

# An intelligent system for the detection of ophthalmic features from colour fundus photography

Full Name	Student ID	Email
Zhong Xiaohui	A0249305E	e0938910@u.nus.edu
Lim Chang Siang	A0176266W	changsiang.lim@u.nus.edu
Li Zhenghao, Kelvin	A0031400J	e0396343@u.nus.edu
Zheng Xiaolan	A0249271B	e0938876@u.nus.edu

## Table of Contents

<b>Executive Summary.....</b>	<b>4</b>
<b>Problem Description .....</b>	<b>5</b>
<b>Project Objective .....</b>	<b>6</b>
<b>Aim.....</b>	<b>6</b>
<b>Objective .....</b>	<b>6</b>
<b>Data Set.....</b>	<b>7</b>
<b>Data Preparation and Data Pre-processing .....</b>	<b>7</b>
<b>Data Labelling .....</b>	<b>7</b>
<b>Exploration Data Analysis .....</b>	<b>8</b>
Descriptive Statistics.....	8
<b>Image Pre-processing.....</b>	<b>9</b>
Data Trimming .....	9
Black area re-fill .....	9
Normalization .....	9
Histogram Equalization.....	9
CLANE Histogram Equalization .....	10
Summary of image pre-processing steps.....	10
<b>Data Augmentation methods.....</b>	<b>10</b>
Flip .....	10
Rotation .....	11
Brightness .....	12
Shift.....	12
Zoom.....	13
Process Image data into numbers .....	13
Data pipeline.....	14
<b>Model Design.....</b>	<b>15</b>
<b>Bounding Box Detection Approach.....</b>	<b>15</b>
Sliding Box method .....	15
Anchor box method .....	15
<b>Machine Learning Training with Entire Data Set.....</b>	<b>15</b>
Support Vector Machine.....	15
Random Forest Classifier .....	17
k-nearest Neighbour .....	18
Linear Discriminant Analysis.....	19
<b>Machine Learning Training with Overrepresented Data Set.....</b>	<b>20</b>
Support Vector Machine.....	20
Random Forest Classifier .....	22
k-Nearest Neighbour Classifier .....	23
Linear Discriminant Analysis .....	24
<b>Machine Learning Training with Underrepresented Data Set .....</b>	<b>25</b>
Support Vector Machine.....	25
Random Forest Classifier .....	27
k-Nearest Neighbour Classifier .....	28
Linear Discriminate Classifier.....	29

<b>Summary on supervised machine learning approach.....</b>	<b>30</b>
<b>Neural Network Approach .....</b>	<b>31</b>
ResNet50 + Anchor box .....	31
Darknet + YOLO.....	35
<b>Diagnosis Reasoning System.....</b>	<b>37</b>
<b>System Architecture.....</b>	<b>38</b>
System Architecture.....	38
System Setup .....	38
<b>Conclusion .....</b>	<b>40</b>
<b>References.....</b>	<b>41</b>

## Executive Summary

Healthcare is undergoing a major transformation from reactive care to a more proactive approach. For this purpose, Singapore has recently announced a major healthcare initiative focusing on preventive care: Healthier SG [1, 2]. Amongst the five key features, Healthier SG aims to mobilise family doctors to deliver preventive care and develop health plans for regular health screening.

However, medical care is segmented into specialised areas of expertise. Ophthalmology is one such area where primary care physicians will often seek the help of the Ophthalmologists for diagnosis and management. This increases the time and cost for patient before he or she receives treatment.

Our group project addresses the need for accessible rapid eye diagnosis. Our system identifies salient clinical features via a colour fundus photo to infer a clinical diagnosis and presents it to the physician. This will allow the primary care physician to better determine whether the patient requires a specialist consultation.

## Problem Description

Visual impairment is a global burden with the annual global costs of productivity losses associated with vision impairment estimated to be US\$ 411 billion [3]. The main causes of visual impairment are age-related macular degeneration (AMD), diabetic retinopathy (DR), glaucoma and cataract. Regular eye screening and disease monitoring is important in the prevention of these diseases. For AMD, DR, and glaucoma, the observable changes are seen in the fundus. Fundus examination is traditionally done using a slit-lamp by an Ophthalmologist.

In the early stages, fundus photography was done on film and had the purpose of documentation. However, in recent years, with digital fundus photography becoming more accessible, screening of eye conditions has been done using digital fundus photography. In Singapore, this is evident with Singapore integrated Diabetic Retinopathy Programme which was started in 2010 [4].

Artificial intelligence, and more specifically, deep learning, has been found to be useful in analysing fundus photos to classify disease states. A foundational step in eye diagnosis is the identification of clinical eye features, both physiological and pathological. Most of the published retinal deep learning algorithm generates class labels using the final eye diagnosis but does not have enough granularity in identifying clinical features on the fundus [5, 6]. This has implications on the explainability of the model, and the ability for the model to adapt to new clinical grading systems.

## Project Objective

### Aim

- We aim to design an eye diagnosis system to allow users to obtain basic eye diagnoses by uploading a pair of ophthalmic colour fundus photos.

### Objective

- To employ the use of machine learning and deep learning techniques to achieve automated identification and labelling key features of the fundus image.
- To design and develop an inference system that can provide diagnosis recommendation based on the features found in the fundus image.
- To design a user interface that allows the detected clinical features to be presented, stored and exported for future comparison.

## Data Set

Colour fundus photos were obtained from the Rotterdam Ophthalmic Data Repository- Longitudinal diabetic retinopathy screening fundus photos [7]. This data set contains repeated 4-field colour fundus photos of 70 patients in a diabetic retinopathy screening program. As both eyes were imaged, over a total of 2 visits, we obtained 1120 fundus images for training. These images were reviewed by Dr Kelvin Li, one of the team members who works as an Associate Consultant Ophthalmologist from TTSH, and who is also a master grader for the Singapore integrated Diabetic Retinopathy Programme. Each image was reviewed for the presence or absence of disease and its corresponding features. These include diabetic retinopathy and its stages, epiretinal membrane, choroidal nevus, retinal vein occlusion, etc. Where there were changes between visits, these were also noted.

## Data Preparation and Data Pre-processing

### Data Labelling

The images were labelled using *labelImg* [8], an image labelling application. Images were reviewed one at a time. Image labelling was performed by Dr Kelvin Li, and his team of residents (Dr Low Kok Yao, Dr Marcus Goh), and a medical student (Ms Natalie Tan). Anatomical features such as optic disc, optic cup and fovea, and pathological features such as blot hemorrhage, dot hemorrhages, were both labelled and saved as an xml file. Where the images were deemed unsuitable for labelling or did not show any disease features (for example, in peripheral fundus view), these were not labelled.

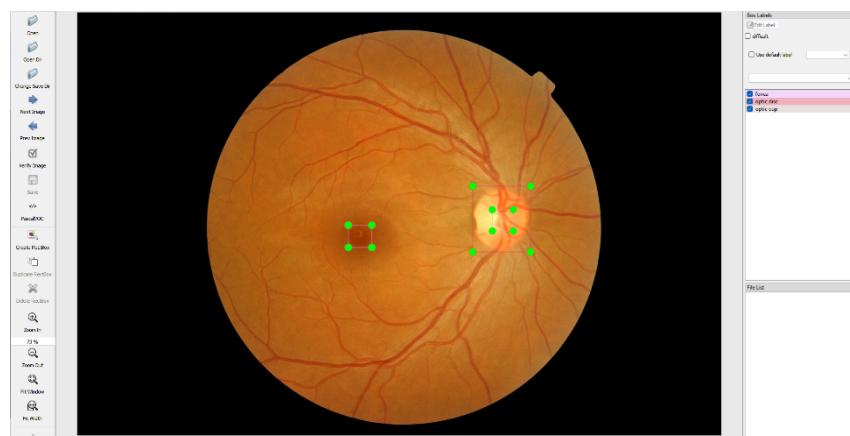


Figure 1: Image of a normal right eye with labelled anatomic features

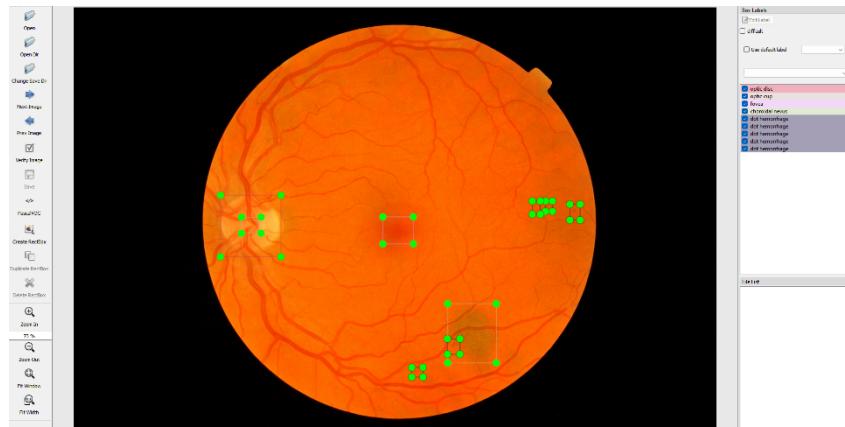


Figure 2: Image of a pathological left eye with labelled anatomic features and pathological features

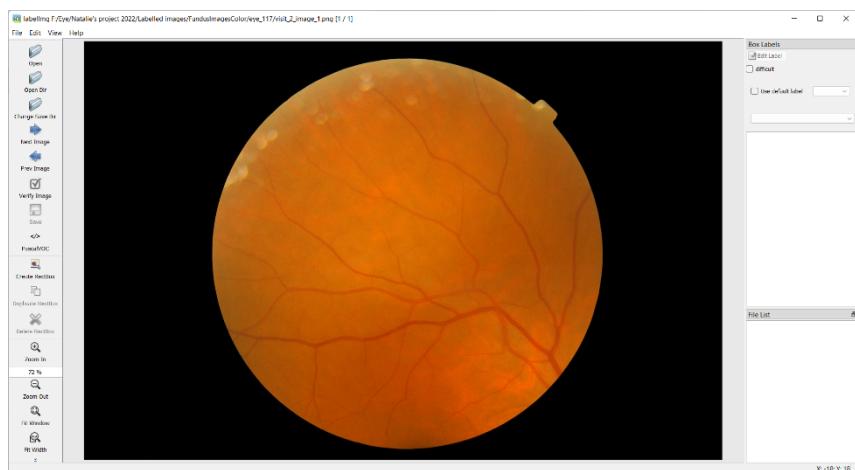


Figure 3: Image without features to label

## Exploration Data Analysis

### Descriptive Statistics

From the Rotterdam Ophthalmic Data Repository dataset, a total of 482 images had satisfactory images and views which were suitable for labelling. We generated 1,616 clinical annotations, of which, there were 16 unique class labels. These were used to train the model.

From the descriptive statistics, we observed a big difference in the sample size of the over-represented data set and the under-represented data set. The difference in magnitude is between 10-fold to 100-fold. The over-represented labels include: "fovea", "optic disc", "optic cup", "drusen" and "dot hemorrhage".

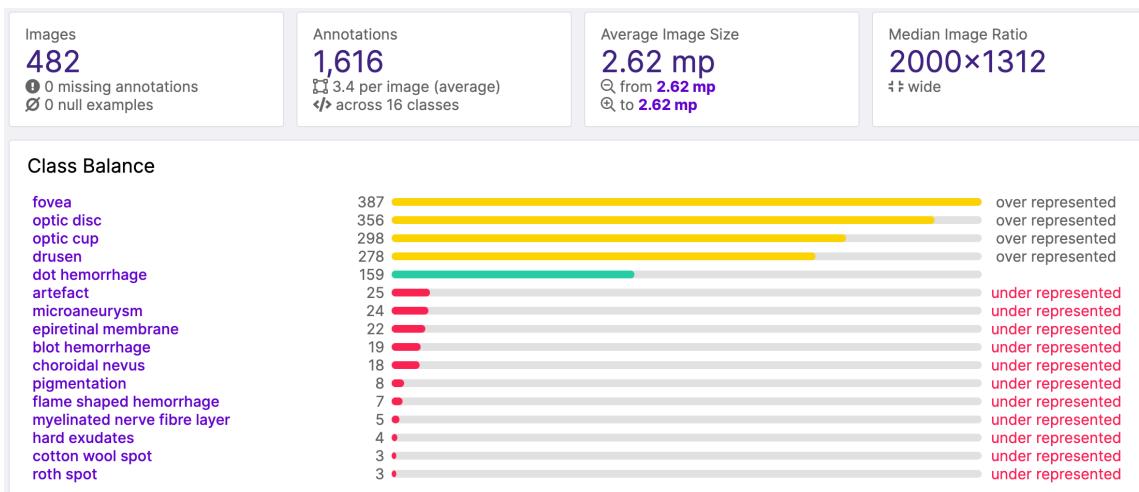


Figure 3: Descriptive Statistic of data set

### Image Pre-processing

In this section, we will describe and demonstrate various image pre-processing technique and steps we have explored and adopted for the training of our models.

#### Data Trimming

The original image that we have obtained has a dimension of 2000 x 1312 pixels. Using the data at its full resolutions will require huge amount of computing resources to train and make inference. With google co-lab we are constrain with limited amount of RAM (12gb) and CPU/GPU resources. To overcome the limited computing resource, we first trim the exceed whitespace around the image to achieve a dimension of 1000 x 1000 pixels. Following that, we further resize the fundus image to 512 x 512 pixels.

For training of the classification model, we crop the Region of Interest (ROI) based on their respective bounding box label. Once the images are cropped, they are further resized into 50 x 50 pixels for model training.

#### Black area re-fill

The fundus images obtained from the fundus camera had whitespaces at the four corners, which were filled in the black. To avoid model bias towards this black background, we replace the black colour with RGB value of [252, 108, 38].

#### Normalization

In order to reduce computational load during training and inference, we performed a min-max normalization of the values from 0 and 255 to between 0 and 1. This transformation did not change the image.

#### Histogram Equalization

To enhance key features of the fundus image, we perform histogram equalization. This transformation spread out the most frequent pixel intensity value. The result is that it reduces the representation of the lighter areas and enhance the darker areas.

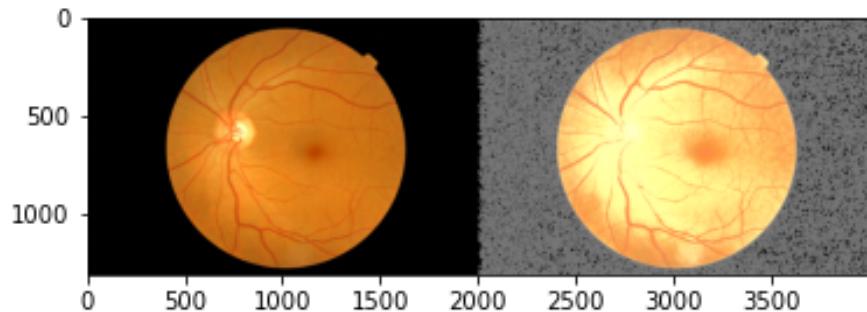


Figure 4: Example of Histogram Normalization

#### CLANE Histogram Equalization

Using a simple histogram equalization may over-amplified the contrast. To address such an issue, we have also explored the use of Contrast Limited Adaptive Histogram Equalization (CLANE). Instead of performing equalization based on the entire image, CLANE focus on a small grid and equalize based on the value of neighbouring pixel, resulting in a much more balanced adjustment of the contrast. Using the OpenCV library, we set the parameter `cliplimit=10` and `tileGridSize=(3,3)` for this adjustment.

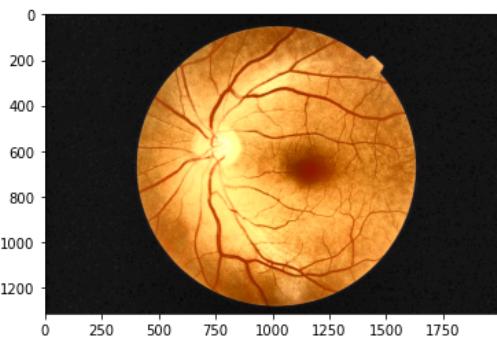


Figure 5: Example of CLANE adjusted image

#### Summary of image pre-processing steps

To evaluate the usefulness of each of the pre-processing techniques we have explored above, we tried each of the pre-processing techniques with a small subset of the data with various machine learning models and evaluate the performance. We found that black area re-fill and normalization yield marginal improvement in Support Vector Machine (SVM) classifier and Random Forest Classifier with overall accuracy improvement from 79% to 86% and 74% to 84% respectively. Additionally, we found that histogram equalization does not yield much improvement on the models.

#### Data Augmentation methods

To improve the generalizability of the model and address issue of under-represented data, we explored several data augmentation techniques.

##### Flip

We augmented the data by performing horizontal and vertical flip of the images. When the images are flipped, the respective coordinates of the bounding boxes will flip as well.

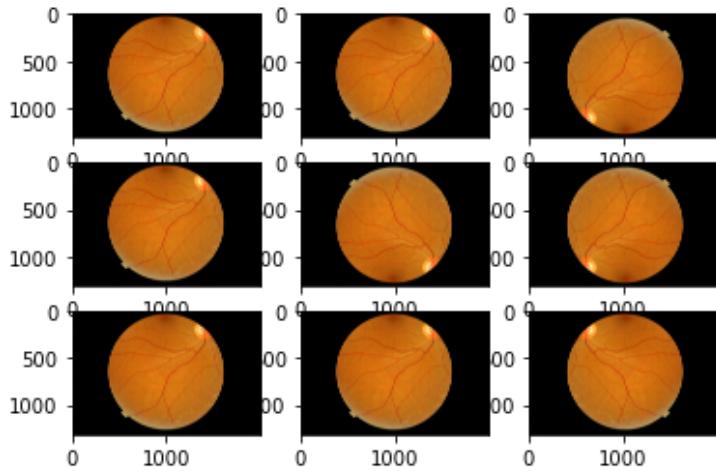


Figure 6: Example of Image Flip

#### Rotation

The fundus image could be rotated by different angles. In the example below, we rotated the image every 30 degrees. When we perform the rotation, we observed that the region of interest may be shifted out of the frame. Nevertheless, we consider this to be a feature of image augmentation that will help to improve the generalizability of the model.

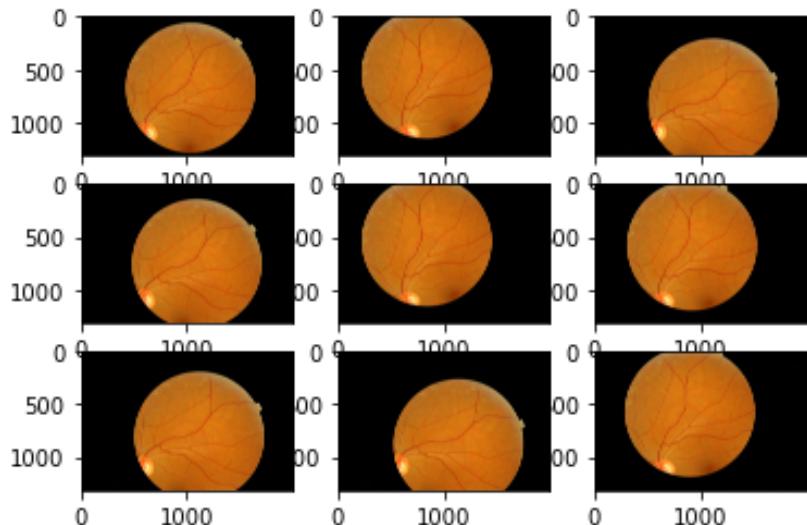


Figure 7: Example of Image Rotation

### Brightness

Different brightness allows the model to generalize across images taken using different brightness settings of different fundus cameras. In patients with cataracts, the images may also appear dimmer.

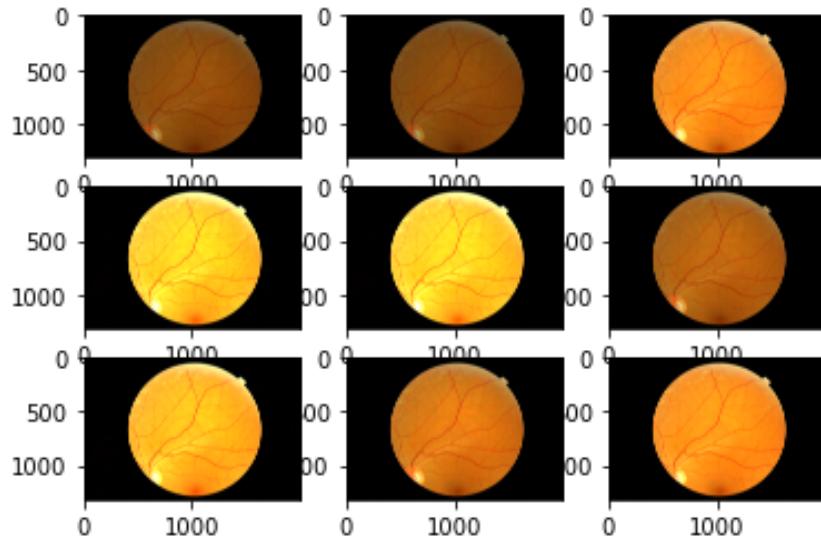


Figure 8: Example of brightness adjustment of Image

### Shift

Apart from rotation, we can also perform pixel shift to augment the data representation to improve the generalizability of the model. In the following example, we performed horizontal and vertical pixel shift without changing the dimension of the image. The shift can potentially improve the generalizability of the model adapting to images taken with at different image dimension by different camera.

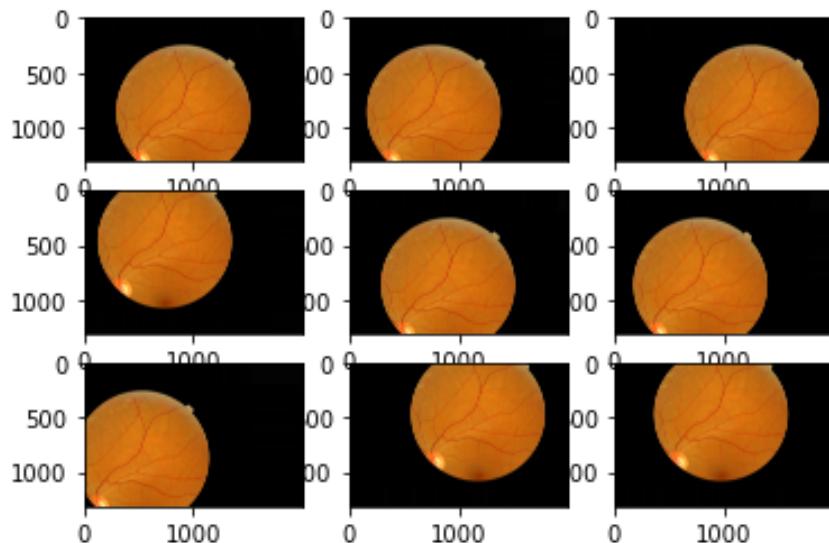


Figure 9: Example of Pixel Shift augmentation

## Intelligent system for eye diagnosis

### Zoom

We have also explored the zoom technique at zoom the images at different zoom level. However, upon discussion with domain expert, we reckon that this augmentation may not be very useful for our use case, as we expect all fundus image to have a similar and fixed focal length.

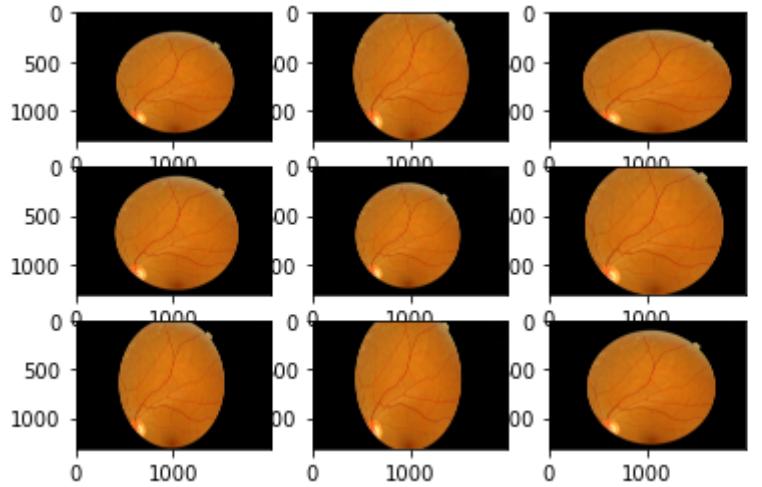


Figure 10: Example of Zoom augmentation

### Process Image data into numbers

In order to apply machine learning and deep learning methods for image recognition, we need to transform image data into numeric information. Following pre-processing methods are applied to transform image data into numeric vector.

#### Transform image into numeric vector

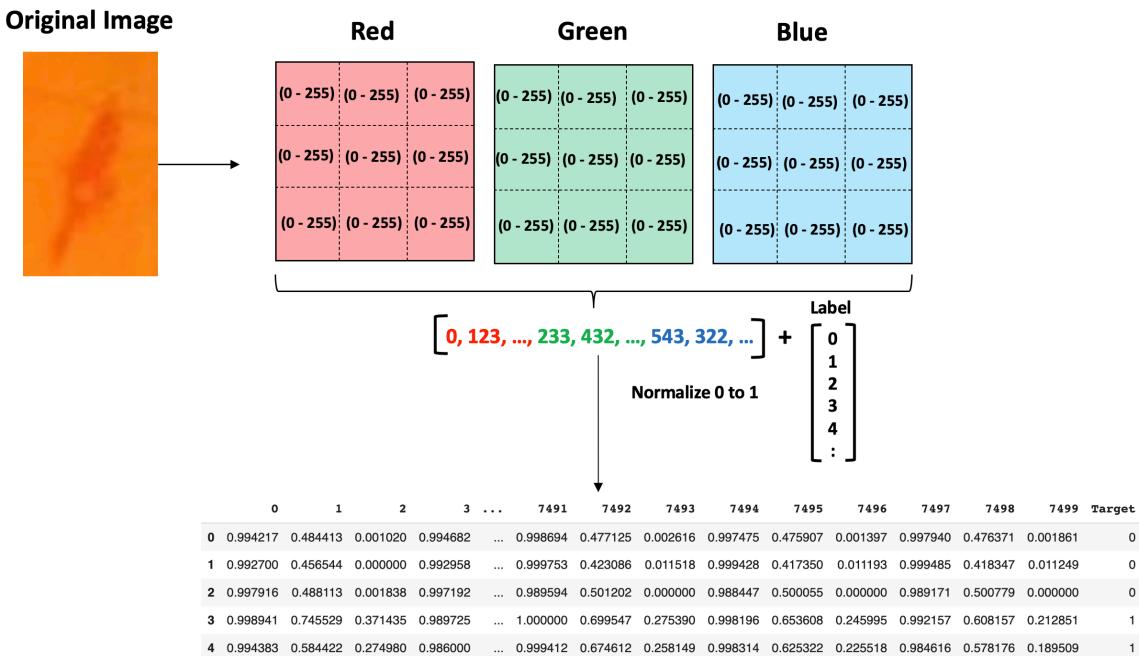


Figure 11: Transform image into a vector

## Intelligent system for eye diagnosis

In an image, each pixel contains a red, green, blue (RGB) value of 0 – 255 representing the intensity of each colour channel in a 2-dimensional matrix. The matrix is linearized to a 1-dimension vector and normalize to value between 0 and 1. The nominal label for each category is represented by the integer value of 0 – n, where n represent the number of classes. The target label value is then appended to the 1-dimension image vector.

### Processing Bounding Box and Nominal Label data

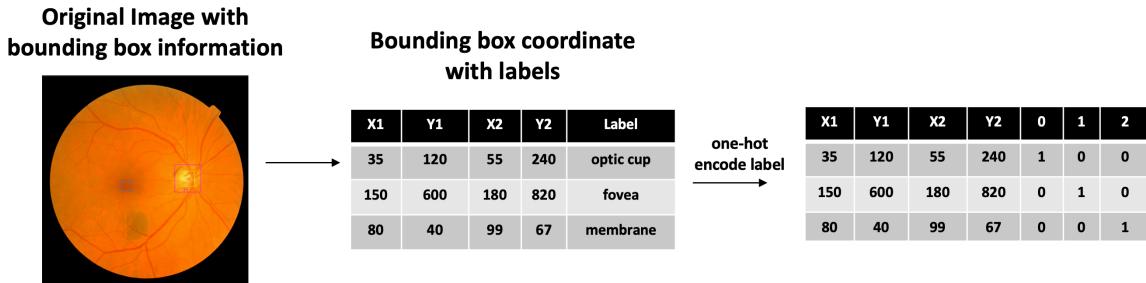


Figure 12: Transform normal label to one-hot encoding

With manual labelling of the data, each image consists of  $n$  number of bounding boxes. Each bounding box is represented by each row of data its respective x1, y1, x2, y2 coordinate and nominal label. For the purpose of machine learning processing, the nominal labels are converted to binary value of 0 and 1 using one-hot encoding method.

### Data pipeline

- Process raw data
- Image cropping and resize
- Transform image into matrix for training and inference

## Model Design

### Bounding Box Detection Approach

#### Sliding Box method

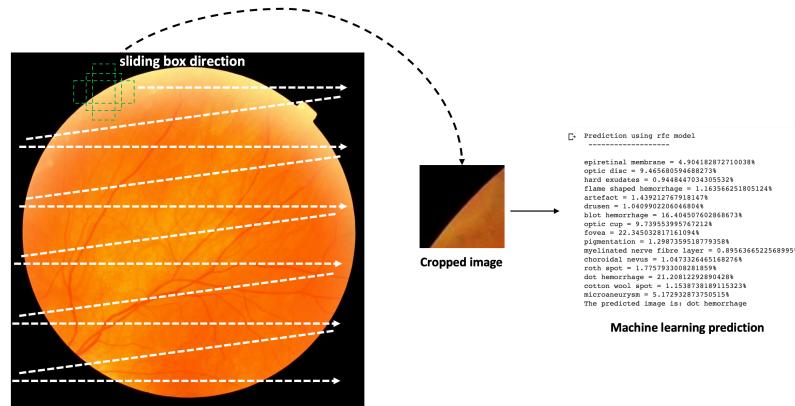


Figure 13: Sliding box detection approach

In a naïve approach, we adopted the sliding box detection method. We predefined a set of bounding boxes and slide the boxes across the image in a sequential matter. At each sequence, a crop section of the image is obtained and use it for prediction of the classification.

#### Anchor box method

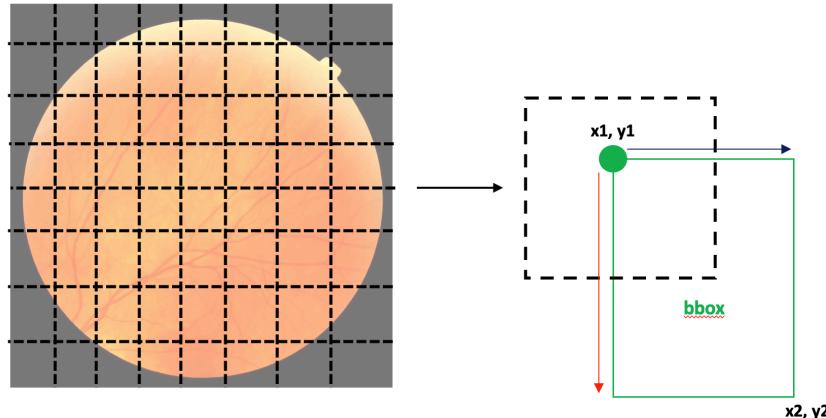


Figure 14: Anchor Box approach.

To capitalize the capability of neural network, the anchor box approach is adopted to improve the speed of bounding box detection. In anchor box approach, the image is represented in an  $8 \times 8$  grid format. In each grid, the  $x1, y1$  of the bounding box will be used as the anchor point of the bounding box and the  $x2, y2$  value will be considered together with its respective anchor point value.

### Machine Learning Training with Entire Data Set

#### Support Vector Machine

```

  ✓  print(f'Best SVM parameters: {model.best_params_}')
0s
  □  Best SVM parameters: {'C': 0.5, 'gamma': 'scale', 'kernel': 'linear'}

```

Figure 15: Best parameter for SVM classifier

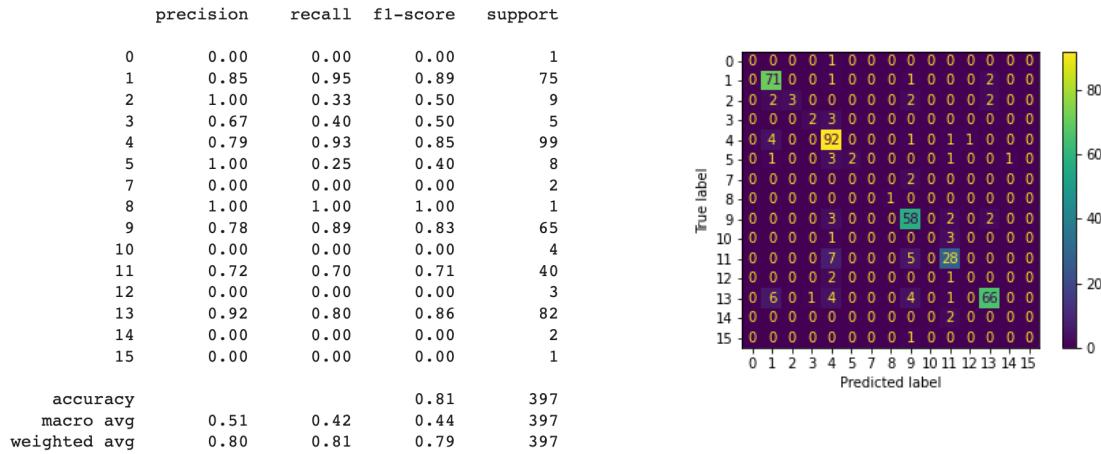
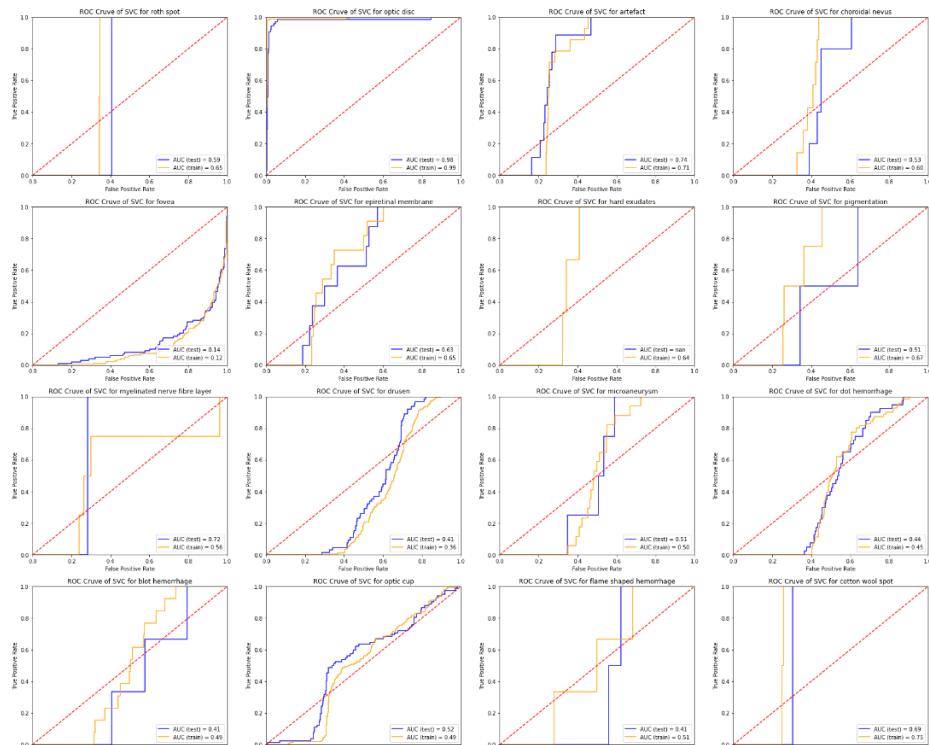


Figure 16: SVM classifier classification report and confusion matrix



**Figure 17: SVM classifier Receiver operating characteristic**

We attempt to train a model using Support Vector Machine classifier with the entire sets of data. The optimization of hypermeter is achieved using GridSearchCV from the scikit-learn library. The accuracy score of the classifier is reported to be 0.81. Looking at the confusion matrix, the classifier was able to accurately predict “optic disc”, “fovea”, “myelinated nerve fibre layer”, “dot haemorrhage” and “optic cup”. Cross referencing the result with the Receiver operating characteristic (ROC), our accuracy observation may be skewed by overly well perform by “optic disc” which has an Area under curve (AUC) of 0.99 for both training and test set. For other labels such as “fovea” has suffered a high false positive rate and other labels performance are affected by low sample representation.

## Intelligent system for eye diagnosis

### Random Forest Classifier

```
print(f'Best RFC parameters: {model.best_params_}')

Best RFC parameters: {'criterion': 'entropy', 'max_depth': 30, 'max_features': 'sqrt'}
```

Figure 18: Best parameter for Random Forest classifier

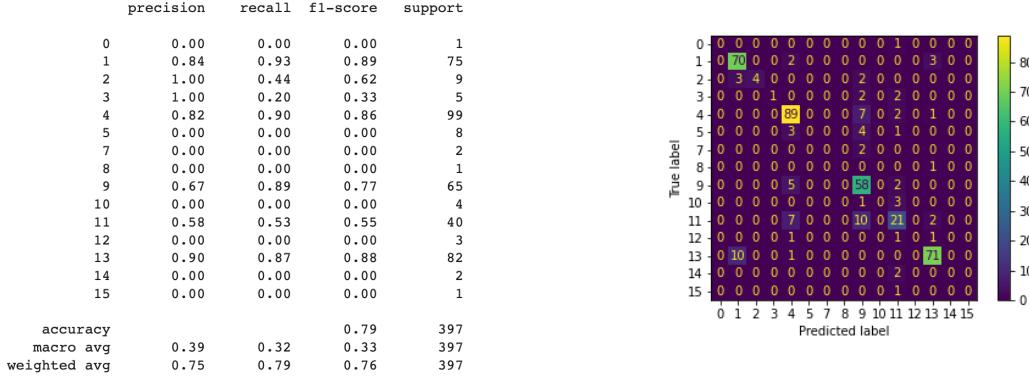


Figure 19: Random Forest classifier classification report and confusion matrix

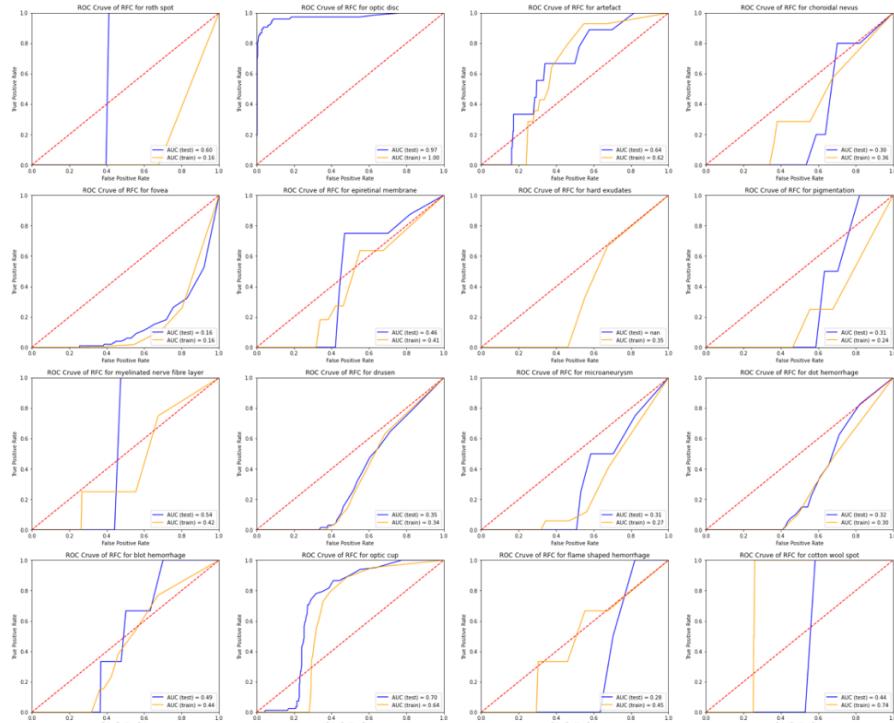


Figure 20: Random Forest classifier Receiver operating characteristic

Next, we attempt to train a model using RandomForestClassifier with the same sets of data. The optimization of hypermeter is achieved using GridSearchCV from the scikit-learn library. The accuracy score of the classifier is reported to be 0.79. Looking at the confusion matrix, the classifier was able to accurately predict “optic disc”, “fovea”, “myelinated nerve fibre layer”, “dot haemorrhage” and “optic cup”. Cross referencing the result with the Receiver operating characteristic (ROC), our accuracy observation may be skewed by overly well perform by “optic disc” which has an Area under curve (AUC) of 0.97 for test data set. For other labels such as “fovea” has suffered a high false positive rate and other labels performance are affected by low sample representation, like SVM classifier model.

## Intelligent system for eye diagnosis

### k-nearest Neighbour

```
▶ print(f'Best KNN parameters: {model.best_params_}')

▷ Best KNN parameters: {'algorithm': 'auto', 'n_jobs': -1, 'n_neighbors': 3, 'weights': 'distance'}
```

Figure 21: Best parameter for KNN classifier

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1
1	0.91	0.80	0.85	87
2	1.00	0.14	0.25	7
3	0.00	0.00	0.00	4
4	0.79	0.79	0.79	95
5	0.00	0.00	0.00	6
6	0.00	0.00	0.00	1
7	0.00	0.00	0.00	2
8	0.00	0.00	0.00	1
9	0.61	0.78	0.69	73
10	0.43	0.50	0.46	6
11	0.38	0.61	0.47	33
12	0.00	0.00	0.00	5
13	0.92	0.82	0.87	73
14	0.00	0.00	0.00	2
15	0.00	0.00	0.00	1
accuracy			0.72	397
macro avg	0.32	0.28	0.27	397
weighted avg	0.73	0.72	0.71	397

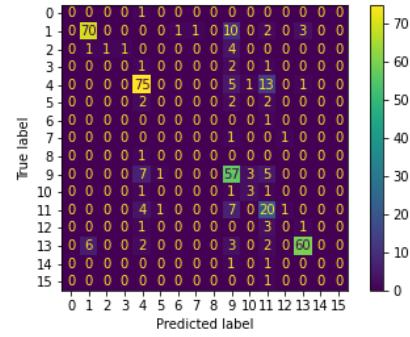


Figure 22: KNN classifier classification report and confusion matrix

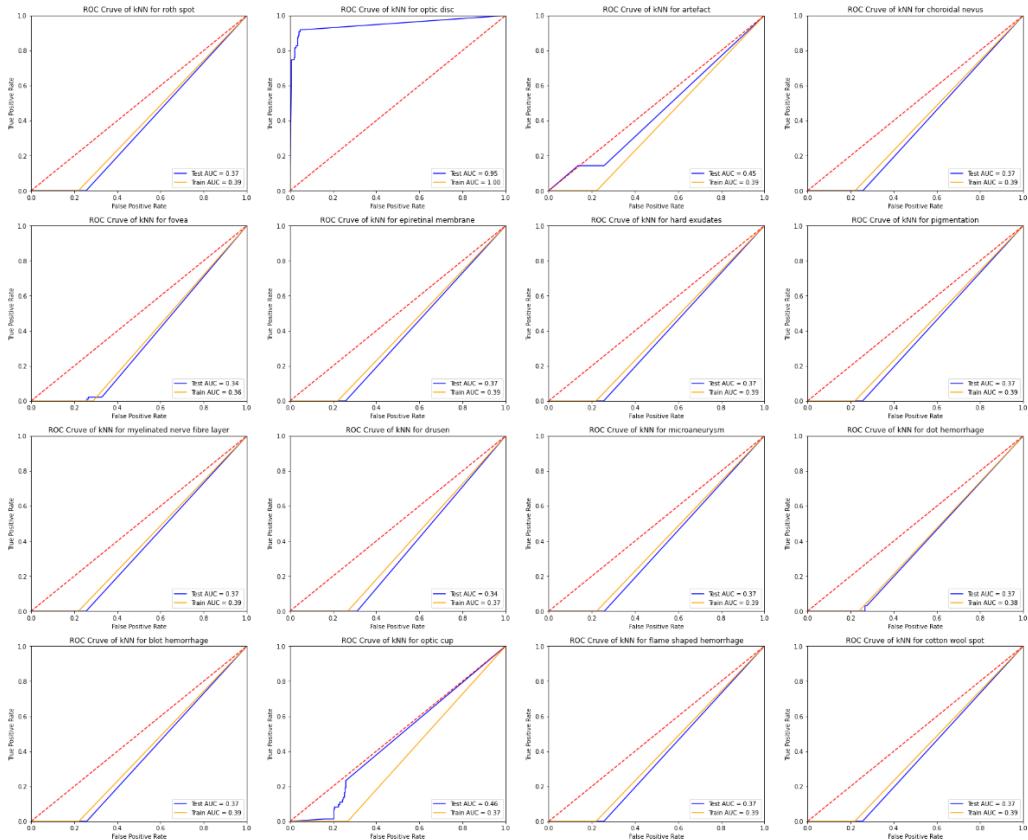


Figure 23: KNN classifier Receiver operating characteristic curve

We attempt to train a model using K-nearest classifier with the same sets of data. The optimization of hypermeter is achieved using GridSearchCV from the scikit-learn library. The accuracy score of the classifier is reported to be 0.72. Looking at the confusion matrix, the

## Intelligent system for eye diagnosis

classifier was able to accurately predict “optic disc”, “fovea”, “myelinated nerve fibre layer”, “dot haemorrhage” and “optic cup”. Cross referencing the result with the Receiver operating characteristic (ROC), our accuracy observation may be skewed by overly well perform by “optic disc” which has an Area under curve (AUC) of 0.95 for test data set. For other labels, the performance of predication is close to being random.

### Linear Discriminant Analysis

```
▶ print(f'Best LDA parameters: {model.best_params_}')
```

↳ Best LDA parameters: {'solver': 'svd'}

Figure 24: Best parameter for LDA classifier

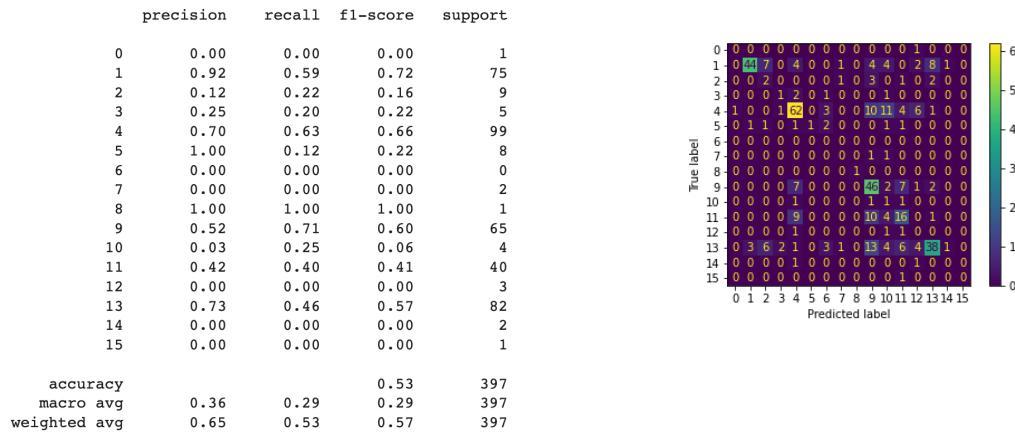


Figure 25: LDA classifier classification report and confusion matrix

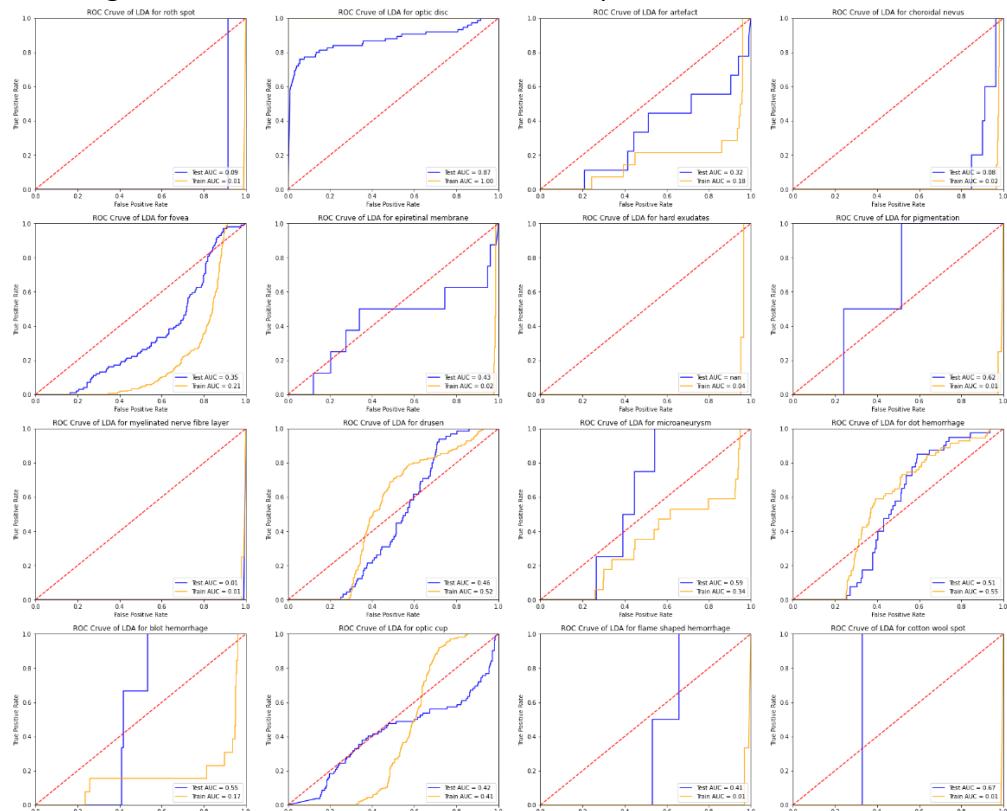


Figure 26: LDA classifier Receiver operating characteristic curve

Finally, we attempt to train a model using Linear Discriminate Analysis with the same sets of data. The optimization of hypermeter is achieved using GridSearchCV from the scikit-learn library. The accuracy score of the classifier is reported to be 0.72. Looking at the confusion matrix, the classifier was able to accurately predict “optic disc”, “fovea”, “myelinated nerve fiber layer”, “dot haemorrhage” and “optic cup”. Cross referencing the result with the Receiver operating characteristic (ROC), our accuracy observation may be skewed by overly well perform by “optic disc” which has an Area under curve (AUC) of 0.95 for test data set. For other labels, the performance of predication is close to being random.

### Machine Learning Training with Overrepresented Data Set

Based on the above experience, we found certain labels trained better while others have poor performance. We suspect 2 potential causes: (i) Imbalanced data set may affect the ability to discriminate the labels effectively, and (ii) too many labels may increase the complexity of the model making it perform poorly for all labels. Hence, we decided the split the data set into two separate set: (i) over-represented labels and (ii) under-represented labels and train the model separately.

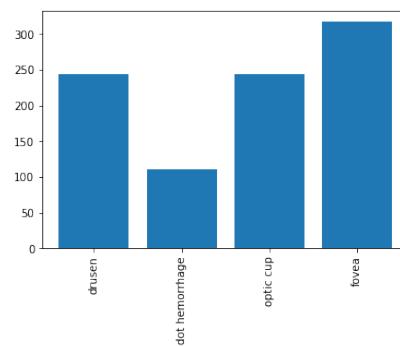


Figure 27: Overrepresented Labels

### Support Vector Machine

```
[16] print(f'Best SVM parameters: {model.best_params_}')

Best SVM parameters: {'C': 0.5, 'gamma': 'scale', 'kernel': 'linear'}
```

Figure 27: Best parameter for SVM classifier for over-represented labels

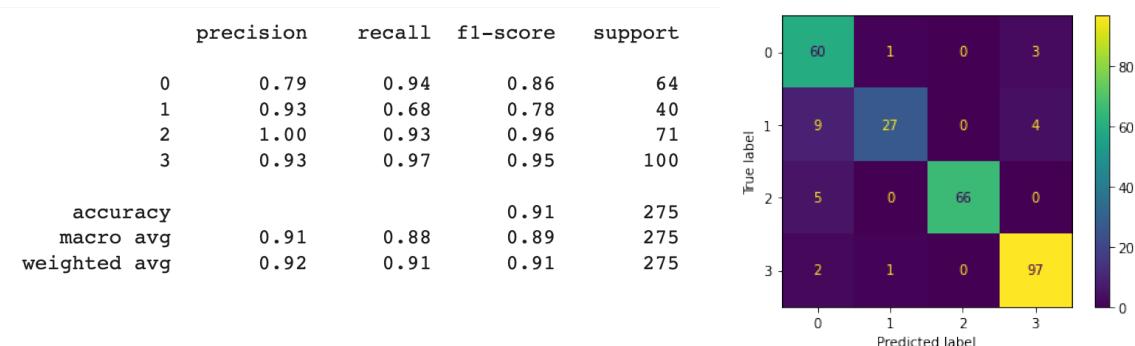


Figure 28: SVM classifier classification report and confusion matrix for over-represented labels

## Intelligent system for eye diagnosis

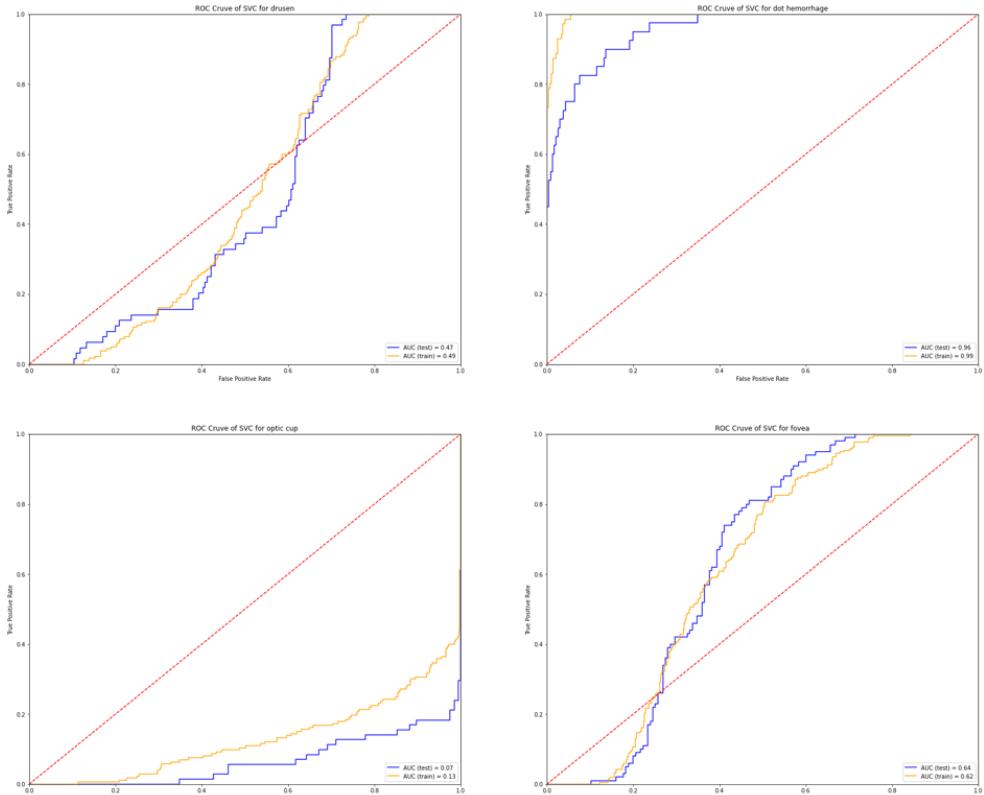


Figure 29: SVM classifier Receiver operating characteristic for over-represented labels

By training with a modified and more balanced data set, we noticed an improved performance in the discrimination for dot haemorrhage (AUC: 0.96, F1: 0.78) and fovea (AUC: 0.64, F1: 0.95). However, the performance for optic cup became worse (AUC: 0.07, F1: 0.96) and drusen had no obvious change.

## Intelligent system for eye diagnosis

### Random Forest Classifier

```
[23] print(f'Best RFC parameters: {model.best_params_}')

Best RFC parameters: {'criterion': 'entropy', 'max_depth': 30, 'max_features': 'sqrt'}
```

Figure 30: Best parameter for Random Forest classifier for over-represented labels

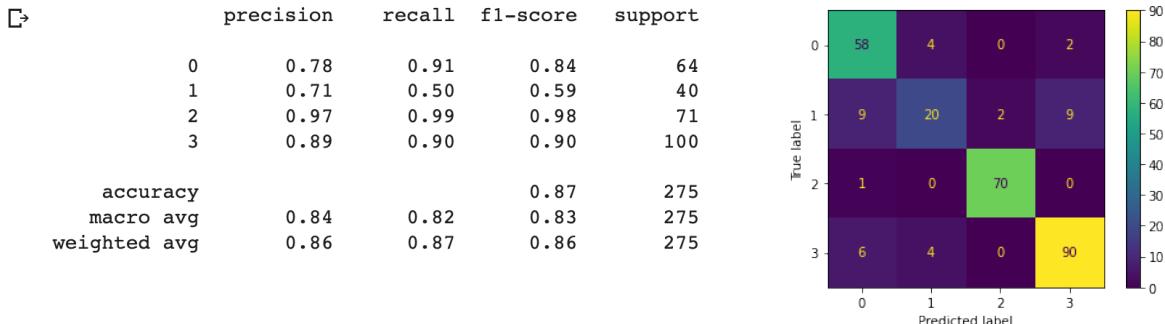


Figure 31: Random Forest classifier classification report and confusion matrix for over-represented labels

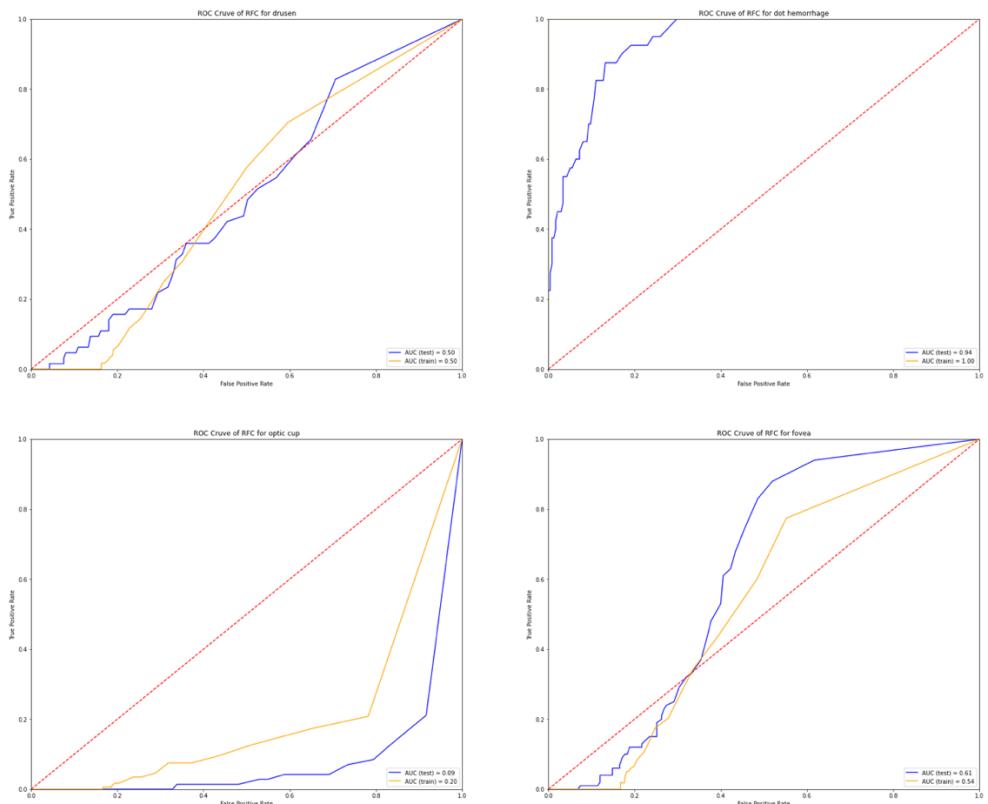


Figure 32: Random Forest classifier Receiver operating characteristic for over-represented labels

Next, we attempt to train the model with Random Forest Classifier. Our observation was similar to that of SVM classifier, where dot haemorrhage (AUC: 0.94, F1: 0.59) and fovea (AUC: 0.61, F1: 0.90) saw significant improvement over the initial model with 16 labels. As per our observation with the SVM classifier, by using a Random Forest classification,

discrimination for optic cup (AUC: 0.09, F1: 0.98) dropped and there was not much difference for drusen.

### k-Nearest Neighbour Classifier

```
print(f'Best KNN parameters: {model.best_params_}')

Best KNN parameters: {'algorithm': 'auto', 'n_jobs': -1, 'n_neighbors': 3, 'weights': 'distance'}
```

Figure 33: Best parameter for k-Nearest Neighbor classifier for over-represented labels

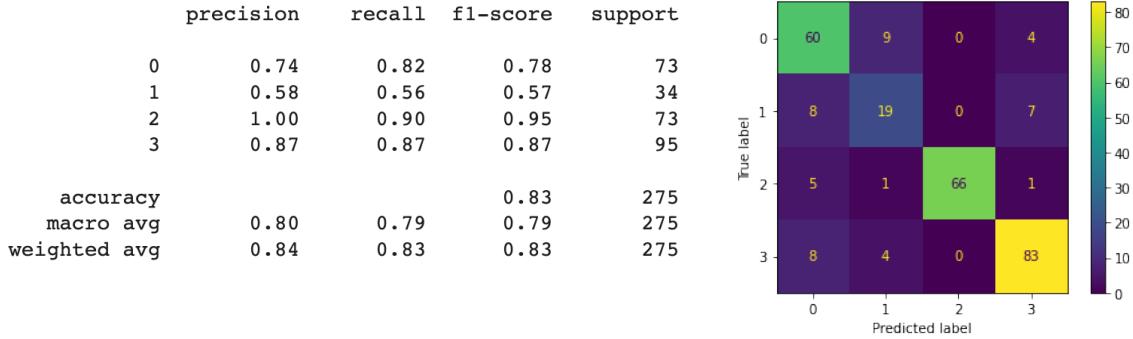


Figure 34: k-Nearest Neighbour classifier classification report and confusion matrix for over-represented labels

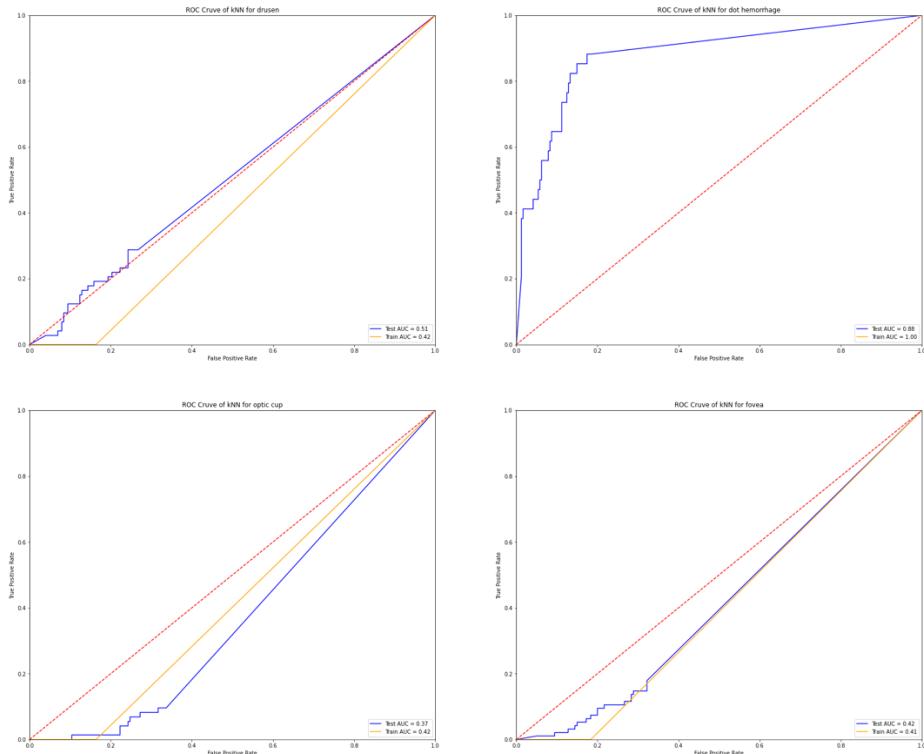


Figure 35: k-Nearest Neighbor classifier Receiver operating characteristic for over-represented labels

Using kNN classifier, we only observed improved performance for dot hemorrhage (AUC: 0.88, F1: 0.57) over the initial model trained using 16 labels. For the rest, the performance was not better than random.

### Linear Discriminant Analysis



```
print(f'Best LDA parameters: {model.best_params_}' )
```

Best LDA parameters: {'solver': 'svd'}

Figure 36: Best parameter for LDA classifier for over-represented labels

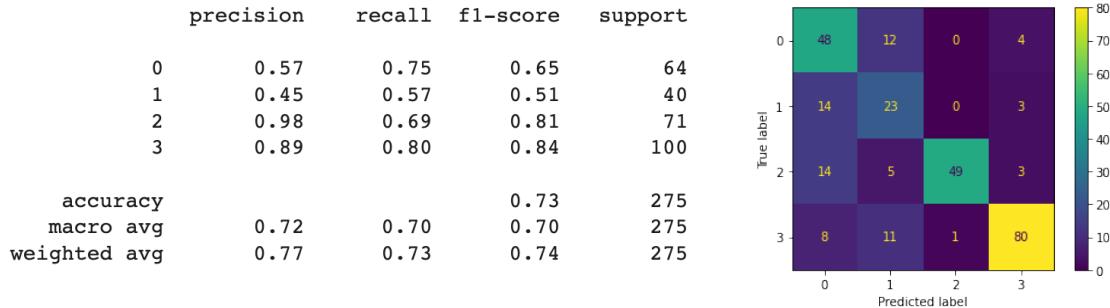


Figure 37: LDA classifier classification report and confusion matrix for over-represented labels

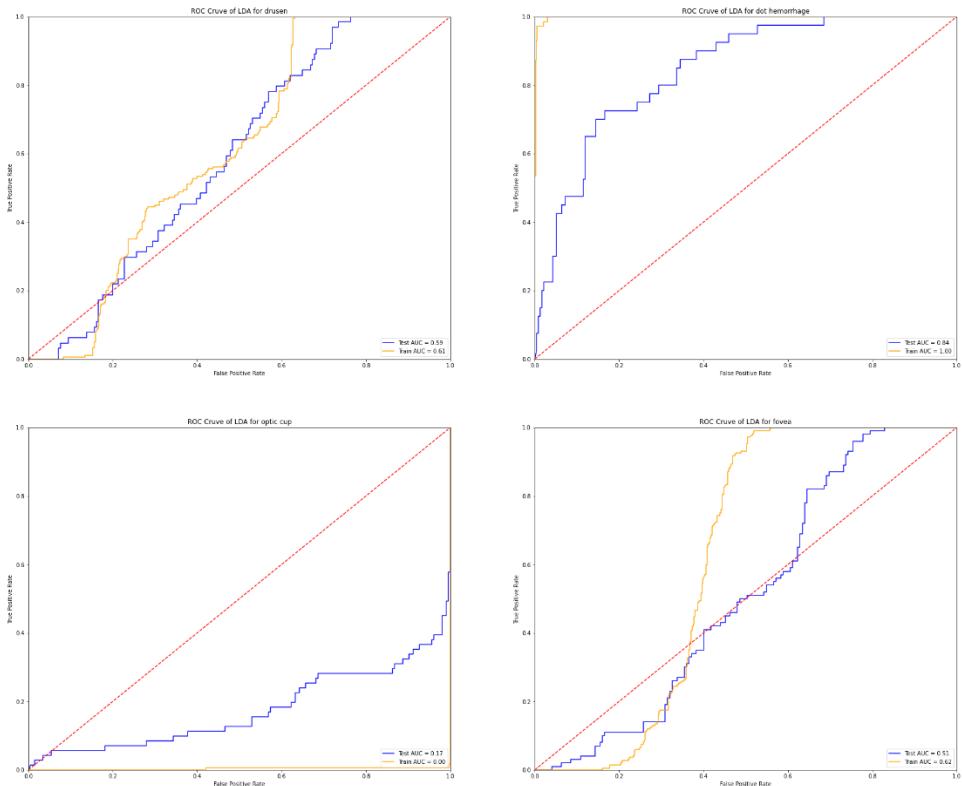


Figure 38: LDA classifier Receiver operating characteristic for over-represented labels

With the LDA classifier, we too observed slight improvement for discriminating dot haemorrhage (AUC: 0.84, F1: 0.51). However, the model performed poorly in accurately classifying drusen, fovea and optic cup.

### Machine Learning Training with Underrepresented Data Set

To address the issue of the model being bias towards the over-represented data and the accuracy being skewed by the over-represented labels, we decided to remove the over-represented labels so that the remaining labels can be trained on a much-balanced data set. Since the over-represented labels are in more than 10x the size of the under-represented labels, simple data augmentation cannot make up the difference. Hence, we decided on splitting the data set into 2 categories – over-represented and under-represented.

The techniques used for the over-represented data set has been discussed above. In this section, we will describe the techniques used in the under-represented data set.

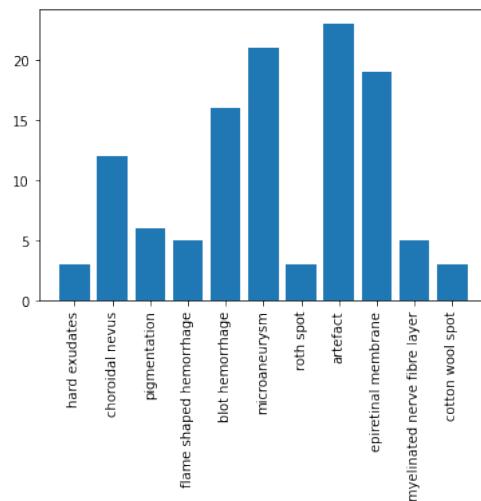


Figure 39: Underrepresented Labels

### Support Vector Machine

```
[13] print(f'Best SVM parameters: {model.best_params_}')

Best SVM parameters: {'C': 0.5, 'gamma': 'scale', 'kernel': 'poly'}
```

Figure 40: Best parameter for SVM classifier for under-represented labels

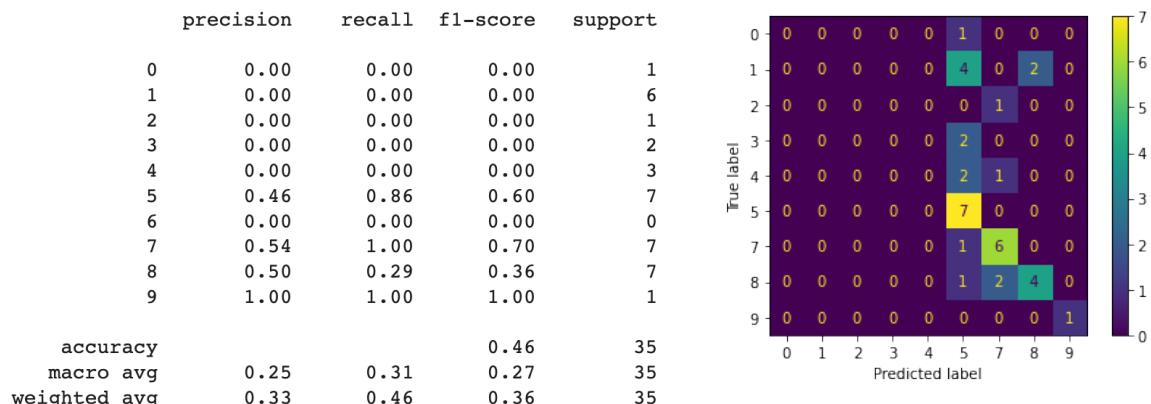


Figure 41: SVM classifier classification report and confusion matrix for under-represented labels

## Intelligent system for eye diagnosis

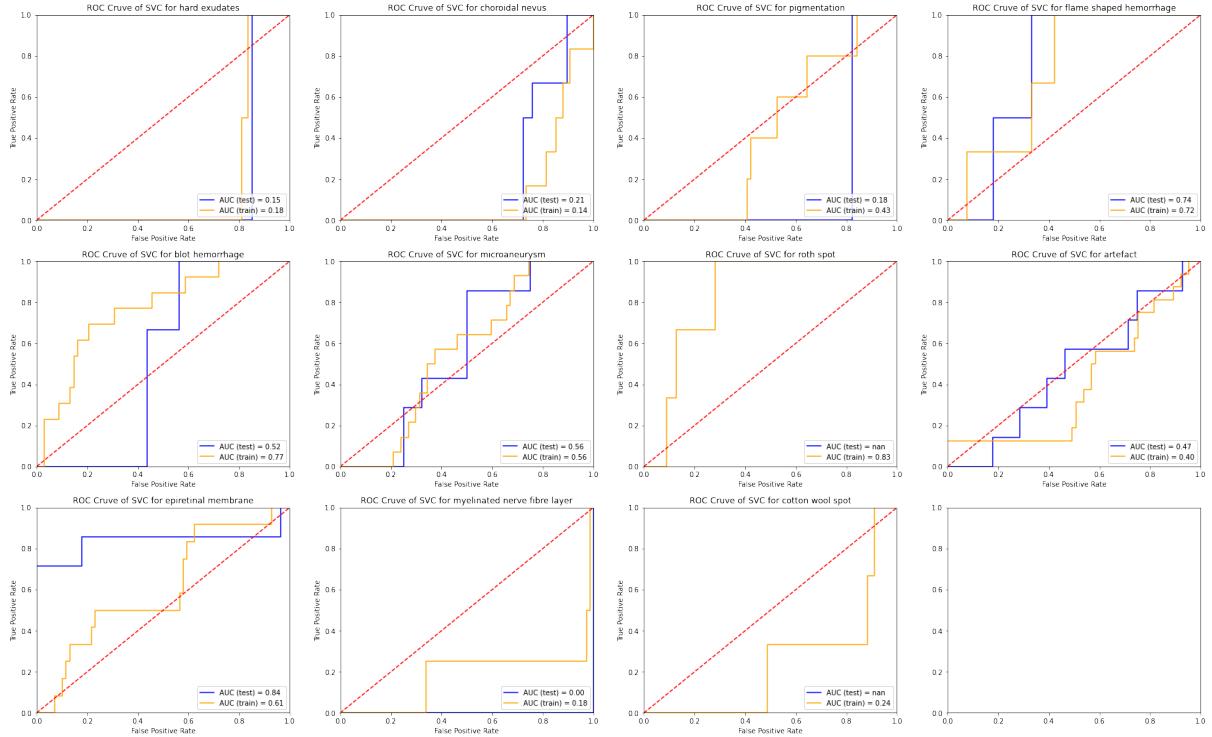


Figure 42: SVM classifier Receiver operating characteristic for under-represented labels

First, using SVM classifier, we notice that the removal of overrepresented labels does not improve the discriminating capacity and performance of the model. This suggest that SVM may not be a suitable model for this task, or that we need to explore further in feature extraction to improve its performance.

## Intelligent system for eye diagnosis

### Random Forest Classifier

```
[20] print(f'Best RFC parameters: {model.best_params_}')

Best RFC parameters: {'criterion': 'entropy', 'max_depth': 30, 'max_features': 'log2'}
```

Figure 43: Best parameter for Random Forest classifier for under-represented labels

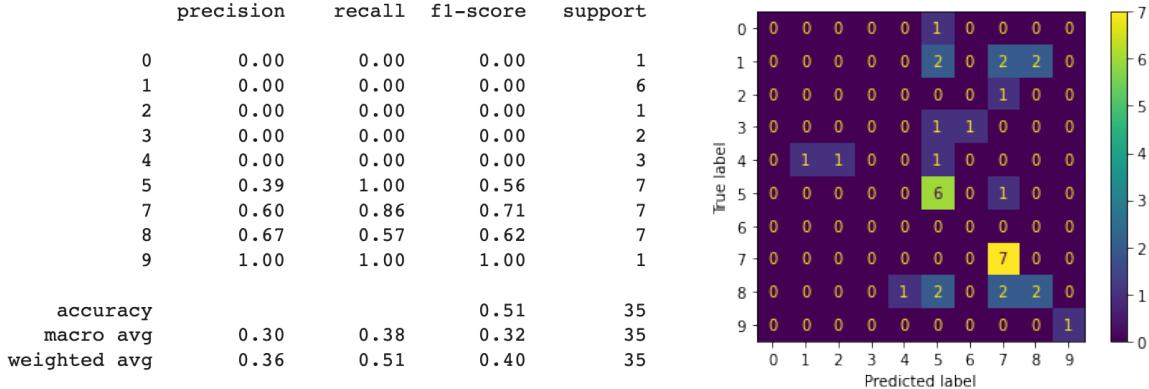


Figure 44: Random Forest classifier classification report and confusion matrix for under-represented labels

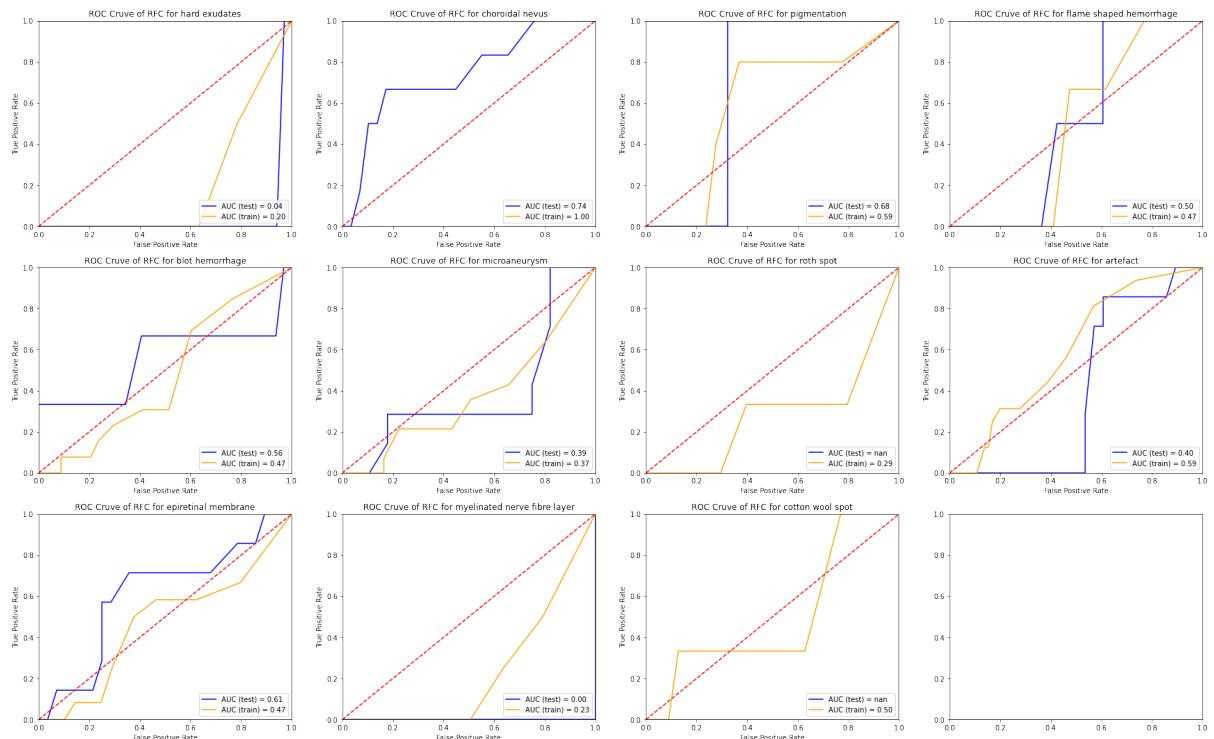


Figure 45: Random Forest classifier Receiver operating characteristic for under-represented labels

Next, similar to SVM classifier, Random Forest Classifier also performed poorly in discriminating these labels. Some labels experience the issue of not having sufficient representation for the training and testing to generate the performance score.

## Intelligent system for eye diagnosis

### k-Nearest Neighbour Classifier

```
print(f'Best KNN parameters: {model.best_params_}')

Best KNN parameters: {'algorithm': 'auto', 'n_jobs': -1, 'n_neighbors': 3, 'weights': 'distance'}
```

Figure 46: Best parameter for k-Nearest Neighbor classifier for under-represented labels

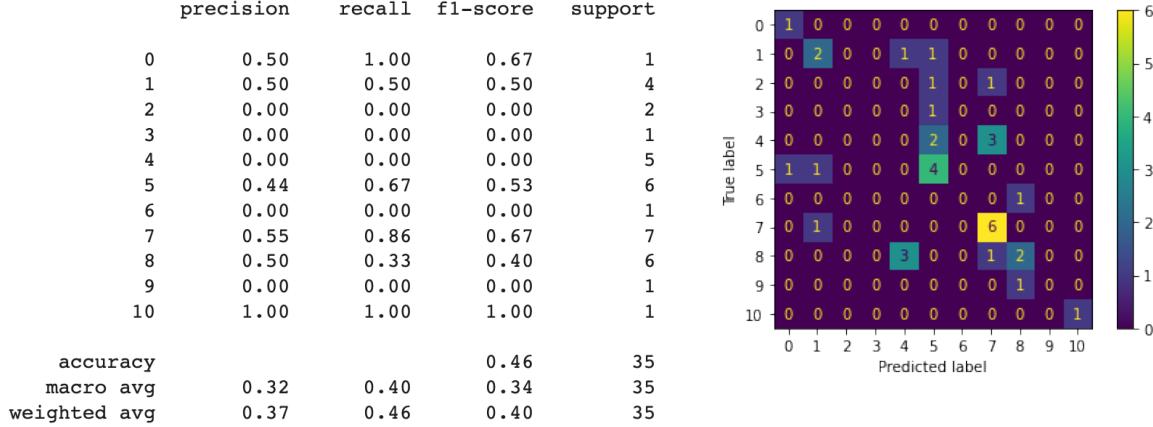


Figure 47: k-Nearest Neighbor classifier classification report and confusion matrix for under-represented labels

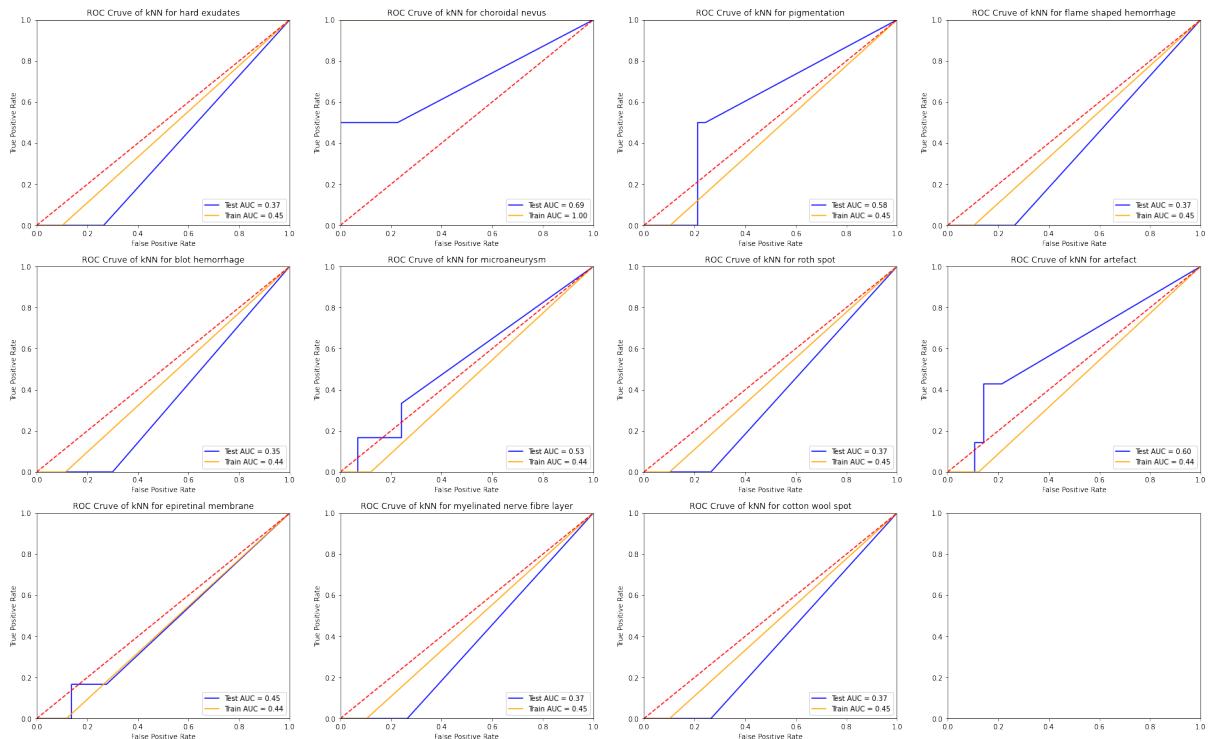


Figure 48: k-Nearest Neighbor classifier Receiver operating characteristic for under-represented labels

Using kNN classifier, the algorithm was not able to identify an optimal model. The final model performance was no better than random guess.

## Intelligent system for eye diagnosis

### Linear Discriminate Classifier

```
print(f'Best LDA parameters: {model.best_params_}' )
```

```
Best LDA parameters: {'solver': 'svd'}
```

Figure 49: Best parameter for LDA classifier for under-represented labels

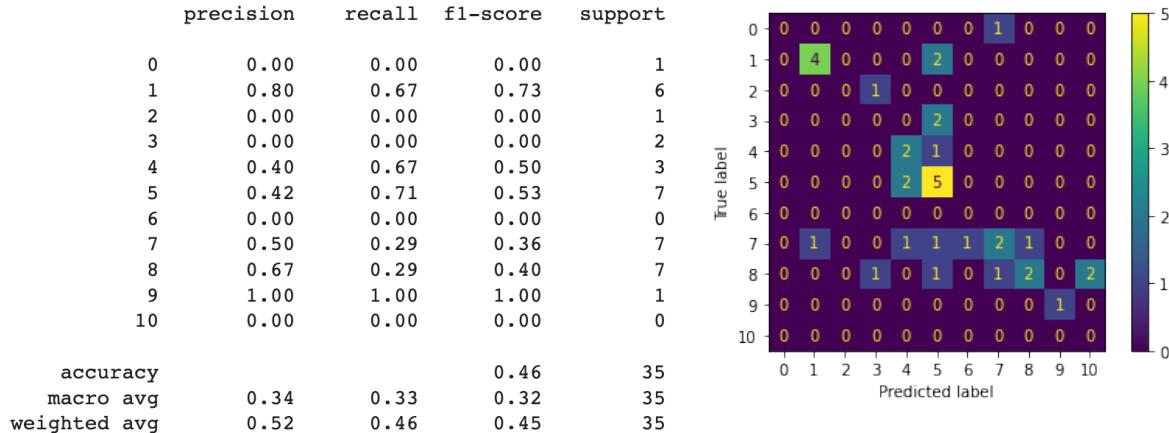


Figure 50: LDA classifier classification report and confusion matrix for under-represented labels

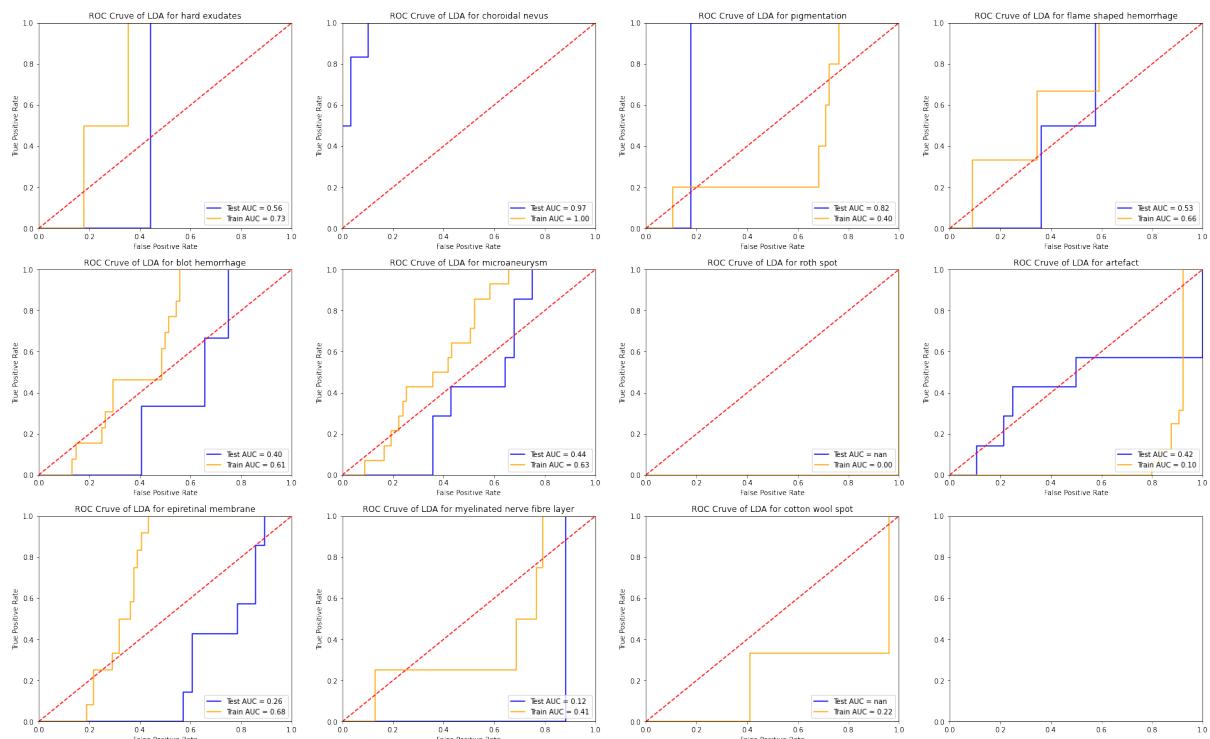


Figure 51: LDA classifier Receiver operating characteristic for under-represented labels

In LDA model, similar to the above 3 supervised learning methods, it is not able to effectively discriminate the labels well.

### Summary on supervised machine learning approach

We started our project by exploring a simple approach using machine learning methods to discriminate fundus image labels. The rational was that a simple method could potentially give us the ability to explain the decisions made by the classification system. However, despite trying out various models (SVM, RandomForest, kNN and LDA), none of them were able provide a satisfactory model for the classification task.

From this experience, we learnt that image classification can be a complex task, especially so when dealing with a large number of labels. The desire to adopt a simple machine learning approach must be coupled with a tedious feature extraction step for each class to enable better discrimination of those labels.

An interesting hypothesis for the reason for failure could be that the features in our images are complex and cannot be linear separated. This is especially so for labels in the “under-represented” category. We observed that certain labels are easier to separate, such as “optic disc” and “dot haemorrhage”. However, on deeper inspection, we found that there were significant number of “optic disc” samples which had a blank corner (Figure 52). We hypothesize that the model was discriminating the sample based on the presence of the “blank space”, instead of the actual “optic disc” feature we intended. Hence, we replaced “optic disc” sample with “optic cup”, as they share very similar characteristics.

Using our approach in supervised machine learning has another limitation. We expected, in our model, to be able to use the sliding box approach for bounding box detection. Yet, when we tried this approach, we found that a scan of a single fundus image with label detection for every window required up to 10 to 15 minutes per image. This is too slow for any real-world application and will not be accepted by the users.

In summary, we have learnt that the supervised machine approach does not work well for our use case and may require much effort in feature extraction for 16 different labels. Hence, we move on to explore the neural network approach to leverage on machine learning capability for feature extraction.

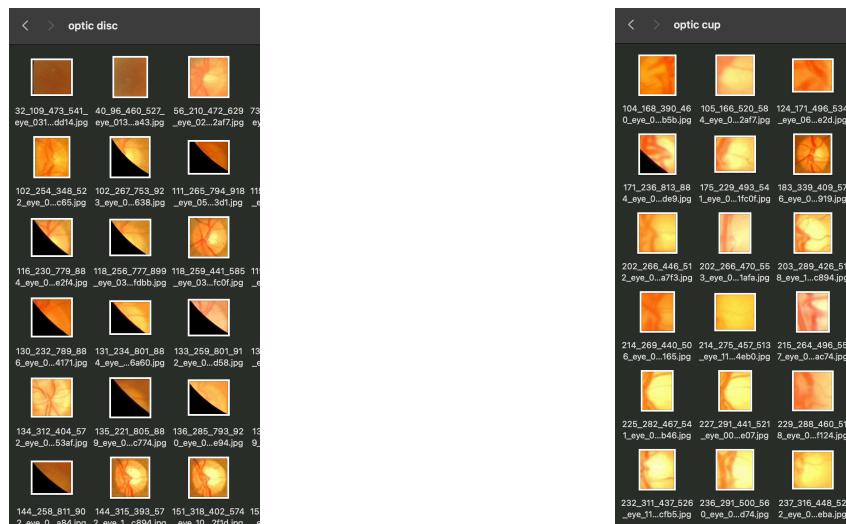


Figure 52: Sample of “optic disc” (left) and “optic cup” (right)

### Neural Network Approach

Following the machine learning approach, we explored the use of neural network to support the feature extraction process and bounding box detection. The use of neural network can also potentially speed up the bounding box detection process compared to the sliding box approach.

Based on our experience with supervised machine learning, we found difficulty in training the model with insufficient samples. Hence, we decided to drop the under-represented data set and focus on training the models with only a few key labels: “optic cup”, “fovea”, “dot haemorrhage” and “drusen”.

#### ResNet50 + Anchor box

First, we attempted to leverage the pretrained network from Keras library and combined it with our conceptual bounding box approach (described in the previous section). Taking reference from an online guide [9], we developed a high-level concept presented in Figure 52. The input image was resized to 512 by 512 pixels with 3 channels. Feature extraction was performed using ResNet50 architecture pretrained with ImageNet. Next, the parameters from the final layer were split into 3 types of predictions: a bounding box vector with 4 values ( $x_1, y_1, x_2, y_2$ ), a single probability value (0 to 1) and a classification prediction with a vector of 4 values (where 1 indicate the class label). Finally, these outputs concatenated into a single vector value.

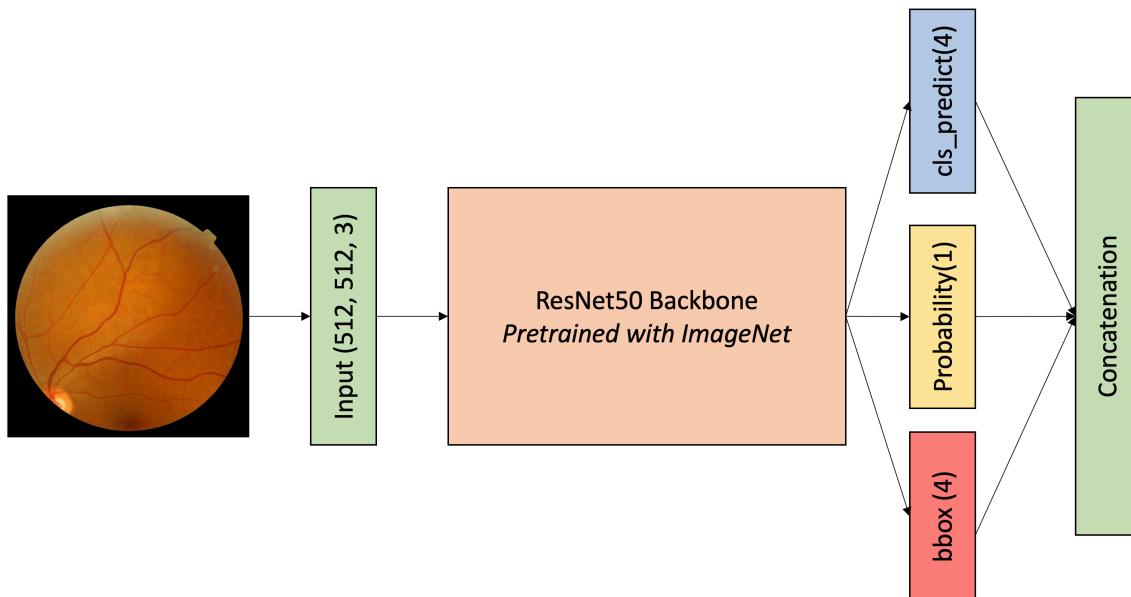


Figure 53: ResNet50 with Anchor Box approach.

## Intelligent system for eye diagnosis

Model: "model"	Layer (type)	Output Shape	Param #	Connected to
<hr/>				
image_input (InputLayer)	[None, 512, 512, 3]	0	2048	[]
resnet50 (Functional)	(None, None, None, 2048)	23587712		['image_input[0][0]']
conv2d (Conv2D)	(None, 16, 16, 32)	589856		['resnet50[0][0]']
max_pooling2d (MaxPooling2D)	(None, 8, 8, 32)	0		['conv2d[0][0]']
x_prob (Conv2D)	(None, 8, 8, 1)	289		['max_pooling2d[0][0]']
tf.math.greater (TFOpLambda)	(None, 8, 8, 1)	0		['x_prob[0][0]']
tf.ones_like (TFOpLambda)	(None, 8, 8, 1)	0		['x_prob[0][0]']
tf.zeros_like (TFOpLambda)	(None, 8, 8, 1)	0		['x_prob[0][0]']
x_boxes (Conv2D)	(None, 8, 8, 4)	1156		['max_pooling2d[0][0]']
tf.where (TFOpLambda)	(None, 8, 8, 1)	0		['tf.math.greater[0][0]', 'tf.ones_like[0][0]', 'tf.zeros_like[0][0]']
x_cls (Conv2D)	(None, 8, 8, 4)	1156		['max_pooling2d[0][0]']
tf.math.multiply (TFOpLambda)	(None, 8, 8, 4)	0		['x_boxes[0][0]', 'tf.where[0][0]']
tf.math.multiply_1 (TFOpLambda)	(None, 8, 8, 4)	0		['x_cls[0][0]', 'tf.where[0][0]']
concatenate (Concatenate)	(None, 8, 8, 9)	0		['x_prob[0][0]', 'tf.math.multiply[0][0]', 'tf.math.multiply_1[0][0]']
<hr/>				
Total params: 24,180,169				
Trainable params: 592,457				
Non-trainable params: 23,587,712				

Figure 54: Neural Network architecture summary

### Loss function definition

To facilitate training of neural network, we have defined the following loss functions.

#### Bounding Box (bbox) loss

The loss function for bounding box is defined as the mean squared error between the true value and predicted value of the multiplied bounding box value.

```
@tf.function
def loss_bb(y_true, y_pred):
    y_true = tf.gather(y_true, idx_bb, axis=-1)
    y_pred = tf.gather(y_pred, idx_bb, axis=-1)

    loss = tf.keras.losses.mean_squared_error(y_true, y_pred)
    return tf.reduce_mean(loss[loss > 0.0])
```

Figure 55: bounding box loss function

#### Probability score loss

The loss function for the portability score is defined as binary cross-entropy loss between true value and predicted value. Where an accurate prediction will be 1 and inaccurate prediction will be 0.

```
@tf.function
def loss_p(y_true, y_pred):
    y_true = tf.gather(y_true, idx_p, axis=-1)
    y_pred = tf.gather(y_pred, idx_p, axis=-1)

    loss = tf.losses.binary_crossentropy(y_true, y_pred)
    return tf.reduce_sum(loss)
```

Figure 56: Probability score loss function

### Classification loss

The loss function for classification label is defined as binary cross-entropy loss between true value and predicted value.

```
@tf.function
def loss_cls(y_true, y_pred):
    y_true = tf.gather(y_true, idx_cls, axis=-1)
    y_pred = tf.gather(y_pred, idx_cls, axis=-1)

    loss = tf.losses.binary_crossentropy(y_true, y_pred)
    return tf.reduce_sum(loss)
```

Figure 57: Classification label loss function

### Overall loss

Each individual loss was computed to overall loss by the sum of all losses as the final loss function.

```
@tf.function
def loss_func(y_true, y_pred):
    return loss_bb(y_true, y_pred) + loss_p(y_true, y_pred) + loss_cls(y_true, y_pred)
```

Figure 58: Overall loss function

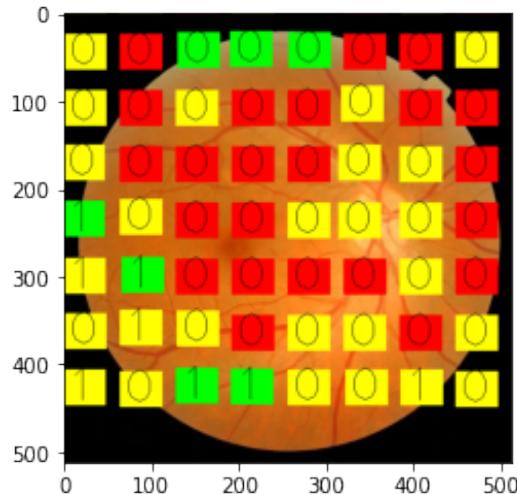


Figure 59: Prediction from untrained network

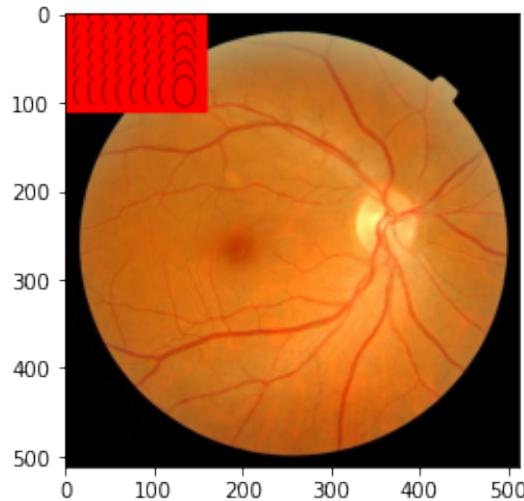


Figure 60: Prediction from trained (300 epoch) network

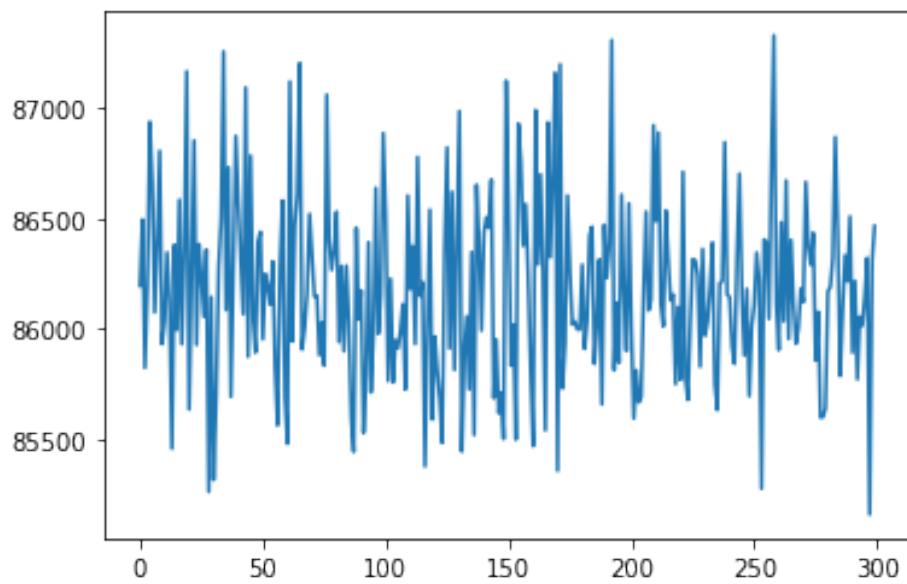


Figure 61: Training Loss over 300 epochs

We faced difficulty in getting the network to converge. There could be several causes: (i) The network is poorly constructed, (ii) the feature extraction model is not suitable for fundus use case, (iii) loss function is poorly defined. Nevertheless, this is our first attempt to construct a custom neural network for bounding box detection with our understanding of how neural network works. There are several challenges in building a custom network, including our limited understanding and experience with the Tensorflow and Keras library. Having said that, it was worthwhile experimenting, and we decided to move on using ready-made libraries for bounding box detection.

### Darknet + YOLO

After facing difficulties in constructing a custom bounding box detection neural network, we decided to adopt existing solutions for this purpose. We evaluated various options, and decided to try out the recent You-Only-Look-Once (YOLO) architecture [10]. The YOLO network is a form of Single Shot Detector approach that was proposed by Meta [10]. Briefly, YOLO network divided the image into several grid with a set of pre-defined bounding boxes. For each bounding box, there will be a object classification. Each positive classification will check for bounding box overlap (Intersection over union). For overlapping boxes that exceed certain threshold, a single bounding box will be selected using Non-Max Suppression approach.

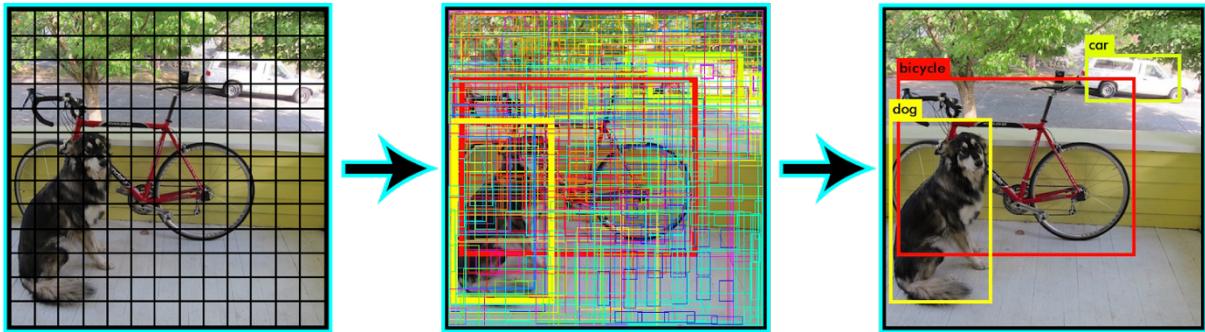


Figure 62: Illustration of YOLO approach (Source: <https://arxiv.org/pdf/1506.02640.pdf>)

In our work, we adopted a variant of YOLO network by Ultralytics, knowns as YOLOv5 [11]. YOLOv5 has been pretrained with COCO dataset and we adopted the following model architecture for our use case.

```

from n    params module
0      -1 1    3520 models.common.Focus
1      -1 1   10560 models.common.Conv
2      -1 1   19904 models.common.BottleneckCSP
3      -1 1   73984 models.common.Conv
4      -1 3   161152 models.common.BottleneckCSP
5      -1 1   295424 models.common.Conv
6      -1 3   641792 models.common.BottleneckCSP
7      -1 1   1180672 models.common.Conv
8      -1 1   656986 models.common.SPP
9      -1 1   1248768 models.common.BottleneckCSP
10     -1 1   131510 models.common.Conv
11     -1 1       0 torch.nn.modules.upsampling.Upsample
12     [-1, 6] 1       0 models.common.Concat
13     -1 1   378624 models.common.BottleneckCSP
14     -1 1   33024 models.common.Concat
15     -1 1       0 torch.nn.modules.upsampling.Upsample
16     [-1, 4] 1       0 models.common.Concat
17     -1 1   95104 models.common.BottleneckCSP
18     -1 1   147712 models.common.Concat
19     [-1, 14] 1       0 models.common.Concat
20     -1 1   313088 models.common.BottleneckCSP
21     -1 1   590336 models.common.Conv
22     [-1, 10] 1       0 models.common.Concat
23     -1 1   1248768 models.common.BottleneckCSP
24     [17, 20, 23] 1   24273 models.yolo.Detect

```

Figure 63: YOLOv5 architecture adopted for training

## Intelligent system for eye diagnosis

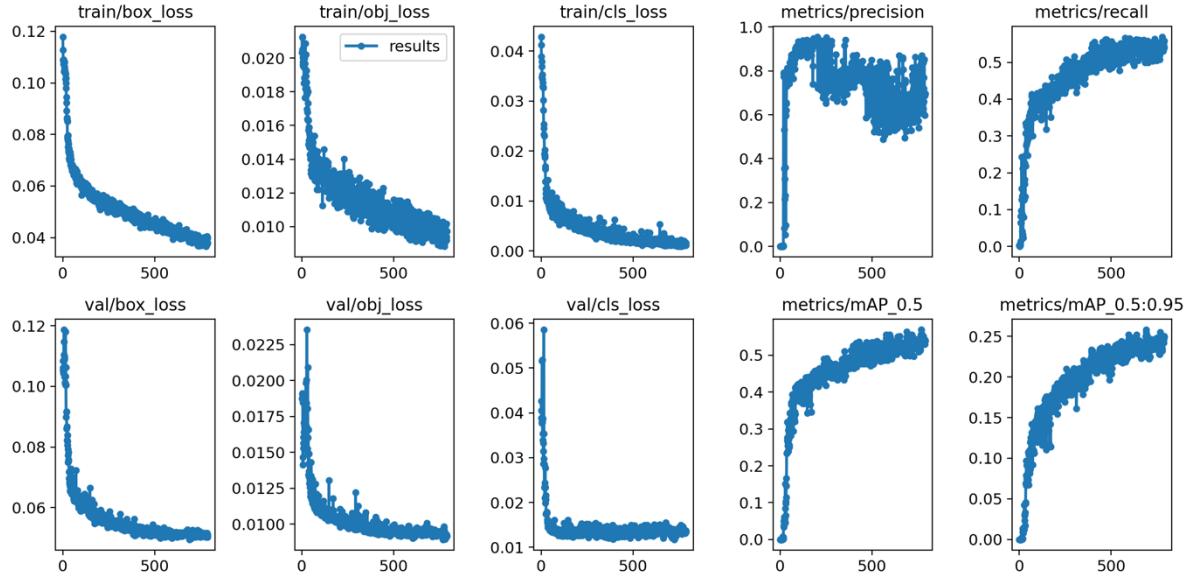


Figure 64: Training loss and performance metrics over 700 training epochs

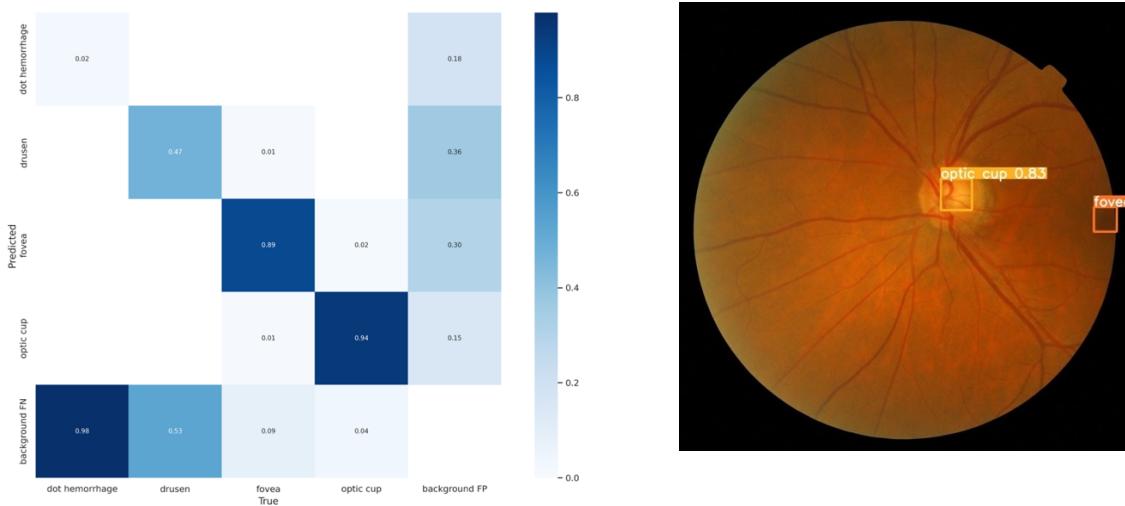


Figure 65: Confusion matrix of trained network after 700 epochs (left) and Example output of the model (right)

We trained the model with different number of epochs and found that the training loss did not further reduce after 700 epochs. Based on the confusion matrix, the model is able accurately identify “fovea” and “optic cup”. However, it does not perform as well for dot “haemorrhage” and “drusen”. When evaluating the loss value, the model can perform on all 3 tasks: locating the bounding box, identifying the correct number of objects in an image and identifying the correct class. By evaluating the Mean Absolute Precision at 0.5 threshold, we can see that the model is able to accurately identify the bounding box position after around 500 epochs (mAP@0.5 above 0.5).

## Diagnosis Reasoning System

	Optic disc	Optic cup	Myelinated nerve fibre	Fovea	Artifact	Microaneurysm	Blot haemorrhage	Dot haemorrhage	Hard exudate	Cotton wool spot	Drusen
Normal	Present	Present	Present	Present	Absent	Absent	Absent	Absent	Absent	Absent	Absent
mild non-proliferative diabetic retinopathy	Present	Present	Ignore	Present	Absent	Present	Absent	Absent	Absent	Absent	Absent
moderate non-proliferative retinopathy	Present	Present	Ignore	Present	Absent	Absent	Either one or more present			Absent	Absent
Age-related macular degeneration	Present	Present	Ignore	Present	Absent	Absent	Absent	Absent	Absent	Absent	Present

Table 1: Logic for Diagnosis Reasoning System

Based on the features detected in the fundus image, we can then use that information to make inferences on the status of the patient's retina. We followed the deductive thinking process of a physician when evaluating a fundus image and referenced existing disease classification systems to ultimately develop the logic presented in Table 1. In our system, we combined the output of a pattern recognition system and a reasoning system to deliver an automated diagnosis system that can provide a predicted ophthalmic diagnosis with a feature-based explanation.

## System Architecture

### System Architecture

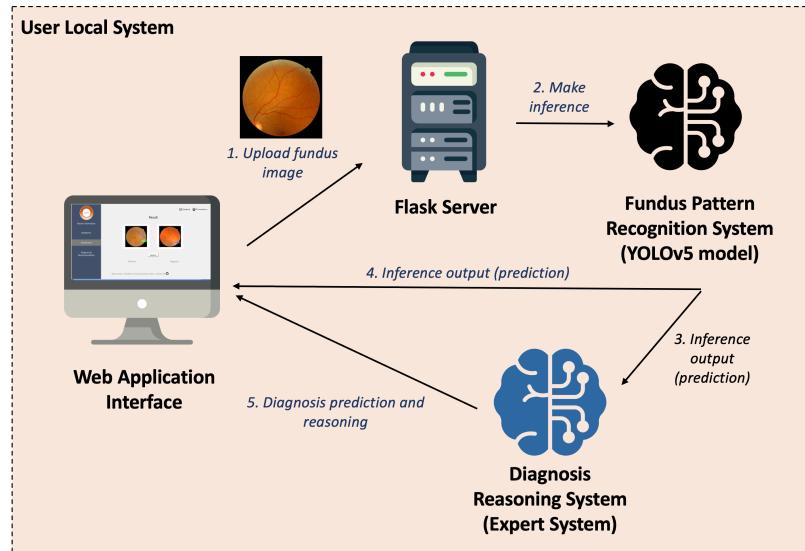


Figure 66: System Architecture

For our proof-of-concept software solution, we created a web application interface that allows the user to interact directly with the pattern recognition system. User will upload a pair of fundus images onto the web browser. This image will be sent to the backend server running on python using flask. Flask server will hand the image to the inference model running on Tensorflow to identify the labels, classification and bounding box locations. This information is then fed into a diagnosis reasoning system (expert system) to make the diagnosis inference and create an explanation text. At the same time, the pattern recognition system will also send the inference output to the web interface, allowing the expert user to see the disease labels and verify.

### System Setup

The application is a self-containing Flask server. The following are the system requirements

- A Windows or Mac OS operating system
- An ARM64 or x86\_64 CPU with at least 2.5Ghz
- Python 3.7
- Chrome Browser

Upon first set up. Unzip the content and run the following command from the folder to install the dependencies.

```
>> cd SystemCode/app
>> pip install requirements.txt
```

After the dependencies are installed. Run the following command to set the flask server.

```
>> python app.py
```

## *Intelligent system for eye diagnosis*

Once the app is up and running, run Chrome browser and navigate to <http://localhost:8005/> and use the following credentials to login to the system.

- Username: demo
- Password: demo

Some sample images are provided in the “demo” folder that can be uploaded to the application to demonstrate the inference capability.

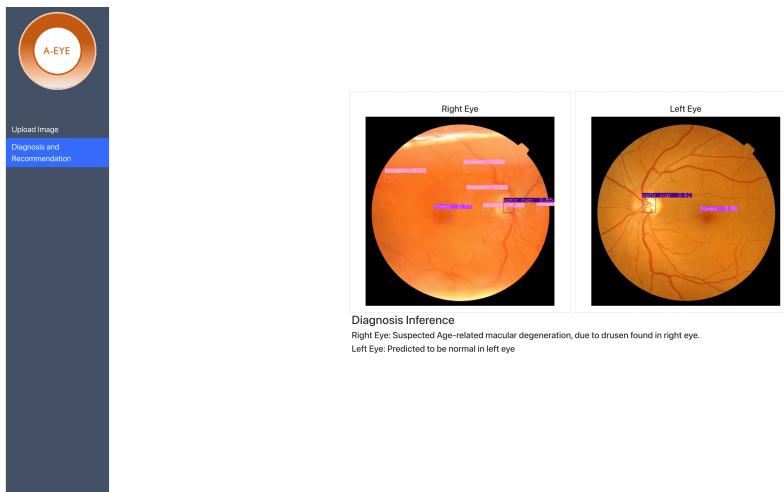


Figure 67: Example Diagnosis Inference using our system

## Conclusion

In this project, we developed a diagnosis inference system using a deep learning model for identifying features from a colour fundus photo and performed diagnosis inference using an expert system. We initially explored the use of supervised machine learning methods such as SVM classifier, Random Forest Classifier, K-nearest neighbour and Linear Discriminate Analysis (LDA) classifier. After several attempts, we realized that these models were not suitable as they did not perform well in the classification task. We reckon this could be due to several factors: (i) images were not linearly separable, (ii) insufficient feature extraction, (iii) lack of sufficient labelled samples. As we went through the training process, we decided that it would be overly tedious to hand-engineered the feature extraction for labelled data and we decided to try the neural network approach.

Initially, we had hoped to experiment with various pretrained neural network architecture for feature extraction (e.g., ResNet50, VGG16, InceptionV3). However, we soon realized that the classification architecture we constructed had several flaws such that the model was not able to converge. Due to the lack of time, we decided to adopt an a recently popular object detection model You Only Look Once (YOLO) for our use case. YOLO uses the DarkNet backbone for feature extraction and YOLO architecture for bounding box prediction. The model that we adapted was pretrained with COCO dataset. Based on our experience with supervised learning, we clean up our data set to focus on labels that are well represented and trained the model for around 700 epochs. Despite the model being able to converge on certain labels (such as, “fovea” and “optic cup”), the model still suffers from performance issue on other labels (such as, “dot haemorrhage”). This may be due to limited transferability of pretrained feature extraction model.

Colour fundus image is significantly different from the image sample from the COCO dataset. For example, the fundus images do not have a wide range of edges and colours. This may explain the limited transferability of the model. This suggests that more data samples are needed to train a new feature detection model using data samples that closely resemble those for the use case.

As mentioned in our Problem Description, the explainability of an AI system is extremely important in the healthcare industry. The ability of an AI system to explain its decision confers confidence and trust to the users (doctors and nurses), thereby increasing acceptance and adoption of the system. To achieve explainability, we consulted the domain expert on his process of reaching an eye diagnosis and, with that knowledge, co-designed an expert system for diagnosis inference. By combining the features detected by our pattern recognition system, our software is able to provide a suggested diagnosis with some explanation and present the result in full to the user.

Finally, we envision our use case can be expanded to encompass a wider range of diagnoses, with a market goal of deploying our solution to be used in tandem with commercial fundus cameras to provide real-time diagnosis suggestion to the user.

## References

1. Ministry of Health (Singapore) (n.d.). *Speech by Mr Ong Ye Kung, Minister for Health, at the Ministry of Health Committee of Supply Debate 2022*. Ministry of Health - News Release. Retrieved November 5, 2022, from <https://www.moh.gov.sg/news-highlights/details/speech-by-mr-ong-ye-kung-minister-for-health-at-the-ministry-of-health-committee-of-supply-debate-2022/>
2. Ministry of Health Singapore (n.d.). White Paper on Healthier SG. Ministry of Health. Retrieved November 5, 2022, from <https://file.go.gov.sg/healthiersg-whitepaper-pdf.pdf>
3. World Health Organisation. Blindness and vision Impairment. Factsheets. Retrieved November 5, 2002, from <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>
4. HealthXChange. Diabetic Retinopathy Screening: How SiDRP Can Help. Singapore Health Services Pte Ltd, Retrieved November 5 2022, from <https://www.healthxchange.sg/head-neck/eye-care/sidrp>
5. Ting DSW, Cheung CY, Lim G, Tan GSW et al. Development and Validation of a Deep Learning System for Diabetic Retinopathy and Related Eye Diseases Using Retinal Images From Multiethnic Populations With Diabetes. JAMA. 2017 Dec 12;318(22):2211-2223. doi: 10.1001/jama.2017.18152. PMID: 29234807; PMCID: PMC5820739.
6. Chea, Nakhim, and Yunyoung Nam. "Classification of Fundus Images Based on Deep Learning for Detecting Eye Diseases." *Computers, Materials & Continua*, vol. 67, no. 1, 2021, pp. 411–426., <https://doi.org/10.32604/cmc.2021.013390>.
7. K.M. Adal, P.G. van Etten, J.P. Martinez, L.J. van Vliet, K.A. Vermeer. Accuracy Assessment of Intra and Inter-Visit Fundus Image Registration for Diabetic Retinopathy Screening. *Invest Ophthalmol Vis Sci*. 2015.
8. Heartexlab. labellImg. GitHub. <https://github.com/heartexlabs/labellImg>
9. Seachaos. A simple way to understand and implement object Detection from scratch, by pure CNN. Medium. Retrieved November 5, 2022, from <https://medium.com/tree-rocks/a-simple-way-to-understand-and-implement-object-detection-from-scratch-by-pure-cnn-36cc28143ca8>
10. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015, June 8). *You only look once: Unified, real-time object detection*. arXiv.org. <https://arxiv.org/abs/1506.02640>
11. YOLOv5: The friendliest AI architecture you'll ever use, Ultralytics. Retrieved November 05, 2002, from <https://ultralytics.com/yolov5>