COMPARING THE EFFICIENCY OF NOVEL POINT-OF-CARE, LOW-COST NEURAL NETWORKS IN IDENTIFYING SPECIFIC STAGES OF DIABETIC RETINOPATHY ACROSS A LIMITED RETINAL DATASET

Aum Dhruv & Nicholas Harty | Fort Myers High School

PROBLEM

Diabetic retinopathy is a complication caused by an existing history of diabetes/diabetic symptoms that can slowly deteriorate a person's vision and can even lead to partial blindness. Often this diabetic complication can be treated by timely management of the condition and/or modern laser eye treatments/surgery. However, this treatment of diabetic retinopathy cannot completely cure the disease and, when left untreated, its effects seemingly take on worse results. These severe cases are often seen in developing nations where general access to medical testing is bleak with most patients unsure of their severity until physical implications become apparent. In fact, 79% of the adults with diabetes reside within low-to-middle income nations and serve a major risk of joining the estimated 93 million people worldwide suffering from diabetic retinopathy. Without significant medical testing, which can often prove expensive, and local ophthalmologists, this population is the most at risk of developing blindness and suffering the consequences of vision imparity. In this sense, the recent development of low-cost neural networks as a means of detecting early stages of diabetic retinopathy has become a much-needed solution to centuries of unknown suffering as a result of this complication. However, such solutions must be affordable/accessible as well as accurate in their results. This is the basis for this study and the reason for such urgency when it comes to verifying solutions.

RATIONALE

Over the past few years of neural processing, when it comes to computational innovation, two major post-processing algorithms, specifically within the realm of image classification, have proven suitable for low-cost, efficient medical testing: the convolutional neural network algorithm and the k-nearest neighbors algorithm. These low-process algorithms are highly dependent upon the amount of available training samples provided to their framework and the variability of the samples provided within testing. In previous research, both of these algorithms have been utilized to accommodate open image recognition that draws similarities to the detection of retinal hemorrhages, microaneurysms, hard exudates, and soft exudates that are necessary for classifying the five stages of diabetic retinopathy. Over the past year, the researchers have been studying the effectiveness and frameworks of a multitude of neural network algorithms in relation to biomarker detection. Through such development, along with investigative research into the biological foundation of diabetic retinopathy, the researchers narrowed down the two primary algorithms that are functionally and economically efficient in regards to biomarker analysis. In combining these two aspects of inquiry, the researchers developed an experimental study that analyzed the various accuracies of the convolutional neural network algorithm and the k-nearest neighbors algorithm in differentiating the stages of diabetic retinopathy amongst equal retinal samples while maintaining low-level constraints.

RESEARCH QUESTION

What is the accuracy of convolutional neural network algorithms compared to k-nearest neighbors algorithms, over different training sample sizes, in identifying specific stages of diabetic retinopathy severity in retinal images?

HYPOTHESIS

Hypothesis: If the investigators develop deep-learning algorithms of equal constraints and processes to identify specific stages of diabetic retinopathy, then the convolutional neural network algorithm will produce greater mean accuracy across higher training sample sizes in its identification process compared to a k-nearest neighbor algorithm with the same initial objective, which will produce greater mean accuracy across lower training sample sizes.

Null Hypothesis: If the investigators develop deep-learning algorithms of equal constraints and processes to identify specific stages of diabetic retinopathy, then there will be no statistical significance between the type of algorithm (convolutional neural network and k-nearest neighbor) and its mean accuracy across training sample sizes in its identification process.

VARIABLES

Independent:
- Training sample sizes (intervals at 125, 250, 375, 500, 1000, 1500, 2000, 2500)
- Type of NN algorithm

Dependent:
- Accuracy of each trial
- Correlation between training sample size and mean accuracy
- Difference between CNN and KNN with increasing sample sizes (line graph)

Control:
- Database of images
- Number of testing trials (10 trials with 100 unseen samples)

ASSUMPTIONS/BIASES & LIMITATIONS

While KNN and CNN algorithms are of similar structure, there are differences that may act as a limitation. The main difference between the convolutional neural network algorithm and the k-nearest neighbors algorithm exists in their ability to discern between distinct features of an image. The convolutional neural network algorithm processes feature regardless of spatial orientation because of its utilization of kernels that process training images while the k-nearest neighbors algorithm identifies features based upon color/intensity and takes spatial orientation into account when classifying testing images. Another difference exists in their individual training processes. The k-nearest neighbors algorithm trains upon its sample dataset by conforming the data into logit-based matrices which requires little computational power as it simply is building a record/ledger for later classification. On the other hand, the convolutional neural network algorithm uses higher computational power in its training process as it cycles through multiple Convolutional Layers as well as the refinement of its feature identification across a series of predefined epochs.
- Although these two algorithms are mathematically distinct in almost every component of their individual function, their output, in terms of correlation, are the same and they share similar efficiency when tasked with the training/classification of bitmapped medical datasets, hence the comparison is justified.

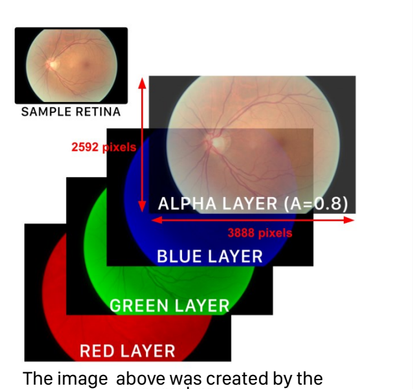
In conditions with a lower sample count, KNN and CNN algorithms may output a greater spread of results due to less of a dataset to work with.
- In order to combat this, the researcher incorporated multiple sample sizes.

The essential assumptions of the investigator's study include that the initial dataset, conceived through EyePACS's diabetic retinopathy study, was accurate in the classifications provided by physicians. Another assumption can be that the algorithms are accurately established to produce equal probabilities in fundamental testing (see "Methods" for more information). An assumption for the replication of the investigator's study can be seen in the equality of hardware across testing. This ensured equal distribution of RAM (1GB) and CPU clock speed (locked at 1.4 GHz). The final major assumption made in experimentation comes in the form of equality across sample bitmap dimensions. Each image within the investigator's dataset was minimized down to a 250px by 250px bitmap to ensure efficiency within testing. In a system that undergoes experimental procedures with higher input dimensions, the probabilities could differ based on the new fidelity of the developed feature-set.

BACKGROUND / LITERATURE RESEARCH

Introduction:

Millions of people across the globe suffer vision complications as a result of diabetic retinopathy, especially due to the lack of appropriate testing as to the severity of the condition. Diabetic retinopathy is a complication caused by an existing history of diabetes/diabetic symptoms that can slowly deteriorate a person's vision and can even lead to partial blindness. Often this diabetic complication can be treated by timely management of the condition and/or modern laser eye treatments/surgery. However, this treatment of diabetic retinopathy cannot completely cure the disease and, when left untreated, its effects seemingly take on worse results. These severe cases are often seen in developing nations where general access to medical testing is bleak with most patients unsure of their severity until physical implications become apparent. In fact, 79% of the adults with diabetes reside within low-to-middle income nations and serve a major risk of joining the estimated 93 million people worldwide suffering from diabetic retinopathy. Without significant medical testing, which can often prove expensive, and local ophthalmologists, this population is the most at risk of developing blindness and suffering the consequences of vision imparity. In this sense, the recent development of low-cost neural networks as a means of detecting early stages of diabetic retinopathy has become a much-needed solution to centuries of unknown suffering as a result of this complication. However, such solutions must be affordable/accessible as well as accurate in their results. This is the basis for the researcher's study and the reason for such urgency when it comes to verifying solutions. The focus of the study relies upon the two major post-processing algorithms, specifically within the realm of image classification, that have proven suitable for low-cost, efficient medical testing: the convolutional neural network algorithm and the k-nearest neighbors algorithm. Research into the function of these algorithms, as well as their applications, is important in reducing the need for physician training and increasing the mobility of worldwide medical testing.



The image above was created by the researchers

Convolutional Neural Network (CNN) Research:

The first of these recognition methods, the convolutional neural network algorithm, works by pooling various elements of a training image, assigning importance/weight to various aspects of an image, and then undergoing a convolution process that exponentially decreases the size of the image into basic constraints. Its ability to decrease the various prominent features of media allows it to function as an essential component in the classification of bitmaps, moving graphics, and audible samples. In this study, the description of this algorithm will focus on image classification as it relates to the entirety of the neural network's functionality. This entire process mimics the human brain's (or rather a human's Visual Cortex) ability to discern the components of an image and contrast the features of two corresponding images when dealing with necessary classification. This system of assigning importance to various features of an image is described as the CNN. Initially, this system breaks down training images into their essential channels: the Red Layer, the Green Layer, the Blue Layer, and, in some cases, the Alpha/Intensity Layer. This breakdown is a result of each pixel, in a standard training image, being composed of three RGB values between 0-255 in intensity and an alpha value between 0-1 in opacity. From this point of divergence, the algorithm begins dissecting the image for features through a series of four scientific steps defined by most computer analysts as essential "layers". These layers are composed of the Convolution Layer, the Pooling Layer, the Rectified Linear Unit (ReLU) Layer, and the Fully Connected Layer.

The first layer of CNN, the Convolutional Layer, processes the training image through a variety of kernels, also referenced as filters, that allow edges, corners, lines, shadows, and other elements of the image to become apparent. For each of these kernels, there is an output of a 2D map/matrix which outlines the intensity of each individual pixel within the filtered photo. Each channel (RGBA), as mentioned previously, is analyzed by the kernel. The kernel functions are a system that moves across an image and analyzes a specific area for features. For example, a 5x5 pixel image may have a 3x3 pixel kernel that moves across the image horizontally, starting from the left side, one pixel at a time analyzing each 9-pixel area for new features. Once finished with its horizontal motion, the kernel moves down one pixel and begins, once again, from the left side of the image. All throughout this process, these intensities/weights of these features within individual kernels are recorded in a stored model on the available RAM. Once the entire image has been outlined by kernels, the kernel either moves to another channel or, if all channels are covered, finishes its analysis by computing any additional biases provided by the investigators. Following this layer of computation, CNN shifts to the Pooling Layer.

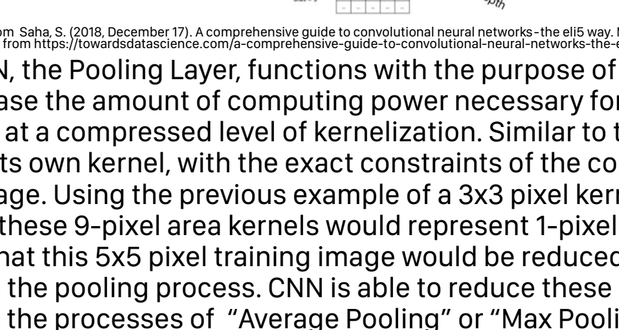


Image taken from Saha, S. (2018, December 17). A comprehensive guide to convolutional neural networks- the eli5 way. Medium. Retrieved December 29, 2021, from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

The second layer of CNN, the Pooling Layer, functions with the purpose of reducing the size of the image in order to decrease the amount of computing power necessary for convolution and to extract further features at a compressed level of kernelization. Similar to the Convolutional Layer, the Pooling Layer uses its own kernel, with the exact constraints of the convolutional kernel, to simplify the training image. Using the previous example of a 3x3 pixel kernel analyzing a 5x5 pixel training image, each of these 9-pixel area kernels would represent 1-pixel on the resulting image. This revelation means that this 5x5 pixel training image would be reduced to a 3x3 pixel resultant image after undergoing the pooling process. CNN is able to reduce these larger kernels into single pixels by either utilizing the processes of "Average Pooling" or "Max Pooling". Average Pooling takes the numerical average of all of the individual RGBA values within a kernel and uses that average to compose the RGBA of a new corresponding pixel. On the other hand, Max Pooling provides the dimensional compression provided by Average Pooling while also applying a layer of noise reduction to the resulting pixel. This process of Max Pooling helps the reduced training image maintain aspects of the original image and generally performs with greater accuracy when compared to Average Pooling CNN models. After the completion of the Pooling Layer, the CNN continues to cycle through the Convolution-Pooling layers until the necessary features are extracted and the original training image and been broken down into its essential pixels. The continuous nature of this cycle is determined by the number of cycles that are predefined by the investigators. In general, the accuracy of the model, when tested against its own training dataset, increases as it maintains further cycles. After this defined period of cycling between the Convolution and Pooling layers has ended, the CNN progresses into the ReLU and Fully-Connected Layers.

The Rectified Linear Unit (ReLU) Layer works in conjunction with the previously assessed data in order to "flatten"/familiarize the data before passing it on to the final layer. In the scope of image recognition, the ReLU Layer serves as a final simplification of pixels in a set linear matrix (1 pixel per training image that resulted from Convolution-Pooling Layers). From this layer, this linear pixel matrix is processed through the Fully-Connected Layer.

The final layer of CNN, the Fully-Connected Layer, processes this new linear matrix as well as the recorded weights from kernelization by cycling through a series of epochs, defined by the investigators, that helps increase the classification accuracy of the model. These epochs train the model on its existing training images which helps it distinguish dominant features from each individual classification group. This prolonged training of CNN's ability to discern between defining features is an asset used within image classification and the proficiency of trained data/images can often correlate directly to the assumptions of the algorithm when given new/key images. This later classification is refined using the "SoftMax Technique". This technique helps when identifying low-level features and refining the necessary recorded weights/biases from the Convolution Layer. Resulting from the classification of new images through "SoftMax" classification, the investigators should receive corresponding confidence intervals for each of their predefined classification groups.

```
CNN Results (Ex: Diabetic Retinopathy Severity):
    (0): 0.02      (1): 0.945     (2): 0.005
    (4): 0.015     (5): 0.015
```

For example, in the resulting dataset listed above, the investigators could assume, with 94.5% accuracy, that the testing retina image had mild symptoms of diabetic retinopathy based upon the training CNN.

K-Nearest Neighbors (KNN) Research:

The second of these algorithms, the k-nearest neighbors algorithm, is trained on various groups of images in order to classify a given unknown image. Its simple implementation allows it to function as an essential component in image recognition, video recognition, and speech recognition. KNN is used for classification and regression. In this study, the description of this algorithm will focus on image classification as it relates to the entirety of the neural network's functionality. This process uses a predetermined dataset, such as "MobileNet", to depose images into a series of 1000 logits (vectors of raw predictions that a model generates). This can also be done by developing a rudimentary algorithm to break down images via kernelling or random pixel sampling, into the necessary logits. To elaborate, for each group of images, a matrix is created of shape [n, 1000], in which "n" is the number of samples per group. MobileNet deposes new images into logits, which are then normalized to unit length. This means that the logit vectors are scaled to unit vectors using the formula below.

$$u = \frac{v}{\|v\|}$$

Image taken from Thorat, N. (2018, June 20). How to build a teachable machine with Tensorflow.js. Observable. Retrieved December 29, 2021, from https://observablehq.com/@nsthorat/how-to-build-a-teachable-machine-with-tensorflow-js)

Following this, the logits are concatenated to the end of the matrix, forming an additional row.

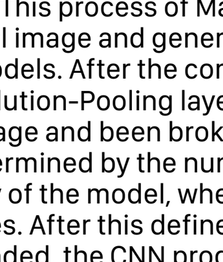


Image taken from Thorat, N. (2018, June 20). How to build a teachable machine with Tensorflow.js. Observable. Retrieved December 29, 2021, from https://observablehq.com/@nsthorat/how-to-build-a-teachable-machine-with-tensorflow-js)

The matrix ends up looking similar to the image above.
When given an image with no existing classification, MobileNet deposes it into logits, then normalizes to a vector of shape [1000]. After this, the unknown image's vector is multiplied to the matrix.

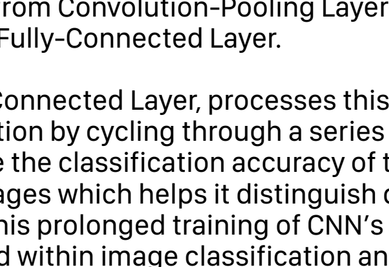


Image taken from Thorat, N. (2018, June 20). How to build a teachable machine with Tensorflow.js. Observable. Retrieved December 29, 2021, from https://observablehq.com/@nsthorat/how-to-build-a-teachable-machine-with-tensorflow-js)

The equation ends up looking similar to the image above. The resulting vectors are sorted by distance from the unknown image in increasing order. The closest vectors are called the "nearest neighbors", and the K value determines the extent to which "neighbors" are taken into account when classifying the unknown image. Hence, the algorithm's name is the k-nearest neighbors algorithm.

Concluding Comparison Between Neural Networks:

The main difference between the convolutional neural network algorithm and the k-nearest neighbors algorithm exists in their ability to discern between distinct features of an image. The convolutional neural network algorithm processes feature regardless of spatial orientation because of its utilization of kernels that process training images while the k-nearest neighbors algorithm identifies features based upon color/intensity and takes spatial orientation into account when classifying testing images. Another difference exists in their individual training processes. The k-nearest neighbors algorithm trains upon its sample dataset by conforming the data into logit-based matrices which requires little computational power as it simply is building a record/ledger for later classification. On the other hand, the convolutional neural network algorithm uses higher computational power in its training process as it cycles through multiple Convolutional Layers as well as the refinement of its feature identification across a series of predefined epochs. Although these two algorithms are mathematically distinct in almost every component of their individual function, their output, in terms of correlation, are the same and they share similar efficiency when tasked with the training/classification of bitmapped medical datasets.

METHODS

Data Training Collection:
The researchers collected data from EyePACS, an organization which provided access to multiple retinal images classified by different stages of severity. The stages consisted of "no present diabetic retinopathy", "mild diabetic retinopathy", "moderate diabetic retinopathy", "severe diabetic retinopathy", and "proliferative diabetic retinopathy." 250 randomly selected images were taken out of the sample to later be used for testing. To reduce bias, researchers also incorporated random sampling in training the k-nearest neighbors model and convolutional neural network model. Both neural network models were trained to a random sample of 25 images for each stage (totaling 125 retinal images), 50 images for each stage (totaling 250 images), 75 images for each stage (totaling 375 images), 100 images for each stage (totaling 500 images), 200 images for each stage (totaling 1000 images), 300 images for each stage (totaling 1500 images), 400 images for each stage (totaling 2000 images), and 500 images for each stage (totaling 2500 images).

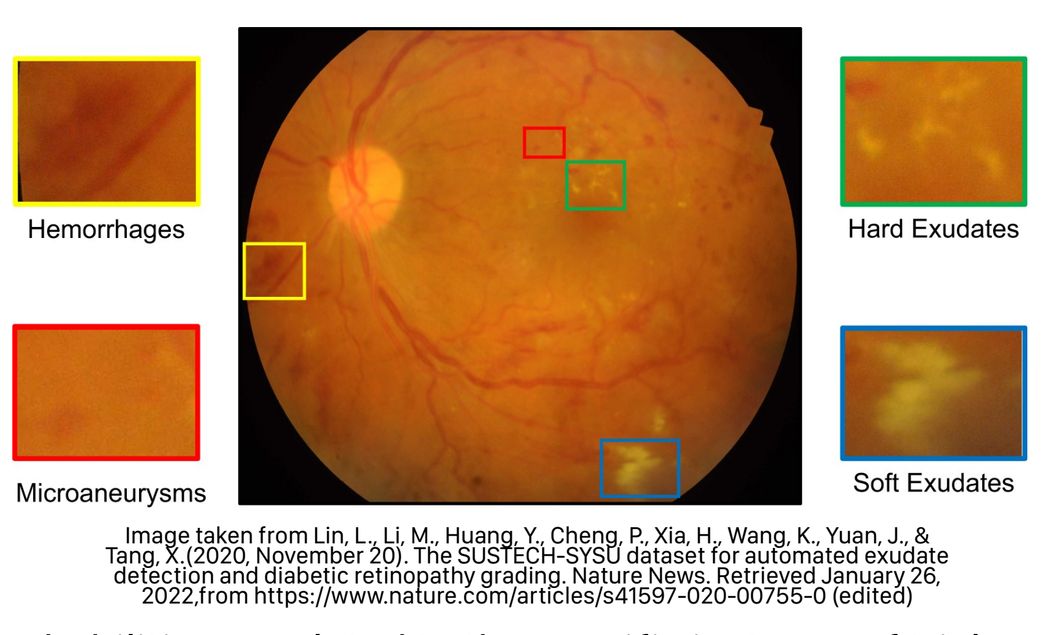


Image taken from Lin, L., Li, M., Huang, Y., Cheng, P., Xia, H., Wang, K., Yuan, J., & Tang, X.(2020, November 20). The SUSTECH-SYSU dataset for automated exudate detection and diabetic retinopathy grading. Nature News. Retrieved January 26, 2022,from https://www.nature.com/articles/s41597-020-00755-0 (edited)

Data Processing:
The data was processed through its raw form into composed tables that discerned between probabilities as related to the specific instance of trial and the severity of diabetic retinopathy (DR) provided in the sample image. From this point, the data is converted to be displayed within two radar graphs (as well as two bar graphs) that display the size/area exchange across the change in training sample sizes within each individual DR severity. The data from algorithms is then composed into two lines graphs: one displaying the raw mean accuracy across the change in training sample size and the other displaying the buffered mean accuracy across the change in training sample size.

Algorithm Design/Accuracy:
The KNN and CNN algorithms were designed to develop and run within a standard web/HTML environment. This environment was specifically selected to broaden the accessibility of the software to all modern hardware across most operating systems (although the investigators conducted all logistic experimentation on hardware of equal limitations). These algorithms were developed through TensorFlow API in tandem with the JavaScript library "ml5.js" to develop a competent web utility. The fundamental design of these algorithms was reliant upon the constraints provided by the investigators. These constraints were balanced through preliminary testing to assure essential equality of outcome. This outcome came in the form of image classification probabilities and the image datasets used for this balancing act were provided by Google in the form of the TensorFlow 64x64 Clothing dataset (the usage of this dataset was based on the recommendation of data scientists in relation to TensorFlow). This data provided a standardized reference point for how to balance these algorithms and much of the adjustment occurred in the altering of the epoch count in the CNN program. Unlike the KNN algorithm that quickly builds a ledger of logits in its training procedures, CNN undergoes a series of training trials to perfect its feature identification. The number of allowed epochs would shift the basic equality of these algorithms at a fundamental level. The researchers found that ~50 epochs within CNN training were enough to assure equality in fundamental testing. This allowed precision within the investigators' experimentation when it came to identifying areas of the difference without the bias of adjustable internal factors. This aided in securing a design that was stable in testing and worthy of true statistical comparison.

Data Testing Collection:
Prior to training the neural network models, 250 images were taken out of the sample data collected from EyePACS. In each trial, a random sample of 25 images was tested on each trained neural network for both models, for a total of 10 trials. This process was duplicated for each severity level, in total, to utilize 1250 images in testing with a combination of 50 net trials over the 5 severities.

Statistical Background:
The investigators collected a combination of nominal (type of algorithm, type of training sample size, and DR severity) and ratio data (accuracy). The type of training sample size consisted of n=125, n=250, n=375, n=500, n=1000, n=1500, n=2000, and n=2500. The stages of DR severity consisted of "no present diabetic retinopathy", "mild diabetic retinopathy", "moderate diabetic retinopathy", "severe diabetic retinopathy", and "proliferative diabetic retinopathy." The investigators' data examined the relationship between the type of algorithm and accuracy, for different training sample sizes in identifying specific stages of diabetic retinopathy in retinal images. For each type of training sample size, researchers incorporated an independent measures design in which different retinal images were used per trial, and the quantitative variable (accuracy) was measured in reference to two categorical variables (type of algorithm and DR severity). Because of this, a two-way ANOVA test (analysis of variance) with a significance level (α) of 0.05 was used to determine the experiment's statistical significance.

MATERIALS

- Retinal Dataset (EyePACS/Google Retinal Dataset)
- TensorFlow API
- Computing Utility (Chrome Capable Device)
- Data Sheet Recording Software
- External Graphing Software (Statistical Analysis)
- Online Testing Portal that allows for the Virtual Testing of the Researchers' Algorithms

EXPERIMENTATION PROGRAM

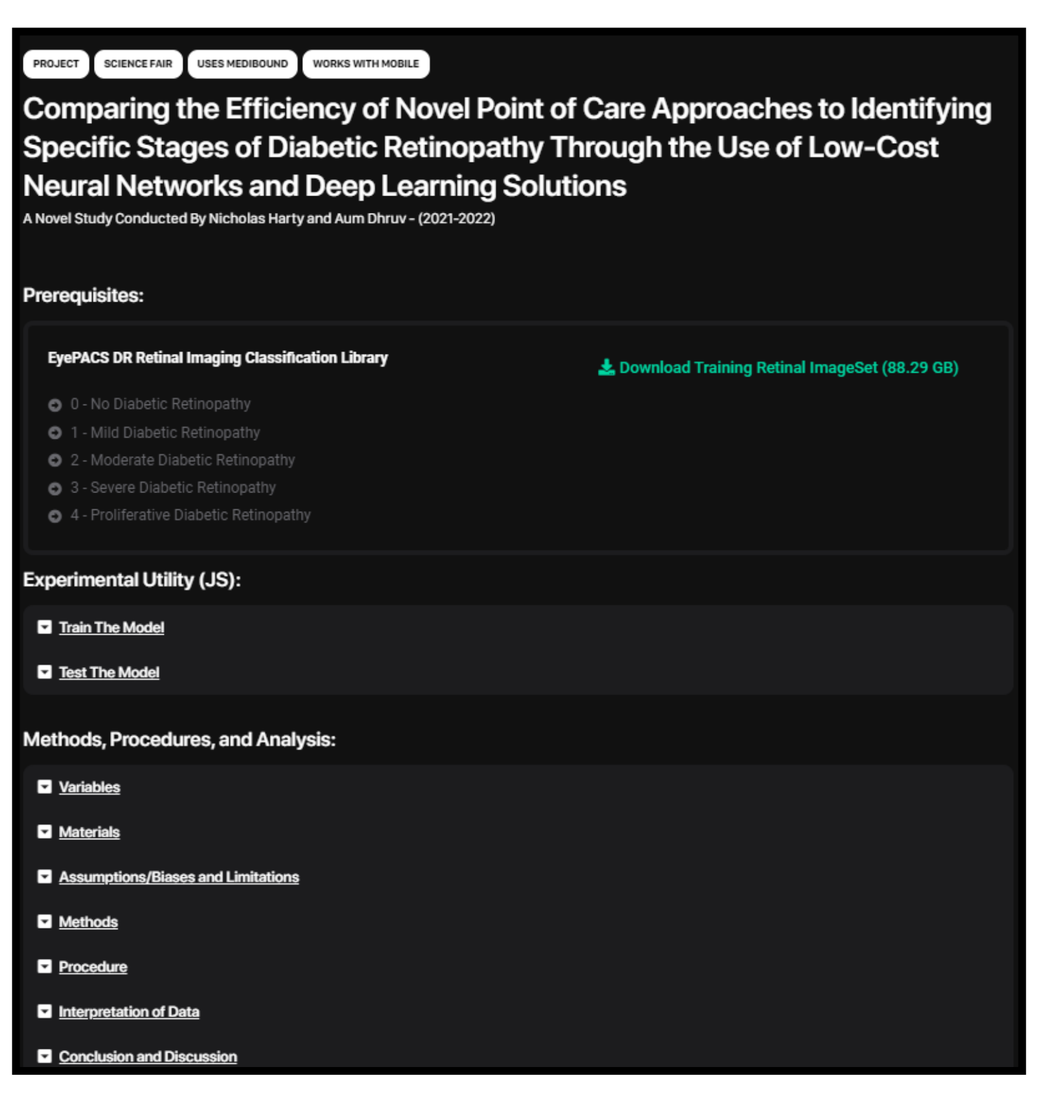


Image taken from medibound.com/project

Please see the website outlined in the procedure to view the investigators' algorithm training software

PROCEDURE

1. Researchers should prepare a modern internet computing device.
2. After doing so, open up a browser to run the neural networks. Google Chrome, Firefox, or any other browser will work.
3. Go to the Search tab and type up the link: "medibound.com/project". Click enter. This website is programmed to run and compare the available neural networks based upon an uploaded graphic of a retinal 20D image. Upload the training sample images to the "Train The Model" dropdown with an interval of n = 125, 250, 375, 500, 1000, 1500, 2000, 2500. Following this upload, activate the training module and download the appropriate Neural Network models (3 for CNN, 1 for KNN).
   a. **[OR] Alternative Ground-Up Development Plan:**
      i. Develop two contrasting neutral networks (a CNN configuration and a KNN configuration) within Kera's TensorFlow architecture creator.
      ii. Ensure that the following constraints are provided to the CNN algorithm:
         1. Debug = true
         2. Inputs = [64 (px), 64 (px), 4]
         3. Task: 'imageClassification',
         4. Epochs = 50
      iii. Ensure that the following constraints are provided to the KNN algorithm:
         1. K = Auto
         2. Debug = true
      iv. Upload the training sample images to the "Training" dropdown with an interval of n = 125, 250, 375, 500, 1000, 1500, 2000, 2500. Following this upload, activate the training module and download the appropriate Neural Network models for each interval of sample images (3 for each CNN, 1 for each KNN, 32 in total).
4. After storing the models from the training module, go to "project.medibound.com" on the web and select the "Test The Model" dropdown.
5. Insert 25 for the "Random Sample Size per Individual Trial Group (# of Images)" and distribute 250 testing retinal images into each severity category in "Testing Samples".
6. From this point, upload both of the model sets received from the training model for a specific training sample size and insert the corresponding training sample size under "Sample Size of Training Group (# of Images)"
7. Confirm that all data has been recorded. If any data is missing, redo that part of the neural network testing.
8. Repeat Steps 6 and 7 until all training sample size models are experimented upon and properly recorded.
9. Analyze and format all collected data.

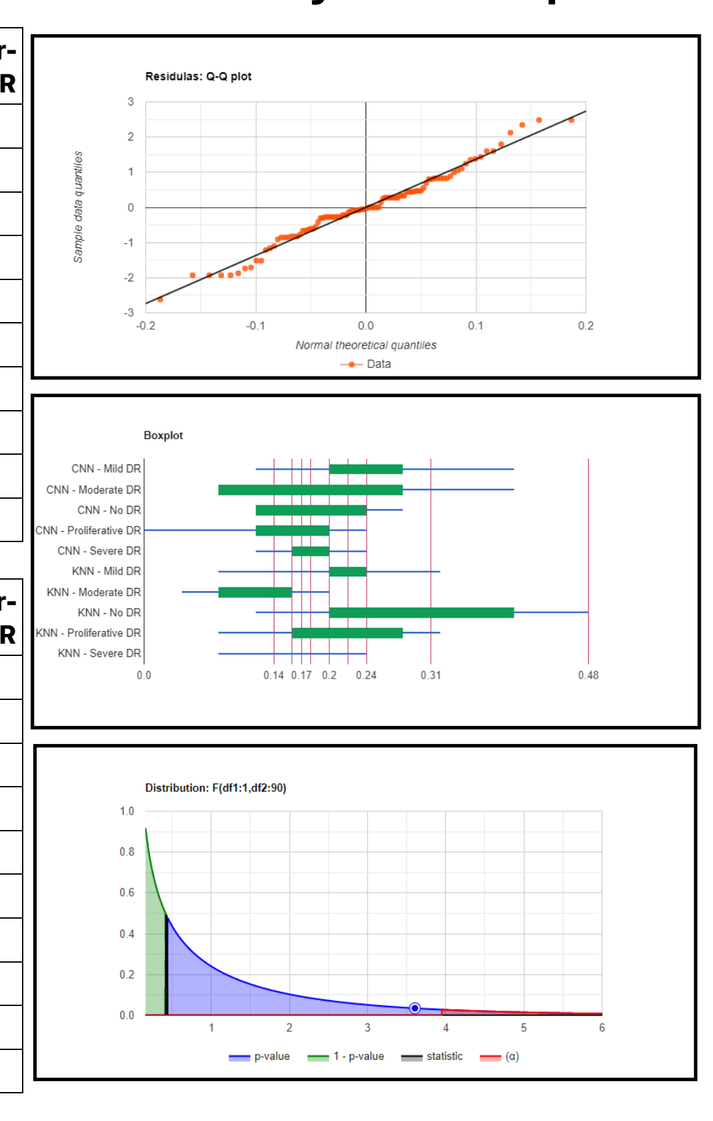
RAW DATA FROM EXPERIMENTATION

n=125 Training Sample Accuracy:

KNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.2	0.24	0.08	0.24	0.2
Trial 2	0.4	0.2	0.16	0.16	0.2
Trial 3	0.44	0.24	0.12	0.16	0.08
Trial 4	0.48	0.28	0.2	0.2	0.2
Trial 5	0.28	0.2	0.16	0.16	0.12
Trial 6	0.36	0.32	0.12	0.08	0.32
Trial 7	0.3	0.16	0.04	0.16	0.16
Trial 8	0.2	0.2	0.08	0.16	0.32
Trial 9	0.32	0.24	0.16	0.12	0.16
Trial 10	0.12	0.08	0.16	0.16	0.28

CNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.2	0.28	0.24	0.12	0.12
Trial 2	0.24	0.24	0.4	0.24	0.08
Trial 3	0.12	0.12	0.24	0.24	0
Trial 4	0.18	0.28	0.08	0.12	0.12
Trial 5	0.12	0.28	0.28	0.2	0.2
Trial 6	0.24	0.24	0.2	0.2	0.2
Trial 7	0.28	0.18	0.08	0.2	0.16
Trial 8	0.12	0.24	0.08	0.16	0.24
Trial 9	0.16	0.2	0.2	0.16	0.16
Trial 10	0.16	0.4	0.4	0.16	0.12

n=125 Two-Way ANOVA Graphs:

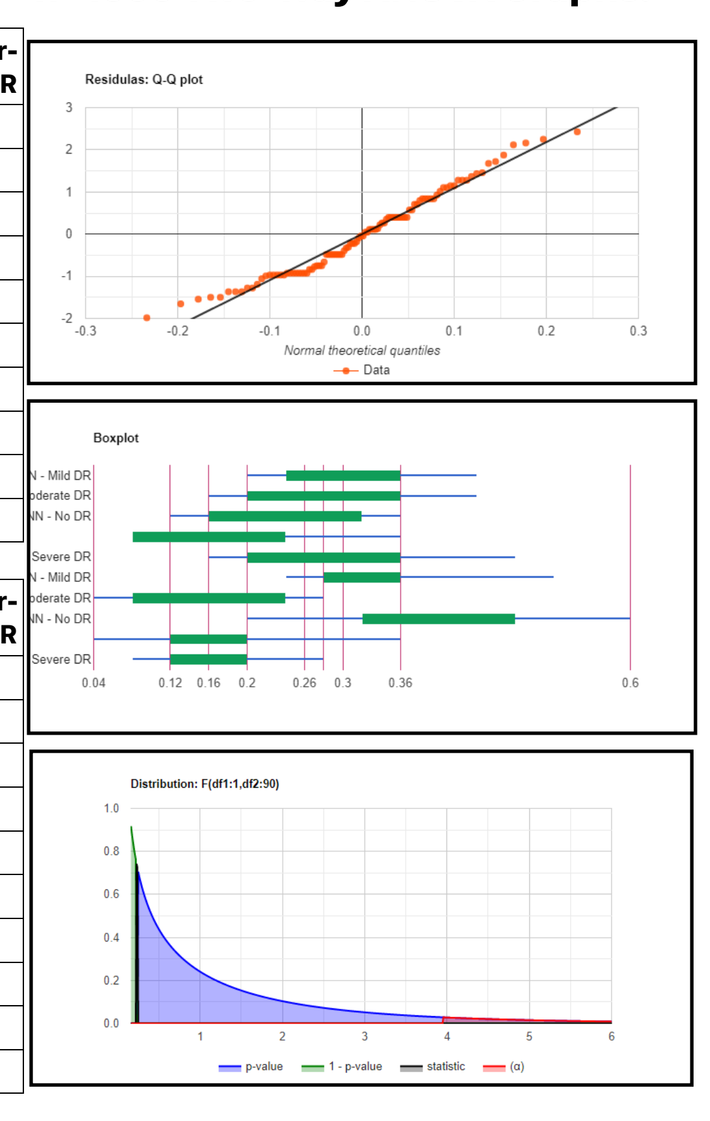


n=1000 Training Sample Accuracy:

KNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.2	0.24	0.28	0.12	0.04
Trial 2	0.36	0.28	0.2	0.08	0.2
Trial 3	0.32	0.4	0.04	0.2	0.2
Trial 4	0.48	0.36	0.04	0.2	0.12
Trial 5	0.48	0.52	0.24	0.2	0.16
Trial 6	0.6	0.28	0.2	0.28	0.2
Trial 7	0.4	0.28	0.24	0.12	0.32
Trial 8	0.36	0.28	0.16	0.08	0.36
Trial 9	0.36	0.36	0.28	0.12	0.2
Trial 10	0.24	0.24	0.08	0.24	0.1

CNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.12	0.28	0.4	0.36	0.08
Trial 2	0.28	0.4	0.44	0.2	0.32
Trial 3	0.36	0.24	0.2	0.28	0.08
Trial 4	0.16	0.44	0.16	0.16	0.08
Trial 5	0.24	0.36	0.24	0.28	0.24
Trial 6	0.32	0.2	0.36	0.16	0.16
Trial 7	0.24	0.24	0.2	0.4	0.36
Trial 8	0.32	0.32	0.32	0.2	0.08
Trial 9	0.28	0.36	0.32	0.24	0.08
Trial 10	0.12	0.24	0.2	0.48	0.2

n=1000 Two-Way ANOVA Graphs:

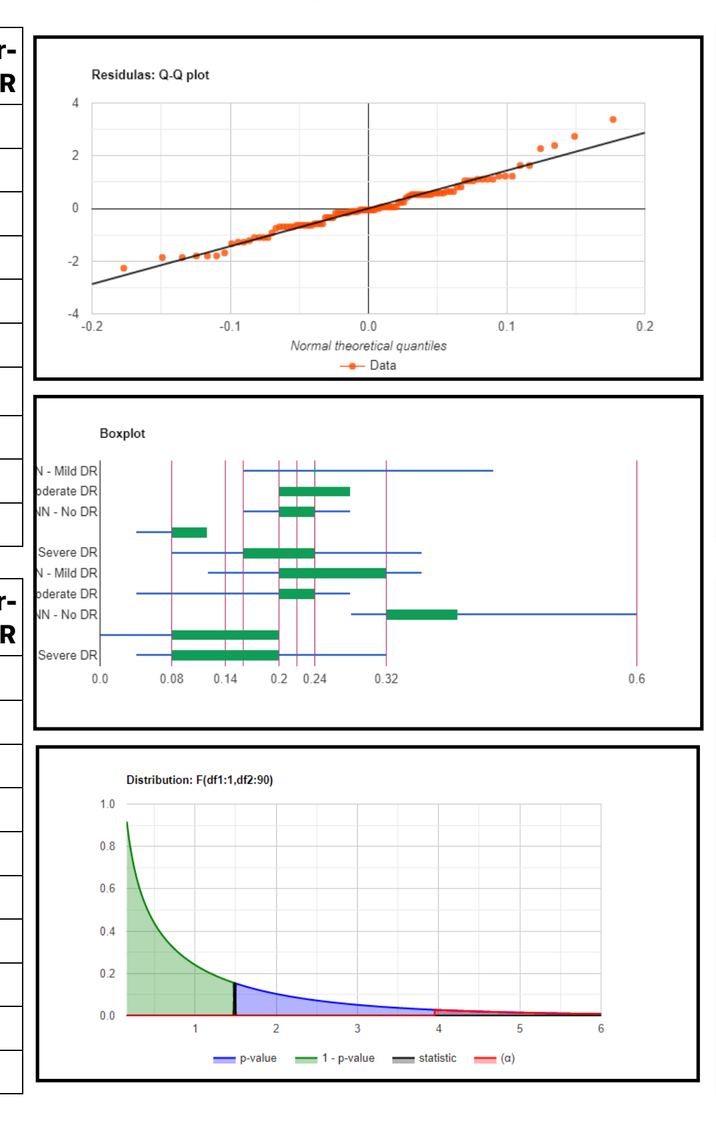


n=250 Training Sample Accuracy:

KNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.36	0.24	0.24	0.16	0.04
Trial 2	0.4	0.2	0.08	0.28	0.16
Trial 3	0.36	0.28	0.24	0.2	0.08
Trial 4	0.24	0.28	0.12	0.4	0.12
Trial 5	0.28	0.48	0.32	0.12	0.16
Trial 6	0.24	0.12	0.08	0.16	0.16
Trial 7	0.28	0.12	0.08	0.08	0.08
Trial 8	0.4	0.2	0.16	0.16	0.16
Trial 9	0.28	0.32	0.24	0.04	0
Trial 10	0.52	0.2	0.08	0.16	0.24

CNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.2	0.24	0.24	0.08	0.08
Trial 2	0.2	0.24	0.2	0.24	0.08
Trial 3	0.16	0.24	0.2	0.2	0.04
Trial 4	0.2	0.24	0.28	0.24	0.08
Trial 5	0.24	0.16	0.24	0.24	0.04
Trial 6	0.24	0.24	0.2	0.08	0.12
Trial 7	0.28	0.28	0.28	0.28	0.12
Trial 8	0.24	0.2	0.28	0.36	0.12
Trial 9	0.2	0.24	0.2	0.16	0.08
Trial 10	0.28	0.44	0.28	0.16	0.08

n=250 Two-Way ANOVA Graphs:

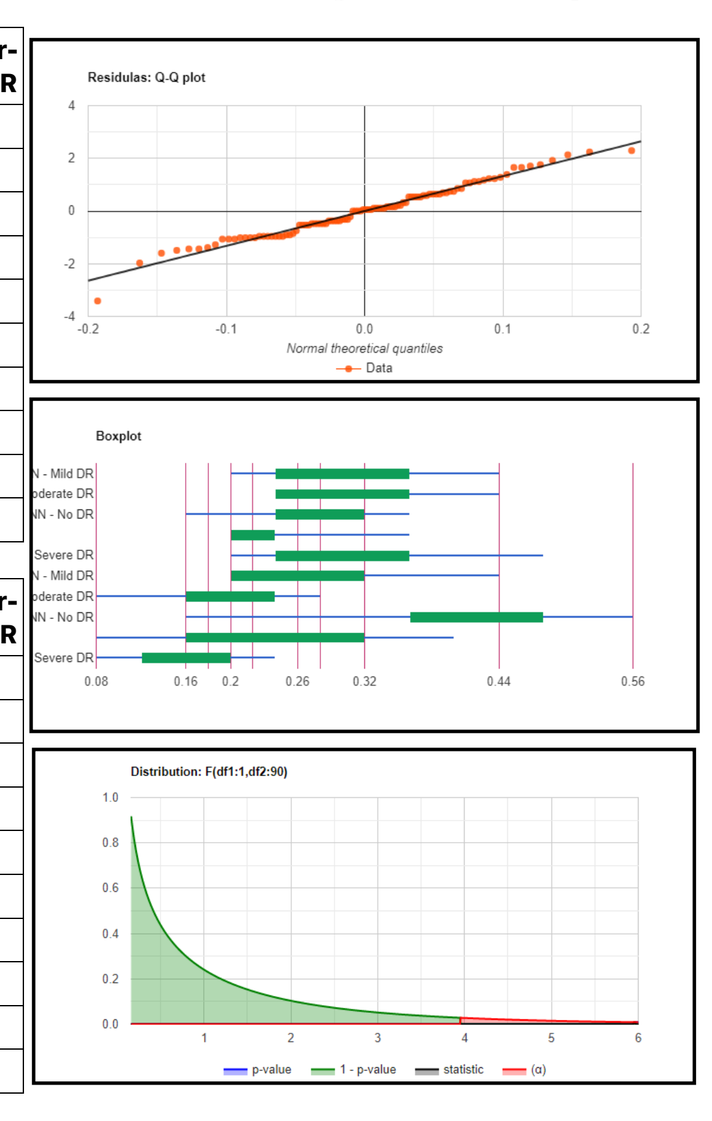


n=1500 Training Sample Accuracy:

KNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.16	0.2	0.16	0.12	0.12
Trial 2	0.4	0.28	0.24	0.16	0.16
Trial 3	0.32	0.32	0.16	0.24	0.36
Trial 4	0.44	0.2	0.08	0.12	0.08
Trial 5	0.44	0.44	0.2	0.16	0.24
Trial 6	0.48	0.2	0.28	0.2	0.16
Trial 7	0.56	0.2	0.24	0.16	0.4
Trial 8	0.48	0.28	0.16	0.08	0.32
Trial 9	0.52	0.32	0.2	0.16	0.2
Trial 10	0.36	0.28	0.12	0.2	0.24

CNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.24	0.32	0.4	0.36	0.2
Trial 2	0.28	0.4	0.44	0.24	0.32
Trial 3	0.36	0.24	0.24	0.28	0.2
Trial 4	0.16	0.44	0.36	0.36	0.2
Trial 5	0.24	0.36	0.24	0.28	0.24
Trial 6	0.32	0.2	0.36	0.36	0.2
Trial 7	0.24	0.24	0.24	0.4	0.36
Trial 8	0.32	0.32	0.32	0.2	0.2
Trial 9	0.28	0.36	0.32	0.24	0.2
Trial 10	0.24	0.24	0.24	0.48	0.24

n=1500 Two-Way ANOVA Graphs:

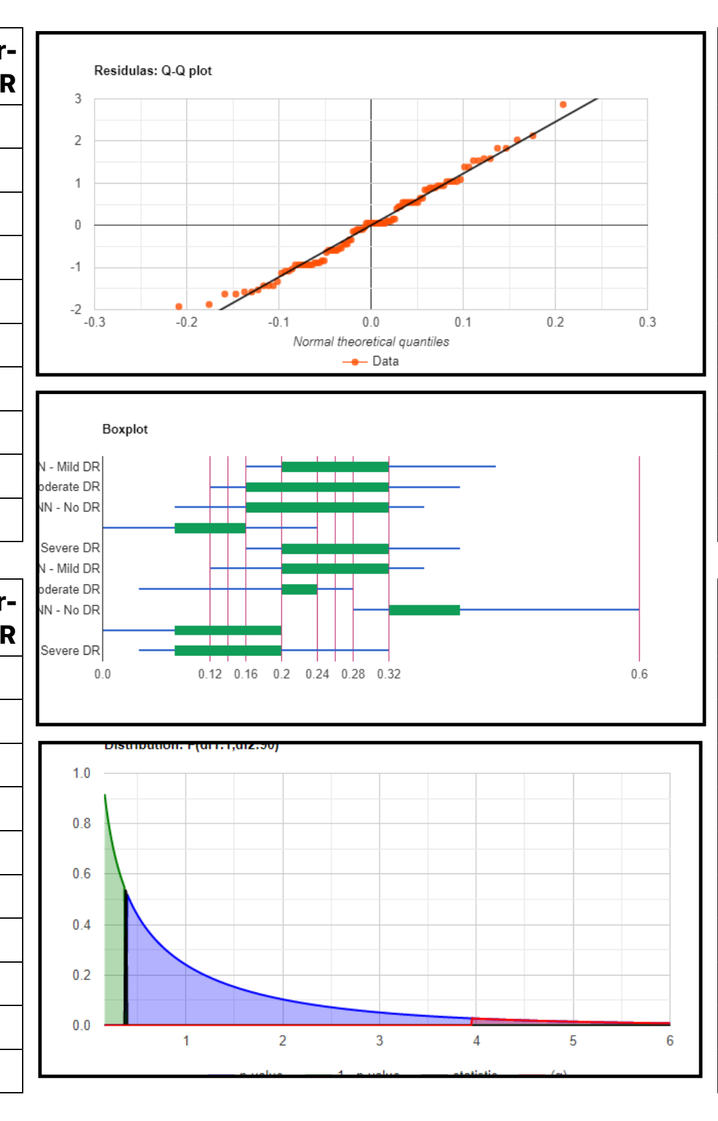


n=375 Training Sample Accuracy:

KNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.36	0.32	0.28	0.04	0
Trial 2	0.32	0.24	0.04	0.24	0.08
Trial 3	0.4	0.24	0.2	0.32	0.12
Trial 4	0.32	0.24	0.12	0.2	0.16
Trial 5	0.6	0.32	0.28	0.16	0.2
Trial 6	0.32	0.32	0.2	0.08	0.2
Trial 7	0.28	0.12	0.2	0.08	0.08
Trial 8	0.32	0.2	0.2	0.2	0.16
Trial 9	0.28	0.36	0.24	0.08	0.04
Trial 10	0.48	0.12	0.2	0.16	0.2

CNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.08	0.28	0.32	0.24	0.16
Trial 2	0.16	0.28	0.4	0.32	0.12
Trial 3	0.24	0.2	0.4	0.36	0.04
Trial 4	0.24	0.16	0.12	0.16	0.08
Trial 5	0.32	0.24	0.32	0.28	0.16
Trial 6	0.24	0.2	0.24	0.2	0.24
Trial 7	0.36	0.24	0.16	0.28	0
Trial 8	0.36	0.32	0.24	0.2	0.08
Trial 9	0.16	0.32	0.2	0.4	0.16
Trial 10	0.16	0.44	0.12	0.32	0.12

n=375 Two-Way ANOVA Graphs:

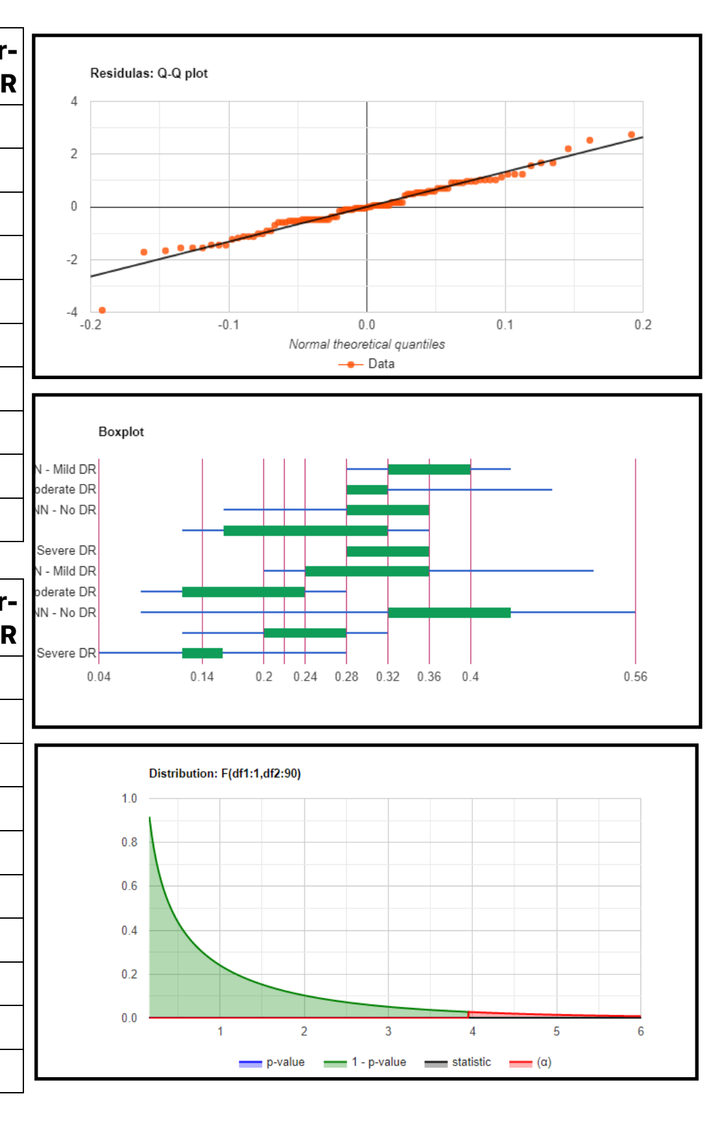


n=2000 Training Sample Accuracy:

KNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.08	0.32	0.2	0.16	0.2
Trial 2	0.44	0.4	0.2	0.12	0.2
Trial 3	0.28	0.36	0.08	0.28	0.24
Trial 4	0.36	0.32	0.08	0.12	0.12
Trial 5	0.44	0.52	0.24	0.28	0.24
Trial 6	0.44	0.24	0.28	0.12	0.16
Trial 7	0.56	0.2	0.2	0.16	0.32
Trial 8	0.36	0.24	0.12	0.04	0.32
Trial 9	0.44	0.36	0.24	0.12	0.2
Trial 10	0.32	0.2	0.24	0.16	0.28

CNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.32	0.32	0.28	0.36	0.32
Trial 2	0.28	0.44	0.48	0.28	0.28
Trial 3	0.28	0.36	0.28	0.32	0.16
Trial 4	0.16	0.4	0.28	0.36	0.12
Trial 5	0.36	0.28	0.28	0.28	0.36
Trial 6	0.28	0.32	0.32	0.36	0.24
Trial 7	0.36	0.36	0.28	0.28	0.24
Trial 8	0.2	0.32	0.32	0.28	0.16
Trial 9	0.28	0.4	0.28	0.36	0.24
Trial 10	0.36	0.44	0.36	0.32	0.32

n=2000 Two-Way ANOVA Graphs:

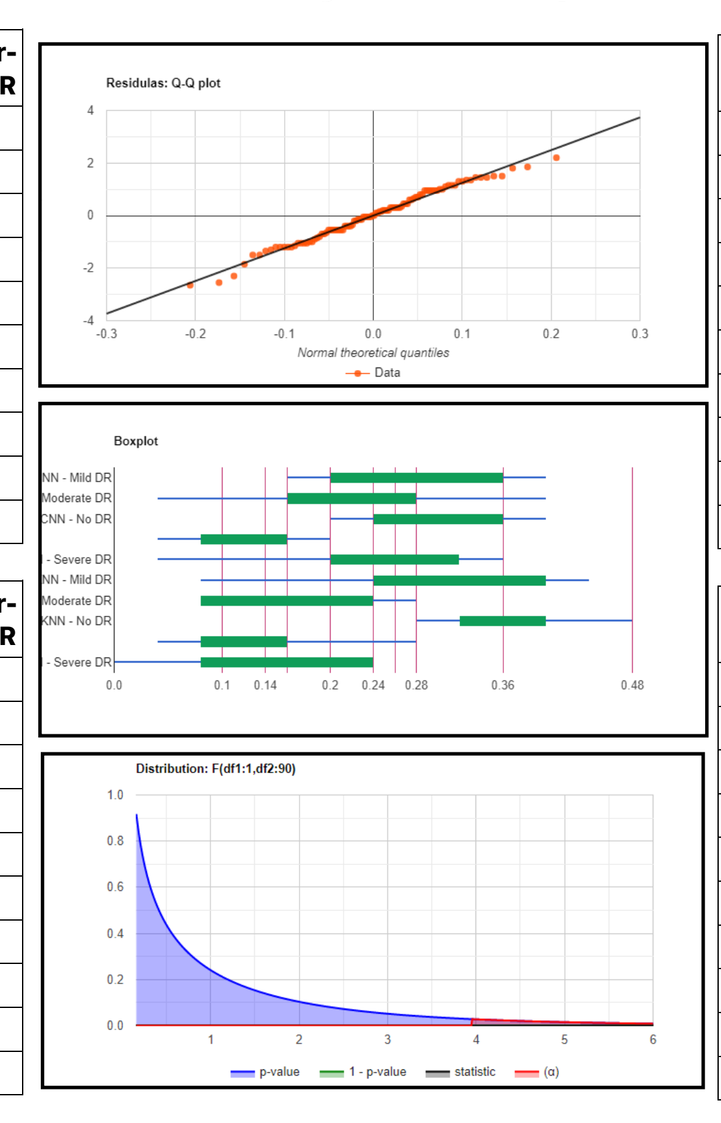


n=500 Training Sample Accuracy:

KNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.4	0.32	0.2	0	0.04
Trial 2	0.4	0.28	0.16	0.24	0.12
Trial 3	0.48	0.2	0.2	0.24	0.16
Trial 4	0.32	0.4	0.08	0.24	0.04
Trial 5	0.44	0.44	0.28	0.08	0.24
Trial 6	0.36	0.28	0.24	0.2	0.16
Trial 7	0.28	0.08	0.2	0.04	0.28
Trial 8	0.28	0.24	0.08	0.16	0.08
Trial 9	0.32	0.4	0.24	0.16	0.08
Trial 10	0.36	0.28	0.08	0.12	0.16

CNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.24	0.16	0.4	0.32	0.12
Trial 2	0.4	0.4	0.24	0.24	0.16
Trial 3	0.36	0.28	0.28	0.32	0.08
Trial 4	0.28	0.2	0.16	0.2	0.08
Trial 5	0.2	0.2	0.24	0.24	0.16
Trial 6	0.24	0.16	0.04	0.2	0.08
Trial 7	0.36	0.28	0.24	0.04	0.12
Trial 8	0.24	0.4	0.24	0.32	0.08
Trial 9	0.32	0.36	0.28	0.36	0.04
Trial 10	0.2	0.36	0.12	0.2	0.2

n=500 Two-Way ANOVA Graphs:

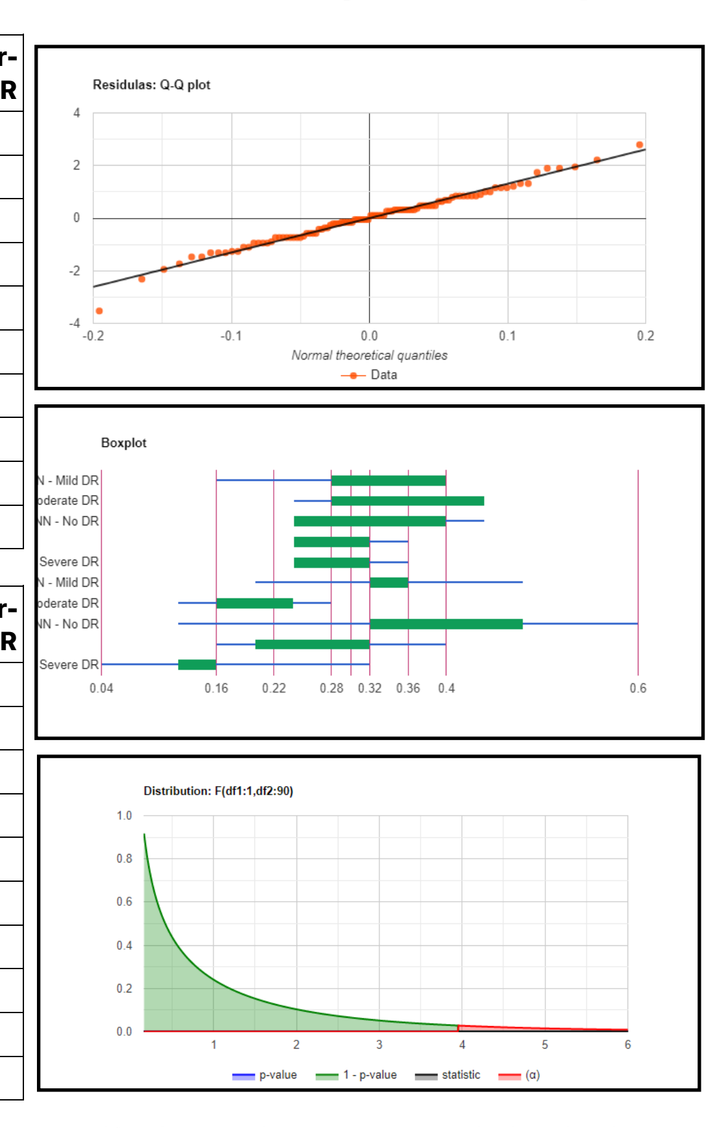


n=2500 Training Sample Accuracy:

KNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.12	0.32	0.2	0.16	0.4
Trial 2	0.32	0.32	0.24	0.12	0.4
Trial 3	0.36	0.32	0.12	0.16	0.16
Trial 4	0.44	0.48	0.12	0.08	0.2
Trial 5	0.44	0.4	0.24	0.32	0.28
Trial 6	0.48	0.32	0.28	0.16	0.2
Trial 7	0.6	0.2	0.16	0.2	0.16
Trial 8	0.36	0.32	0.2	0.12	0.24
Trial 9	0.52	0.28	0.24	0.04	0.2
Trial 10	0.24	0.36	0.24	0.16	0.32

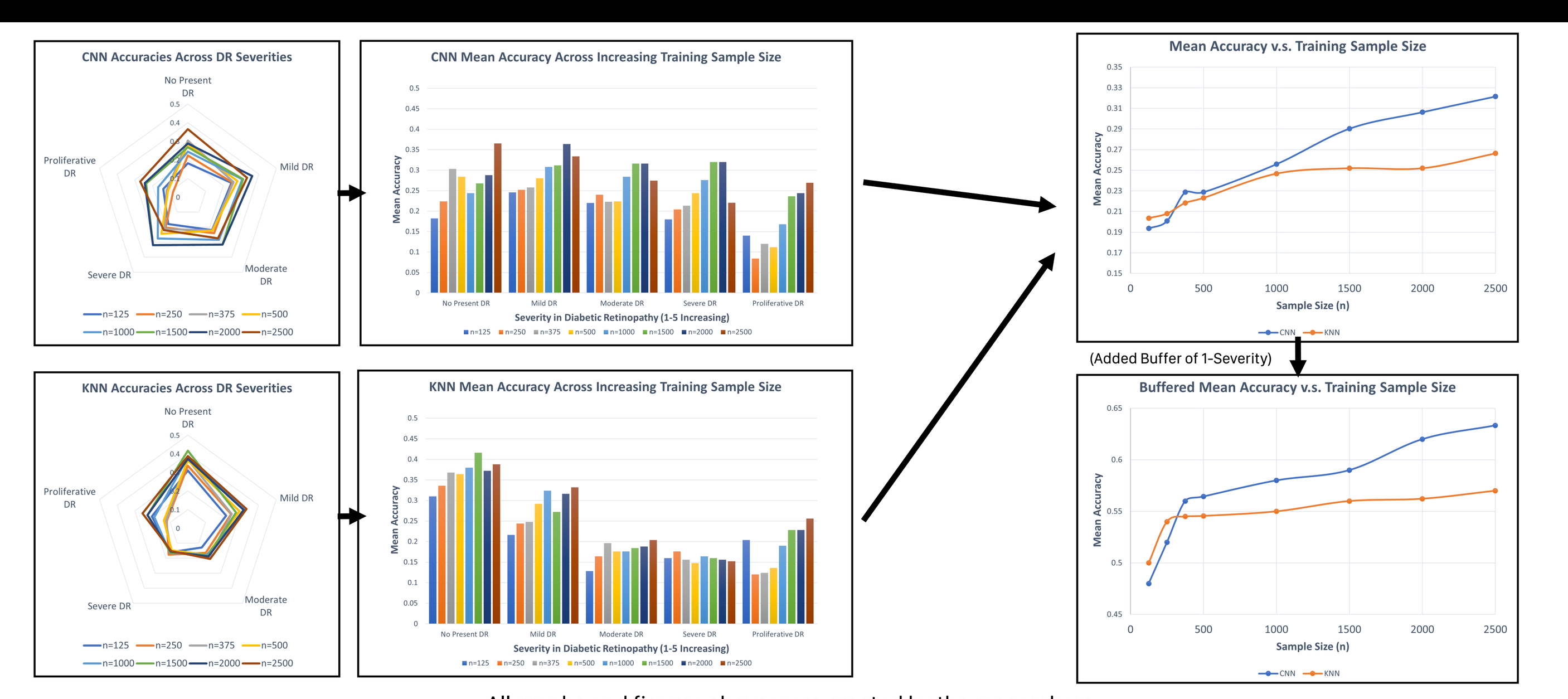
CNN	No Present DR	Mild DR	Moderate DR	Severe DR	Proliferative DR
Trial 1	0.24	0.36	0.28	0.32	0.24
Trial 2	0.4	0.4	0.4	0.24	0.28
Trial 3	0.36	0.28	0.28	0.32	0.32
Trial 4	0.44	0.36	0.28	0.36	0.28
Trial 5	0.36	0.4	0.24	0.28	0.24
Trial 6	0.24	0.16	0.44	0.24	0.28
Trial 7	0.36	0.28	0.36	0.28	0.28
Trial 8	0.24	0.4	0.44	0.32	0.24
Trial 9	0.32	0.36	0.44	0.36	0.32
Trial 10	0.44	0.36	0.36	0.24	0.36

n=2500 Two-Way ANOVA Graphs:



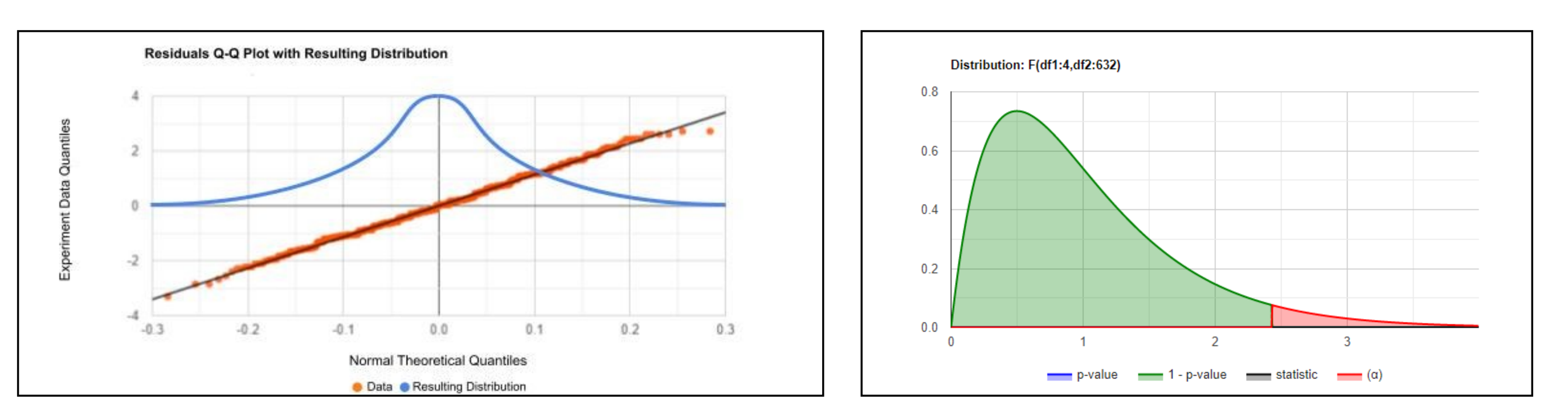
All graphs and figures above were created by the researchers with graphical assistance from the source: Two Way ANOVA Calculator. (n.d.). Retrieved January 23, 2022, from https://www.statskingdom.com/two-way-anova-calculator.html

OVERALL GRAPHICAL ANALYSIS OF DATA



All graphs and figures above were created by the researchers

OVERALL STATISTICAL ANALYSIS OF DATA



All graphs and figures above were created by the researchers with graphical assistance from the source: Two Way ANOVA Calculator. (n.d.). Retrieved January 23, 2022, from https://www.statskingdom.com/two-way-anova-calculator.html

CONCLUSION AND DISCUSSION

When presented with training sample sizes of 125 and 250 retinal images, KNN was more accurate, which could justify its use over CNN in lower sample sizes. However, in training sample sizes of n=375 and above, CNN was more accurate, which could justify its use over KNN in higher sample sizes. This supported the investigators' hypothesis since the CNN algorithm produced a greater average accuracy across higher training sample sizes in this identification process compared to the KNN algorithm, which produced greater average accuracy across lower training sample sizes. At the largest training sample size of 2500 retinal images, the CNN was over 6% higher in accuracy compared to KNN. This signifies that CNN's accuracy could continue growing at a higher rate with additional training samples. This acknowledgment could prove to be beneficial to larger researchers with worldwide resources.

For each training sample size, the two-way ANOVA test conducted by the researchers generated p-values less than the significance level of 0.05, which meant the null hypothesis, that there is no statistical significance between the type of algorithm and its average accuracy across training sample sizes, was rejected. From this, it can be concluded that there is a statistically significant relationship between the type of algorithm and accuracy in identifying specific stages of diabetic retinopathy in retinal images. This means that the type of algorithm played a role in its accuracy, and wasn't due to chance, proving that the results are likely valid.

These results concur with the community of global issues for which this study was conducted upon. The research and findings of this study will directly aid low-income exhibitions that seek widespread testing for the preventable disease that is diabetic retinopathy. With the data in relation to KNN and CNN, as far as ambiguity in the outcome, the investigators have provided some certainty to the usage of such efficient algorithms within medical treatment. Although ambiguity still lies in the perfection of accuracy within the two computing models, the researchers have provided a defining line between the necessary usage of KNN and CNN in the field. When granted further retinal samples, this study recommends the development of a convolutional neural network to best discern between DR severities. However, when granted fewer retinal samples (n≤250), this study suggests k-nearest neighbors as the more accurate, although not precise, algorithm to best discern between DR severities.

INTERPRETATION OF DATA

Results:
To measure the accuracy of the CNN and KNN algorithms, mean and buffered accuracy was calculated for each training sample size. For both algorithms, the mean accuracy (%) was the average of all 50 trials (10 per DR severity). The buffered accuracy (%) provided a 1-severity margin to either end of the definite outcome, resulting in an amplified marker of the mean accuracy. In a training sample size of 125 retinal images (ranging in severity), the CNN algorithm resulted in a mean accuracy of 19.36% and a buffered accuracy of 48.00%, while the KNN algorithm resulted in a mean accuracy of 20.36% and a buffered accuracy of 50.00%. In a training sample size of 250 retinal images, the CNN algorithm resulted in a mean accuracy of 20.08% and a buffered accuracy of 52.00%, while the KNN algorithm resulted in a mean accuracy of 20.80% and a buffered accuracy of 54.00%. In a training sample size of 375 retinal images, the CNN algorithm resulted in a mean accuracy of 22.88% and a buffered accuracy of 56.00%, while the KNN algorithm resulted in a mean accuracy of 21.84% and a buffered accuracy of 54.50%. In a training sample size of 500 retinal images, the CNN algorithm resulted in a mean accuracy of 22.88% and a buffered accuracy of 56.44%, while the KNN algorithm resulted in a mean accuracy of 22.32% and a buffered accuracy of 54.56%. In a training sample size of 1000 retinal images, the CNN algorithm resulted in a mean accuracy of 25.60% and a buffered accuracy of 58.00%, while the KNN algorithm resulted in a mean accuracy of 24.68% and a buffered accuracy of 55.00%. In a training sample size of 1500 retinal images, the CNN algorithm resulted in a mean accuracy of 29.04% and a buffered accuracy of 59.00%, while the KNN algorithm resulted in a mean accuracy of 25.20% and a buffered accuracy of 56.00%. In a training sample size of 2000 retinal images, the CNN algorithm resulted in a mean accuracy of 30.64% and a buffered accuracy of 62.00%, while the KNN algorithm resulted in a mean accuracy of 25.20% and a buffered accuracy of 56.22%. In a training sample size of 2500 retinal images, the CNN algorithm resulted in a mean accuracy of 32.16% and a buffered accuracy of 63.33%, while the KNN algorithm resulted in a mean accuracy of 26.64% and a buffered accuracy of 57.00%.

Trends:
In both algorithms, there was a general increase in accuracy as the training sample size increased. In the CNN algorithm, this increase was steeper compared to the KNN algorithm. This trend can be noted in the slope of the provided line graphs. In addition to this, the radar graphs and bar graphs highlighted individual upward trends in each category of diabetic retinopathy by increasing sample size. From the line graphs, the pattern of CNN's spike is evident as the CNN mean accuracy line makes a sharp increase at around 375 samples and in doing so, surpasses the mean accuracy of KNN.

Significance:
When presented with training sample sizes of 125 and 250 retinal images, KNN was more accurate, which could justify its use over CNN in lower sample sizes. However, in training sample sizes of n=375 and above, CNN was more accurate, which could justify its use over KNN in higher sample sizes. This supported the investigators' hypothesis since the CNN algorithm produced a greater average accuracy across higher training sample sizes in this identification process compared to the KNN algorithm, which produced greater average accuracy across lower training sample sizes. At the largest training sample size of 2500 retinal images, the CNN was over 6% higher in accuracy compared to KNN. This could signify that CNN's accuracy would continue growing at a higher rate with additional training samples. This acknowledgment could prove to be beneficial to larger researchers with worldwide resources.

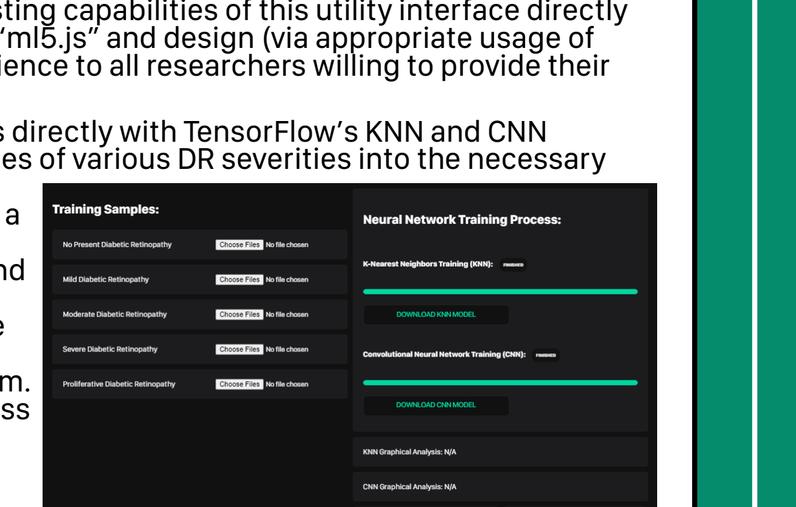
Statistical Analysis:
The investigators collected a combination of nominal (type of algorithm, type of training sample size, and DR severity) and ratio data (accuracy). Their data examined the relationship between the type of algorithm and accuracy, for different training sample sizes in identifying specific stages of diabetic retinopathy in retinal images. For each type of training sample size, researchers incorporated an independent measures design in which different retinal images were used per trial, and the quantitative variable (accuracy) was measured in reference to two categorical variables (type of algorithm and DR severity). Because of this, a two-way ANOVA test (analysis of variance) with a significance level (α) of 0.05 was used to determine the experiment's statistical significance. For a training sample size of 125 retinal images, the ANOVA test generated an interaction p-value of 0.0001347. For a training sample size of 250 retinal images, the ANOVA test generated an interaction p-value of 0.0002419. For a training sample size of 375 retinal images, the ANOVA test generated an interaction p-value of 0.0001786. For a training sample size of 500 retinal images, the ANOVA test generated an interaction p-value of 0.01463. For a training sample size of 1000 retinal images, the ANOVA test generated an interaction p-value of 0.000272. For a training sample size of 1500 retinal images, the ANOVA test generated an interaction p-value of 0.000077. For a training sample size of 2000 retinal images, the ANOVA test generated an interaction p-value of 0.00001857. For a training sample size of 2500 retinal images, the ANOVA test generated an interaction p-value of 0.0002475. All of the p-values from the ANOVA test were less than the significance level of 0.05, which meant the null hypothesis, that there is no statistical significance between the type of algorithm and its average accuracy across training sample sizes, was rejected. This signifies that there is a statistically significant relationship. In addition to this, the residuals followed a normal distribution as shown by the Q-Q plot. Furthermore, every interaction priori power generated by the ANOVA test was greater than 0.9850, signifying that the results are likely valid.

EMBEDDING THE ALGORITHMS

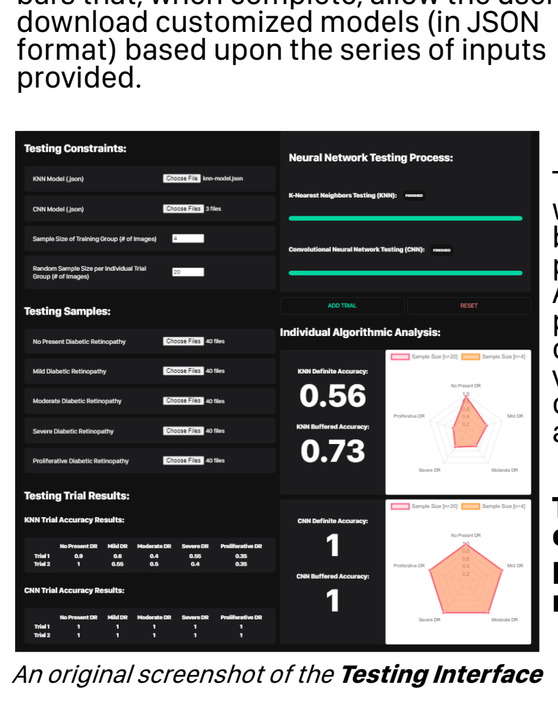
The main function of the experimentation, the two neural network algorithms, are hosted for public use at **medibound.com/project.** On this site, the investigators have also posted a link to download our retinal database from EyePACS that provides the appropriate distinction between varying DR severities. The researchers encourage any viewers of this project to repeat our study and solidify/contest our results for the benefit of all point-of-care researchers across the globe.

These utilities are provided under the "Train The Model" and "Test The Model" tabs on the web application. Both of these utilities were developed by the researchers entirely through web-native JavaScript. The training and testing capabilities of this utility interface directly with TensorFlow API through the JS library "ml5.js" and design (via appropriate usage of CSS) to provide a familiar/accessible experience to all researchers willing to provide their time to the training process.

The "Train The Model" utility tab interfaces directly with TensorFlow's KNN and CNN subscript by transferring input retinal images of various DR severities into the necessary inputs for each algorithm. For example, the researcher's web-JS program breaks down a sample retinal image into both a matrix of logits specifically for KNN preprocessing and a compressed RBG(A) image array for CNN preprocessing. From this point forward, the program sustains information as to the scheduling of the training for each algorithm. This progress is visualized via green progress bars that, when complete, allow the user to download customized models (in JSON format) based upon the series of inputs provided.



An original screenshot of the ***Training Interface***



An original screenshot of the ***Testing Interface***

The "Test The Model" utility tab interfaces directly with TensorFlow's KNN and CNN subscript by breaking down input models (from the previous process) and submitting them to the TensorFlow API (through "ml5.js") as prerequisites. the program sustains information as to the scheduling of the testing for each algorithm. This progress is visualized via green progress bars that, when complete, allow the user to view graphical analysis and structured raw table data.

The investigators encourage the evaluation of our internal program as provided at: https://github.com/medibound/medibound.github.io

Image taken from medibound.com/project

REFERENCES

- Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A., & Tan, R. S. (2017, August 24). A deep convolutional neural network model to classify heartbeats. Computers in Biology and Medicine. Retrieved December 29, 2021, from https://www.sciencedirect.com/science/article/pii/S0010482517302810?casa_token=ufM7dWVQvzIAAAAA%3A6Q7840z1UyqkEDSc5HR-GebPpdSi5el09IxikoaeQFj8SVlkSGdhFVaoG7bjfvYfMl7LgaxmaBj0
- Gulshan V;Peng L;Coram M;Stumpe MC;Wu D;Narayanaswamy A;Venugopalan S;Widner K;Madams T;Cuadros J;Kim R;Raman R;Nelson PC;Mega JL;Webster DR; (n.d.). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA. Retrieved January 23, 2022, from https://pubmed.ncbi.nlm.nih.gov/27898976/
- K-nearest neighbor algorithm: Topics by science.gov. (n.d.). Retrieved January 23, 2022, from https://www.science.gov/topicpages/k/k-nearest+neighbor+algorithm
- Lam, C., Yi, D., Guo, M., & Lindsey, T. (2018, May 18). Automated detection of diabetic retinopathy using Deep Learning. AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science. Retrieved January 23, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5961805/
- Nentwich, M. M., & Ulbig, M. W. (2015, April 15). Diabetic retinopathy - ocular complications of diabetes mellitus. World journal of diabetes. Retrieved January 23, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4398904/
- Peterson, L. E. (n.d.). K-Nearest Neighbor. Scholarpedia. Retrieved December 29, 2021, from http://www.scholarpedia.org/article/K-nearest_neighbor
- Saha, S. (2018, December 17). A comprehensive guide to convolutional neural networks-the eli5 way. Medium. Retrieved December 29, 2021, from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- Thorat, N. (2018, June 20). How to build a teachable machine with Tensorflow.js. Observable. Retrieved December 29, 2021, from https://observablehq.com/@nsthorat/how-to-build-a-teachable-machine-with-tensorflow-js)
- Two Way ANOVA Calculator. (n.d.). Retrieved January 23, 2022, from https://www.statskingdom.com/two-way-anova-calculator.html
- Zhang, Z. (2016, June). Introduction to machine learning: K-Nearest Neighbors. Annals of translational medicine. Retrieved January 23, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4916348/
- Lin, L., Li, M., Huang, Y., Cheng, P., Xia, H., Wang, K., Yuan, J., & Tang, X. (2020, November 20). The SUSTECH-SYSU dataset for automated exudate detection and diabetic retinopathy grading. Nature News. Retrieved January 26, 2022, from https://www.nature.com/articles/s41597-020-00755-0

FURTHER / FUTURE RESEARCH

The data analysis and resulting accuracies of the past year of experimentation suggests that the utilization of a CNN Algorithm, in categorizing diabetic retinopathy (DR) cases, can produce greater accuracy of results when given high initial sample sets (high fidelity retinal images). Given that the platform of this experimentation, TensorFlow, is a low-cost, efficient alternative to most in-field medical solutions. These factors, in addition to TensorFlow's available usage within web applications (as seen within the investigators' research/experimentation), allow for tools with similar architecture to achieve a wide range of accessibility. This nuanced accessibility allows for outreach that wasn't possible in the twentieth century and should, throughout the course of the next decade, encourage grassroots efforts to combat preventable retinal diseases, like diabetic retinopathy, in low-income areas where they are abundant. Applications in pursuit of such worldwide retinal health can already be seen in efforts such as Google-EyePACS funding efforts towards efficient DR Algorithmic development and, in the medical hardware industry, by ODACS to provide affordable self-utilized equipment to record retinal images/samples. Although KNN, throughout the experiment, only provided greater accuracy in low training sample scenarios, this experiment may, with further research, justify the use of KNN in lesser strenuous fields of study. These fields could include basic color qualifications and instances where training samples are extremely difficult to collect. With these applications in mind, the second year of the investigators' study could explore the external factors of retina imaging through physical experimentation that was inaccessible during the global pandemic. These factors could include on-site retinal imaging through modern point-of-care (POC) utilities and greater physical variance across trials. This variance within independent variables could come in the form of adjusted retinal field-of-view (20D, 30D, 90D) and lens distance when studying/qualifying the previous accuracy of neural networks.