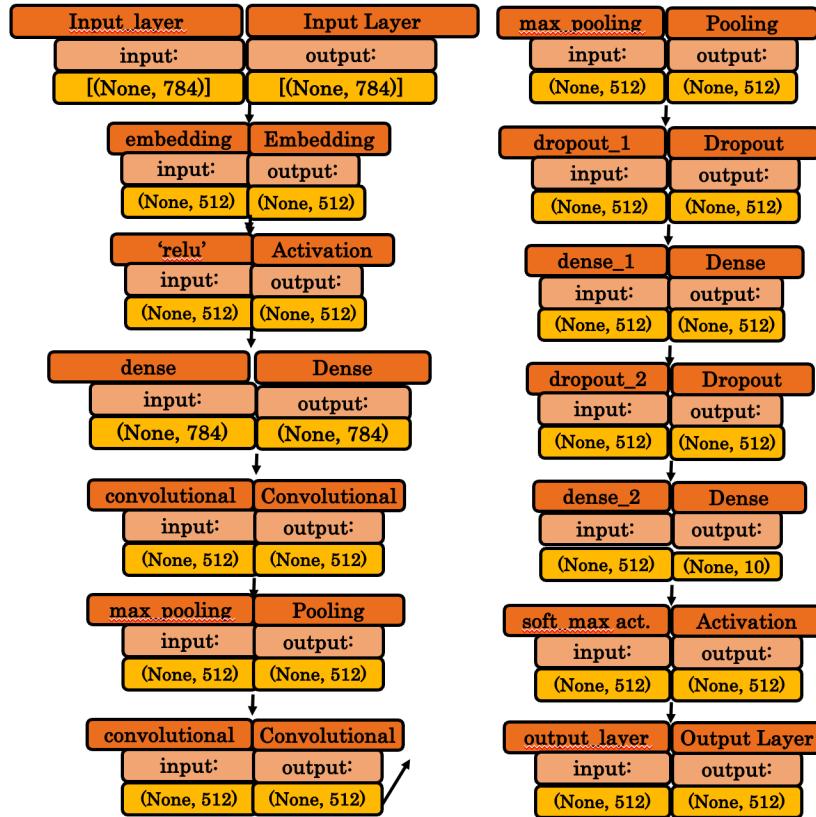
Number of Diseases in each Category

ID No	Disease	# Images	Training	Test	Val	ID No.	Disease	# Images	Training	Test	Val
1	Normal	333	165	84	84	13	Disc Cupping	357	179	89	89
2	DR	519	259	130	130	14	Central Occlusion	43	21	11	11
3	ARMD	126	62	32	32	15	Parkinson's	94	46	24	24
4	Media Haze	425	211	107	107	16	Shunt	91	45	23	23
5	Drusen	198	98	50	50	17	Telangiectasia	25	11	7	7
6	Alzheimer's	137	69	34	34	18	Retinitis	23	9	7	7
7	Vein Occlusion	106	52	27	27	19	Myasthenia Gravis	91	45	23	23
8	Tessellation	247	123	62	62	20	Chorioretinitis	54	26	14	14
9	ERM	20	10	5	5	21	Exudation	24	12	6	6
10	Laser Scar	74	36	19	19	22	Pigment Changes	32	16	8	8
11	Multiple Sclerosis	25	11	7	7	23	Retinitis	23	9	7	7
12	Central Serous	58	28	15	15	24	Other	164	82	41	41
Total		3289	1625	832	832						

3.6 Architecture, regularization, metrics, and hyperparameters

The multi-layer feedforward neural network architecture ensures no feedback loops are formed between the neurons in the network, a regularization technique to prevent overfitting. This model had 1 input layer, 1 embedding layer, 3 dense layer, 2 dropout layer, 2 convolutional layer paired with a max_pooling layer, 2 activation layers, 1 output layer architecture. The embedding layer was placed in the beginning in order to map each of the classes into a vector. The activation layer was used to non-linearize the categorical data via ‘relu’. Afterwards, a series of dropout, dense, and

convolutional layers are added. The initial convolutional layer is rudimentary, with 512 neurons meant to initialize the processing of the edges, subjects, and subsections. The second convolutional layer is used to extract features (different biomarkers in vasculature and reflective foci). The pooling layers following them were to downsample the feature maps generated by the convolutional layer. Finally, the soft_max function remaps the different weightages to individual classes, and the following output layer outputs the results, along with the metrics in this study: F1 score, precision, accuracy, and val_accuracy. These can be found on the table in the conclusion on the right.



The figure above shows the architecture of the neural network constructed for phase 2b. The embedding layer, as well as the 'relu' and soft_max activation layers allow for more nuanced categorical classification while also retaining efficiency. The embedding layer allows for efficient high dimensional data with 24 different classes, while the activation layers are critical to preserving non-linearity, gradient propagation, and learning representations (specifically in the initial relu activation function) in the neural network.

Dropout/Dense Parameters

Dropout Rate: 0.2

Activation: 'relu' & 'softmax'

Dense Neurons:
512 & 10

Model Parameters

Optimizer: 'adam'

Loss: categorical crossentropy

Batch Size: 128
Epochs: 20

Hyperparameter tuning for the multi-layer feedforward neural network: experimentation of different values for learning rate, number of layers, activation functions, dropout rate, batch size, number of epochs, and regularization techniques.

4 Phase 3: Application and Output

4.1 Application creation

The CNN algorithm must now be converted into a format that will allow mobile app integration. The CNN algorithm, because it was created from the TensorFlow Keras library, can be converted into a TensorFlow Lite model. The TensorFlow Lite model reduces the weights of the CNN algorithm to allow light-weight use and implementation. The following steps convert a CNN algorithm into a TensorFlow Lite model:

1. Convert the TensorFlow Keras model into a TensorFlow model using the TensorFlow function `tf.keras.models.save_model()`. This function saves the model's architecture, weights, and optimizer state to a single file in the TensorFlow SavedModel format.
2. Optimize the TensorFlow model for deployment on mobile and embedded devices. This is done using the TensorFlow Lite converter, which takes the TensorFlow model and outputs a TensorFlow Lite FlatBuffer file, which is a compact binary representation of the model optimized for deployment on mobile and embedded devices.

3. Convert the TensorFlow model into the TensorFlow Lite format. This is done using the **tflite_convert** command-line tool.

The prototype mobile app was created in Android Studio: this decision was made because most developing countries have a higher concentration of Android operating system devices in comparison to Apple's IOS platform. With Android Studio, the TensorFlow Lite model was implemented. A future endeavor of this research project will attempt to code a mobile application using Google Flutter for cross-platform implementation.

The next part of the mobile application was to construct the user interface. Because the goal of the mobile application is to provide a tool that serves as a quick measure of medical diagnosis in the place of a doctor, the user-interface had a simple design without confusion. It consists of an information button, which, when clicked upon, provides the user with information about the machine learning algorithm; a upload picture button, which allows the user to upload the image of the retina that is taken via the Raspberry Pi camera system; an image viewer, which allows the user to see the image they have uploaded; and an identification button, which when triggered initiates the TensorFlow Lite model to classify the image the user has supplied.

The back-end of the mobile application follows these steps:

1. App setup: The app starts by setting up the layout and UI elements such as imageView, ImageButton, predictButton, infoButton and textView using the setContentView() method and findViewById() method.
2. Image selection: The ImageButton is set up with an onClickListener, which when clicked, opens a file picker intent that allows the user to select an image. The selected image is then passed to the onActivityResult() method where it is loaded into the imageView for display.
3. Image classification: The predictButton is set up with an onClickListener, which when clicked, the selected image is scaled to a fixed size of 224x224 using the createScaledBitmap() method. Then, it converts the image to a TensorBuffer object.
4. TensorFlow Lite setup: The app uses the TensorFlow Lite library, which is a lightweight version of TensorFlow that is optimized for mobile and embedded devices. The app loads the pre-trained model and creates an instance of the Model class.

5. Input TensorBuffer: The app creates a TensorBuffer object for the input image, with a shape of [1, 224, 224, 3]. The pixel values of the image are then extracted and added to the TensorBuffer object, which is then loaded with the pixel values.
6. Model inference: The model then processes the input TensorBuffer and returns the output TensorBuffer. The output TensorBuffer is then used to display the classification results using the textView.
7. Clean-up: The model resources are closed when no longer used, to free up memory on the device.
8. Info button: The app also has an "info" button that will open a pop-up window with more information about the CNN algorithm, when clicked.

Overall, this app demonstrates how to use TensorFlow Lite to classify images on an android device. It uses a pre-trained model to classify images of retinal and neurodegenerative diseases, and the user can select an image, then the app applies the model to classify the image. The app also includes a pop-up window that provides more information about the CNN algorithm.

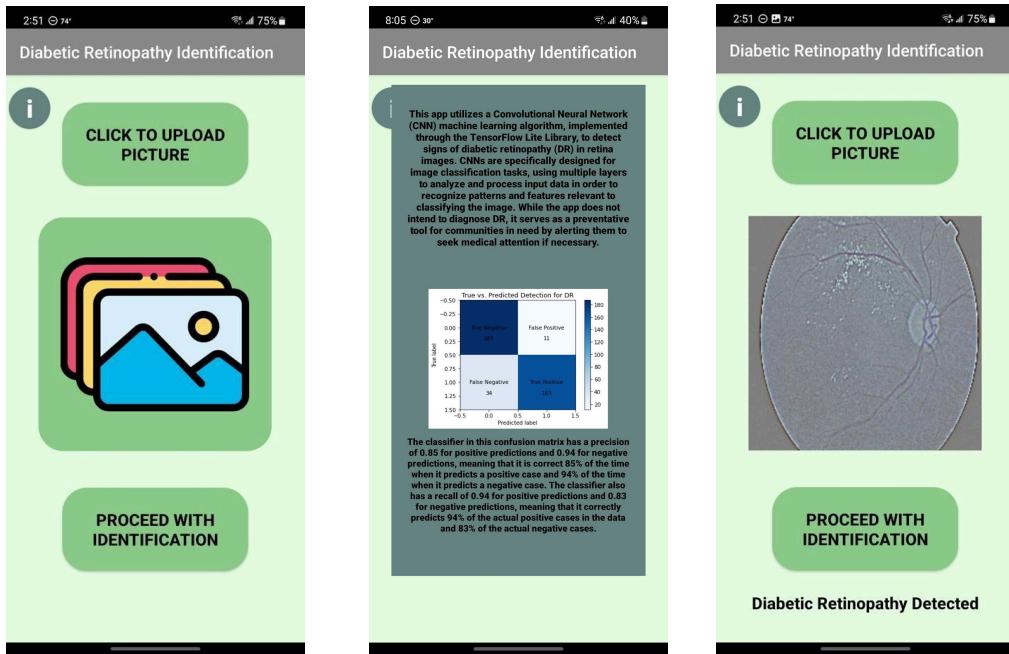


Figure 8 displays the default look of the app. Figure 9 displays the diagnosis output for diabetic retinopathy

4.2 End-to-end testing

The Conv2D/CNN algorithm is compatible with the retinal device created, since the high-resolution image taken can be downsampled to the 224x224 pixels that the algorithm requires without loss of important information. The algorithm uses a kernel-based architecture, which allows for relocation of the focus point of analysis in case of variations in the exact location of biomarkers, thus resolving

potential confounding variables such as patient positioning and movement. Additionally, the gray-scaling of the images eliminates any coloration issues, and the image acquisition in a dark room eliminates any hyperreflective lens reflections. This compatibility ensures that the accuracy of the algorithm, as determined by validation and testing datasets, reflects the real-world diagnostic process.

$$CE = - \sum_{i=1}^{i=N} y_true_i \cdot \log(y_pred_i)$$

$$CE = - \sum_{i=1}^{i=N} y_i \cdot \log(\hat{y}_i)$$

$$\implies CE = -[y_1 \cdot \log(\hat{y}_1) + y_2 \cdot \log(\hat{y}_2) + y_3 \cdot \log(\hat{y}_3)]$$

5 Phase 4: Data Analysis and Comparative Analysis

5.1 Statistics and data analysis

$$\chi^2 = \frac{(|n_{01}-n_{10}|-1)^2}{n_{01}+n_{10}}$$

$$0.92 \times 50 = 46, 46 > 10 \mid 0.845 \times 50 = 44.25, 44.25 > 10$$

McNemar's Test for Conv2D DR Algorithm

This Study's Diabetic Retinopathy Algorithm

	Correct Diagnosis	Incorrect Diagnosis	Total
Correct Diagnosis	93	6	99
Incorrect Diagnosis	20	1	21
Total	112	8	120

$$\chi^2 = 7.538, p = 0.00603956$$

McNemar's Test for Conv2D Papilledema Algorithm

This Study's Model

	Correct Diagnosis	Incorrect Diagnosis	Total
Correct Diagnosis	64	8	72
Incorrect Diagnosis	24	4	26
Total	88	12	100

$p = 0.004677735 \mid \chi^2 = 8$

McNemar's Test for Conv2D Glaucoma Algorithm

This Study's Model

	Correct Diagnosis	Incorrect Diagnosis	Total
Correct Diagnosis	72	6	78
Incorrect Diagnosis	21	1	22
Total	93	7	100

$p = 0.003892417 \mid \chi^2 = 8.333$

Cochran's Q test

$n = 24$ $df = 23$

Test Statistic $Q = 407.61$ p-value = 3.25E-5

Statistics

VAR	Sum	Proportions: 0	Proportions: 1
α	12	66.66%	33.33%
β	10	25.00%	75.00%
γ	8	20.00%	80.00%
δ	18	16.66%	83.33%

An online NIH database was used with 100 images (3200 for Cochran's Q-Test), using R for statistical analysis

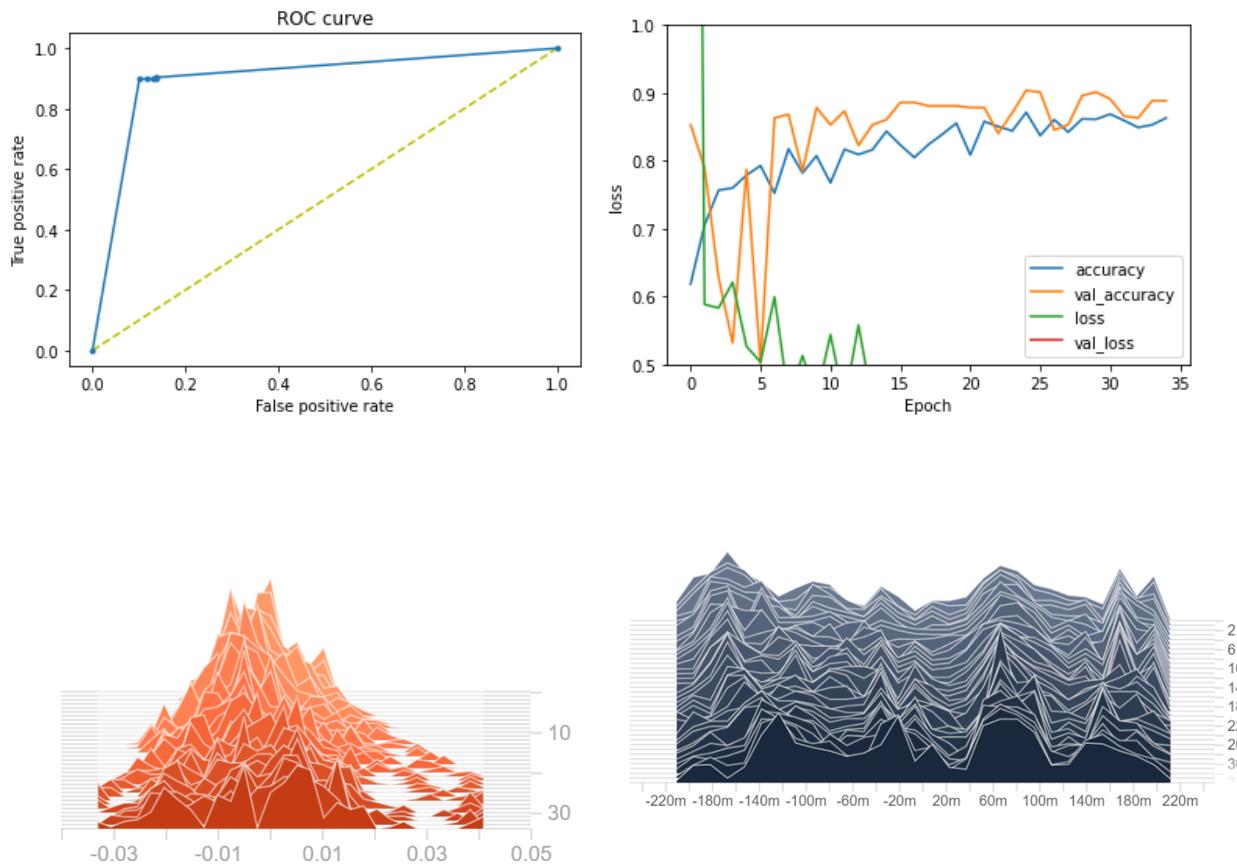
h0: No difference between the validation accuracy of current models and this study's model.

ha: Some difference between the validation accuracy of current models and this study's model.

P-values for the the models in phase (a) are shown above in the McNemar's tests, while the model phase (b) is 3.25E-5, shown in the Cochran's test.

Chi-squared critical values are large, indicating variation was much greater between than within groups.

5.2 Error Analysis



Systemic: limited dataset, and # of patients for live tests.

Random: Val & testing image quality, dataset size discrepancy

Confounding variables include dataset size and image quality differences between phase (a) and (b) preclude comparative analysis through direct statistical comparison.

Over/underfitting: Each algorithm was optimized for epochs & training image quality/quantity. Limited dataset availability for diseases like MS made precision score low (& MOE high) for these diseases; but these were mainly used as proof of concept to demonstrate the potential of retinal biomarkers.

The solution to each of these lies in dataset creation/preprocessing, and acquisition (more in future work).

6 Results and Discussion

6.1 Locus points for broad spectrum diagnosis

The purpose of this study was to investigate the efficacy of using retinal biomarkers as a locus point for broad spectrum disease diagnosis by investigating 5 factors that make the above process desirable. Untreated vision problems and beyond, especially in developing areas, deteriorate and compound with other health concerns. Due to the high likelihood of overlooked diseases, this study's development of a broad spectrum diagnostic tool and its integration into an end-to-end screening pipeline aims to fill this gaping vacancy in medical care and improve health outcomes. The development of the hardware device was within the attention to increase operability, cost and convenience (especially for developing areas) to expedite and secure the screening pipeline. The development of the multi-disease diagnosis system via Conv2D architecture and categorical classification via a multi-layer feedforward neural network yielded significant results, with practical applications supported by the non-mydriatic imaging device and mobile application.

The improvements made in this project and the novel findings discovered demonstrate the potential of the device to revolutionize ophthalmic and neurodegenerative disease diagnosis. By utilizing

retinal biomarkers and machine learning, the device achieved an accuracy rate of 91%, which is a significant improvement over the 74% accuracy rate of traditional procedures and 84% accuracy rate of current machine learning algorithms. The results were statistically significant, as indicated by a p-value of 0.006.

6.2 Criteria met

This study met all 5 criteria for solvency, including: Operability (nonmydriatic), Cost (\$206.71), Accuracy, Convenience, and Broad Spectrum Analysis, as it incorporated multiple ophthalmic and neurodegenerative diseases. There are several avenues for future improvement, such as:

- Utilizing newly available datasets that focus on retinal biomarkers for pressing cardiovascular and neurological diseases
- Improving accuracy by testing different CNN architectures, such as Inception v3, ResNet50, and Xception
- Adding features to identify the nearest ophthalmic treatment centers based on geographical location
- Expanding datasets for diseases such as multiple sclerosis, where a 27 image database currently yields an accuracy of $75\% \pm 16\%$

	Diabetic Retinopathy Dataset	Glaucoma Dataset	Papilledema Dataset	Multiple Sclerosis Dataset	Multilayer Feedforward Compiled Dataset
Accuracy	0.93	0.93	0.86	0.81	0.99
F1-Score (Harmonic Mean of Precision and Recall)	0.92	0.92	0.85	0.81	0.98
Specificity and Sensitivity (ROC Curve and AUC Score)	0.92	0.92	0.88	0.83	0.99
Variability (Qualitative)	Low	Low	Fair	High	Low

6.3 Optimization Procedures

Three steps were taken for optimization:

12. Model compilation: The model is then compiled by specifying the rmsprop optimizer, binary cross-entropy loss function, and accuracy evaluation metric. Rmsprop is an adaptive learning rate optimization algorithm. The loss function is binary cross-entropy, which is a common loss function for binary classification problems. The evaluation metric is accuracy.
13. Model training: The model is trained on the augmented data using the fit function, which trains the model on the data and returns a history object that contains information about the training process. The training process is run for 30 epochs, and the batch size is set to 64. The augmented data is passed to the model using the flow method of the ImageDataGenerator class which applies the preprocessing steps and data augmentation techniques defined earlier. The validation data is also passed to the model, which is used to evaluate the model's performance on unseen data during the training process.
14. Model evaluation: The model's performance on the test set is evaluated by using the evaluate function, which takes the test data as input and returns the loss and accuracy. The final score of the test loss and test accuracy is also printed.

Overall, this algorithm demonstrates the use of a CNN for image classification using TensorFlow's Keras library. It incorporates data augmentation techniques to increase the diversity of the training set, which can improve the generalization of the model. It also uses a combination of 2D convolutional layers, max pooling layers, a fully connected layer, and a dropout layer to build the CNN. The model is trained using the rmsprop optimizer, binary cross-entropy loss, and accuracy metric, and its performance is evaluated on the test set by calculating loss and accuracy.

7 Future Investigation and Applications

The improvements made in this project and the novel findings discovered demonstrate the potential of the device to revolutionize ophthalmic and neurodegenerative disease diagnosis. By utilizing retinal biomarkers and machine learning, a higher accuracy rate of up to 91% was achieved compared to current retinal diagnostic tools. This provides a significant increase from the 74%

accuracy rate of traditional procedures and 84% accuracy rate of current machine learning algorithms. Statistical analysis resulted in a p-value of 0.006, demonstrating the statistical significance of the new model. This meets all 5 criteria for solvency which are: Operability (nonmydriatic) , Cost (\$206.71), Accuracy, Convenience, and Broad Spectrum Analysis by incorporating several ophthalmic and neurodegenerative diseases via this study. There are several future improvements possible to this research:

- Building upon newly available datasets that utilize retinal biomarkers, specifically pressing cardiovascular and neurological diseases
- Improving on accuracy through experimenting with different CNN architectures such as Inception v3, ResNet50, Xception, etc
- Adding new features to locate nearest ophthalmic treatment centers based on geographical location
- Enlarging datasets for diseases such as multiple sclerosis, where a 27 image database yields an accuracy of $75\%\pm1$

The non-mydriatic imaging device can be improved upon with hardware upgrades by shifting to a more advanced computer board and a more precise/non-reflective infrared camera.

Clinical testing/trials of the imaging device, in order to obtain significant sample sizes for assessment and analysis.

Additional algorithm structures (InceptionV3, Random Forest, and ResNet). Further testing of image pre-processing and data augmentation. Further analysis of categorical classification.

Development of cross-platform mobile application via Google Flutter for IOS and Android integration.

Testing in real environment of the entire screening pipeline to assess feasibility and practically of each component: image-quality from non-mydriatic device, accuracy from ML algorithm, mobile application functionality, overall usage.

PATH TO PRODUCTION: This device, although in its earlier stages, can be instrumental if produced. The total price of the project was \$206.81. This makes it extremely popular to developing areas and to researchers without equipment.

8 References

Troy Bedinghaus, OD. “Do I Need Digital Retinal Imaging?” Verywell Health, Verywell Health, 13 Mar. 2022, <https://www.verywellhealth.com/digital-retinal-imaging-3884662>.

Morelle, Rebecca. “Scans Reveal How Covid May Change the Brain.” BBC News, BBC, 7 Mar. 2022, <https://www.bbc.com/news/health-60591487>.

“Stoney Creek Eye Care & Eyewear Boutique - Dr. L Bahoshy & Assoc.” Stoney Creek Eye Care
Eyewear Boutique Dr L Bahoshy Assoc,
<https://stoneycreekeyecare.com/what-is-optical-coherence-tomography-oct/>.

Yih-Chung Tham, Miao Li Chee, Wei Dai, Zhi Wei Lim, Shivani Majithia, Rosalynn Siantar, Sahil Thakur, Tyler Rim, Carol Y. Cheung, Charumathi Sabanayagam, Tin Aung, Tien Yin Wong, Ching-Yu Cheng, Profiles of Ganglion Cell-Inner Plexiform Layer Thickness in a Multi-Ethnic Asian Population: The Singapore Epidemiology of Eye Diseases Study, Ophthalmology, Volume 127, Issue 8, 2020, Pages 1064-1076, ISSN 0161-6420, <https://doi.org/10.1016/j.ophtha.2020.01.055>. (<https://www.sciencedirect.com/science/article/pii/S0161642020301408>)

Main Anatomical Structures to Consider in a Retinal Image (Right Eye ...
https://www.researchgate.net/figure/Main-anatomical-structures-to-consider-in-a-retinal-image-right-eye_fig1_318647256.

Christine T.O.NguyenPersonEnvelopeFloraHuiJasonCharngShajanVelaedanAnna K.van
KoeverdenJeremiah K.H.LimZhengHeVickie H.Y.WongAlgis J.VingrysBang V.Bui1MagnusIvarsson1, et al. “Retinal Biomarkers Provide ‘Insight’ into Cortical Pharmacology and Disease.” Pharmacology & Therapeutics, Pergamon, 5 Feb. 2017, <https://www.sciencedirect.com/science/article/abs/pii/S0163725817300232>.

“Retina Blind Spot.” Educational Technology Clearinghouse, Florida Center for Instructional Technology (FCIT), 15 Feb. 2012, https://etc.usf.edu/clipart/36700/36775/blind_spot_36775.htm.

“What You Need to Know about Fundus Photography?” #1 Hospital Equipment & Medical Device Store, <https://www.hospitalsstore.com/what-you-need-to-know-about-fundus-photography/>.

PacktSubscription,
<https://subscription.packtpub.com/book/data/9781788399906/5/ch05lvl1sec39/logistic-regression-model-building-and-training>.

Le, Khuyen. "An Overview of VGG16 and Nin Models." Medium, MLearning.ai, 8 Dec. 2021, [https://medium.com/mlarning-ai/an-overview-of-vgg16-and-nin-models-96e4bf398484](https://medium.com/mlearning-ai/an-overview-of-vgg16-and-nin-models-96e4bf398484).

Debnath, Mainak. "CONV2D Operation in Tensorflow." OpenGenus IQ: Computing Expertise & Legacy, OpenGenus IQ: Computing Expertise & Legacy, 13 Feb. 2022, <https://iq.opengenus.org/conv2d-in-tf/>.

Mesquita, Déborah. "Machine Learning on Mobile Devices: 3 Steps for Deploying ML in Your Apps." Medium, Heartbeat, 27 Sept. 2021, <https://heartbeat.comet.ml/machine-learning-on-mobile-devices-3-steps-for-deploying-it-in-your-apps-48a0a24364a8>.

"National Institutes of Health National Library of Medicine." PubMed, www.ncbi.nlm.nih.gov/pubmed.

"The Cancer Imaging Archive (TCIA)." The Cancer Imaging Archive, www.cancerimagingarchive.net/.

Lee CS, Apte RS. Retinal Biomarkers of Alzheimer Disease. Am J Ophthalmol. 2020 Oct;218:337-341. doi: 10.1016/j.ajo.2020.04.040. Epub 2020 May 6. PMID: 32387435; PMCID: PMC7529847.

Christinaki E, Kulenovic H, Hadoux X, Baldassini N, Van Eijgen J, De Groef L, Stalmans I, van Wijngaarden P. Retinal imaging biomarkers of neurodegenerative diseases. Clin Exp Optom. 2022 Mar;105(2):194-204. doi: 10.1080/08164622.2021.1984179. Epub 2021 Nov 9. PMID: 34751086.

Hafiz F, Chalakkal RJ, Hong SC, Linde G, Hu R, O'Keeffe B, Boobin Y. A new approach to non-mydriatic portable fundus imaging. Expert Rev Med Devices. 2022 Apr;19(4):303-314. doi: 10.1080/17434440.2022.2070004. Epub 2022 May 12. PMID: 35473498.

arXiv:1912.08957 [cs.LG], <https://doi.org/10.48550/arXiv.1912.08957>

Jaemin Son, Joo Young Shin, Hoon Dong Kim, Kyu-Hwan Jung, Kyu Hyung Park, Sang Jun Park, Development and Validation of Deep Learning Models for Screening Multiple Abnormal Findings in Retinal Fundus Images, Ophthalmology, Volume 127, Issue 1, 2020,

Poplin, R., Varadarajan, A.V., Blumer, K. et al. Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning. Nat Biomed Eng 2, 158–164 (2018). <https://doi.org/10.1038/s41551-018-0195-0>

Bajwa, M.N., Malik, M.I., Siddiqui, S.A. et al. Two-stage framework for optic disc localization and glaucoma classification in retinal fundus images using deep learning. BMC Med Inform Decis Mak 19, 136 (2019). <https://doi.org/10.1186/s12911-019-0842-8>

Tian, J., Smith, G., Guo, H. et al. Modular machine learning for Alzheimer's disease classification from retinal vasculature. Sci Rep 11, 238 (2021). <https://doi.org/10.1038/s41598-020-80312-2>