AI & Deep Learning Applications 4

I. Supervised Machine Learning and Dimensionality reduction

A. Data preprocessing

Initially, it was apparent that several patient scans had features which were nearly all missing, these patients were removed as they add too much noise to the data. Subsequently, the data was split into training and testing sets to prevent data leakage, ensuring each patient was exclusively assigned to either set. This split was stratified by patient health status to maintain a proportional representation across both sets. Following this, the dataset was divided into feature data and labels, with categorical data transformed into binary representation using one-hot encoding to eliminate any ordinal or hierarchical relationships. Furthermore, age data was standardised to align with the normalised features, preventing potential divergence during the learning process. 5-fold cross-validation was used to give a reliable estimate of model performance.

B. K-nearest neighbours (KNN)

The KNN predicts the class label or target value of a new data point by averaging the labels or values of its k nearest neighbors in an n-dimensional feature space. Increasing the number of neighbors (K) typically reduces the model's variance, resulting in more stable predictions less sensitive to data noise. However, overly large K values can lead to inflexible decision boundaries, limiting the model's ability to capture complex patterns in the data. Testing revealed that utilising eight neighbors achieved the optimal balance. Various distance metrics, including Euclidean, Manhattan, and Minkowski distances, are used to measure the proximity between neighbors in KNN models. Euclidean distance, which calculates the straight-line distance between two points in n-dimensional space, is the most widely used metric. It's noteworthy that Euclidean distance is sensitive to scale, necessitating standardisation to ensure that features are on a similar scale. Training results demonstrated the effectiveness of Euclidean distance, which aligns with expectations given that the dataset primarily comprises continuous variables.

C. Support Vector Machine (SVM)

SVMs aim to discover the ideal hyperplane for class separation, maximising the margin between classes. By employing kernels, SVMs can capture intricate relationships by projecting data into higher-dimensional spaces. A grid search was performed, indicating that a linear kernel yielded the highest accuracy, implying the dataset's potential linear separability. Choosing a smaller regularisation parameter, c, facilitates a wider margin (resulting in high recall) but may permit miss-classifications (leading to low precision). To strike a balance, a regularisation parameter of 0.3 was selected, optimising the trade-off between margin width and classification accuracy.

D. Artificial Neural Networks (ANN)

ANNs execute classification tasks by processing input data through layers of neurons, leveraging weighted connections and non-linear activation functions in hidden layers to generate final classification outputs. Throughout training, ANNs iteratively adjust connection weights and biases via supervised learning algorithms such as back-propagation, with the goal of minimising the disparity between predicted and actual outputs. In this particular task, the chosen ANN architecture consisted of an input layer followed by two hidden layers, featuring 16 and 32 neurons, respectively. Each hidden layer employed Rectified Linear Unit (ReLU) activation functions to effectively capture the intricacies present in the data. Initially, the model exhibited signs of overfitting, as evidenced by the continual rise of the training curve while the validation performance plateaued. To mitigate overfitting, L2 (ridge) regularisation was introduced to the hidden layers, preventing excessively large coefficients from fitting too closely to the dataset.

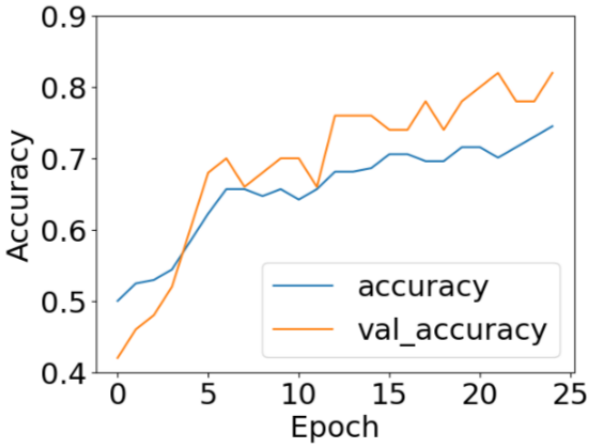

Fig. 1: ANN Model accuracy training curve against epoch number

E. Model Comparison

In the context of MS diagnosis, achieving high recall is paramount as it ensures the accurate identification of most MS cases, facilitating early intervention for improved patient outcomes. Precision is also crucial to mitigate false positives, where healthy individuals are incorrectly diagnosed with MS, preventing unnecessary stress and reducing the burden on healthcare resources. The F1 score serves as a valuable metric, striking a balance between precision and recall, particularly beneficial in imbalanced datasets with limited occurrences of MS. The Receiver Operating Characteristic (ROC) curve shown by figure 2 illustrates the ability for each model to discriminate between positive and negative classes respectively by plotting the true-positive rate (TP) against the false-positive

rate (FP) for different threshold settings. The Area-Under-Curve (AUC) metric evaluates the discrimination power of each model across a variety of thresholds. The Precision-Recall curve shown by figure 3 which illustrates the trade-off between precision and recall. SVM models have the greatest AUC score for both ROC and PR curves indicating that they are the most versatile models across the range of thresholds. Given the MS-diagnosis dataset's characteristics—comprising approximately 26 features and only 138 patients, the KNN algorithm can't perform as well as the curse of dimensionality distorts performance metrics. ANNs and SVMs both perform well predict in high dimensional spaces as ANNs can extract important features from the data and SVMs can find hyper-planes irrespective of the dimensionality. Table II show SVMs outperform ANNs in all performance metrics likely because SVMs have hyper-parameters to reduce overfitting.

Model	Precision	Recall	f1 score	Accuracy
KNN	0.751	0.792	0.816	0.782
SVM	0.903	0.963	0.932	0.927
ANN	0.835	0.937	0.881	0.847

TABLE I: Performance metrics of each KNN, SVM and ANN

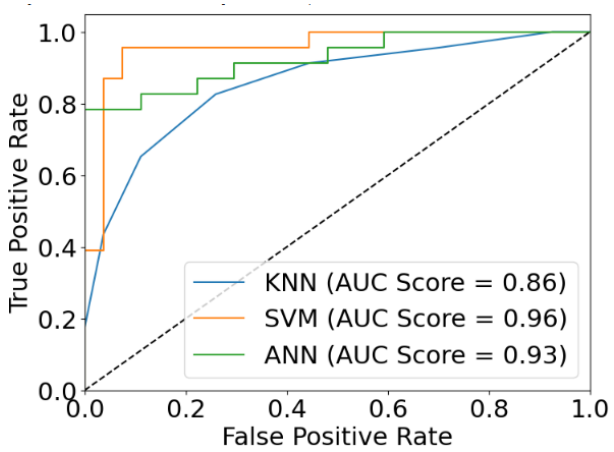

Fig. 2: Receiver Operating Characteristic (ROC) curve for KNN, SVM and ANN with Area under Curve (AUC) score

F. Dimensionality reduction

Dimensionality reduction is a technique used to reduce the number of features whilst preserving as much relevant information as possible. When features are removed, the volume of the data decreases exponentially and reduces the computational cost of training a predictive model, furthermore predictive models are less likely to overfit to datasets with fewer features with greater variance.

1) t-distributed Stochastic Neighbor Embedding (t-SNE): t-SNE is a non-linear dimensionality reduction technique that is often used to visualise higher dimensional datasets which is useful for data exploration and feature engineering. Figures 4 and 5 show the effect of applying both PCA and auto-encoding to the data in two dimensional space.

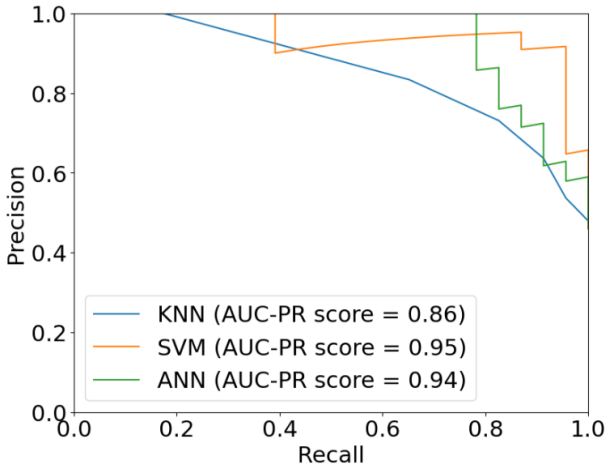

Fig. 3: Precision Recall (PR) curve for KNN, SVM and ANN with Area under Curve (AUC) score

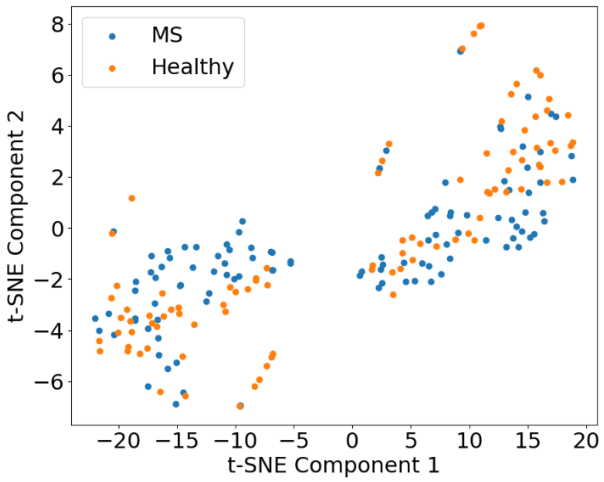

Fig. 4: t-SNE visualisation of PCA

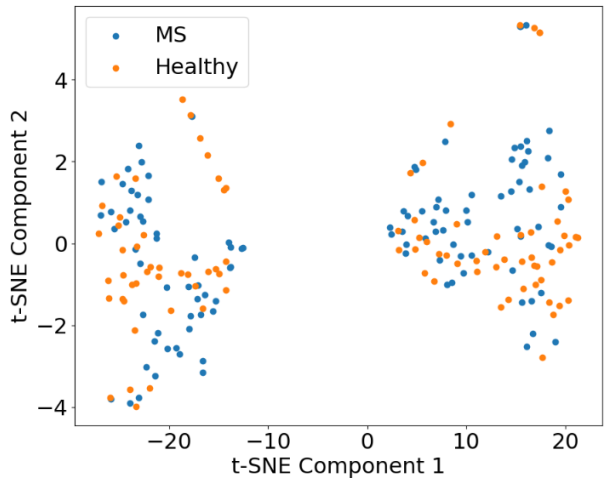

Fig. 5: t-SNE visualisation of Autoencoders

2) Principal Component Analysis (PCA): PCA projects the original data onto a lower-dimensional subspace spanned by a set of uncorrelated variables called the principal components which capture as much variance as possible. For this dataset, as few as 10 principal components capture over 98% of variance within the data.

3) Autoencoders: Autoencoders are indeed a type of ANN commonly used for dimensionality reduction. The typical architecture of an autoencoder involves an encoder network, which takes high-dimensional input data and maps it onto a lower-dimensional latent space, and a decoder network, which reconstructs the data from this compressed representation, aiming to produce an output with fewer dimensions than the original dataset. This process effectively learns a compressed representation of the input data while attempting to minimise the reconstruction error between the input and output data.

4) Comparison of results: Table II presents a comparison of evaluation metrics for SVM with the original dataset, SVM with PCA dimensionality reduction, and SVM with autoencoder dimensionality reduction. SVM with the original dataset exhibits the best performance, as SVM performance typically remains robust in high-dimensional space, while dimensionality reduction inevitably reduces the explained variance within the data. The inclusion of PCA dimensionality reduction only marginally diminishes performance, as the explained variance still retains 98% of the original dataset's information. However, autoencoders demonstrate a noticeable decline in performance compared to PCA, primarily due to their lower computational efficiency on small datasets and the potential loss of information. Figure 6 illustrates the precision-recall curve for both PCA and autoencoder-reduced datasets. While both methods exhibit similar performance at low recall, PCA is able to maintain high precision at high recall values. In summary, PCA was able to achieve nearly identical performance to the original SVM using a fraction of the data features which demonstrates the ability of PCA to reduce the computational cost of machine learning algorithms using dimensionality reduction.

Reduction	Precision	Recall	f1 score	Accuracy
SVM (Original)	0.901	0.966	0.934	0.928
PCA	0.871	0.965	0.912	0.903
Autoencoder	0.853	0.851	0.852	0.847

TABLE II: Performance metrics of each model

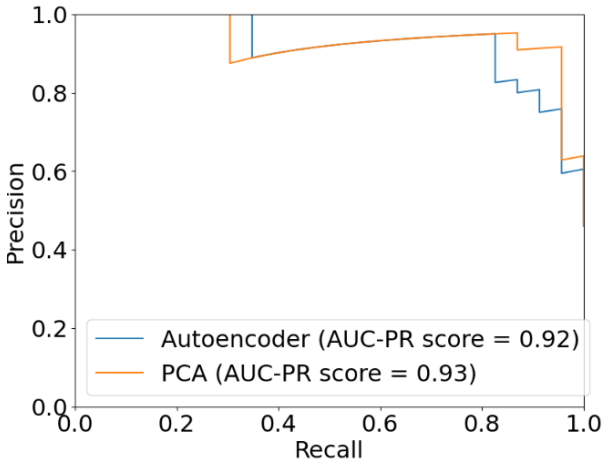

Fig. 6: PR-Curve for Autoencoders and PCA

G. Unsupervised learning

Unsupervised learning techniques can be effectively utilised with datasets related to multiple sclerosis (MS) to cluster patients based on shared features. The rationale behind this approach is that individuals with similar characteristics are more likely to exhibit similar manifestations of the condition. This approach is particularly useful when you don't have any labeled training data but you would still like to make a good prediction. The two most common clustering techniques are k-means and spectral clustering. K-means clustering groups data points by minimising distances to cluster centroids, iteratively updating centroids based on point assignments. Spectral clustering constructs a similarity graph and analyses its connectivity using spectral graph theory. It then identifies clusters based on eigenvectors associated with the smallest eigenvalues. K-means prioritises geometric closeness, while spectral clustering focuses on data connectivity.

1) Preprocessing: The first step in data preprocessing was to standardise the entire dataset, algorithms like k-means clustering calculate groups based on relative distance from each point to a cluster centre, standardisation ensures that all features contribute on an equal scale. PCA was used to reduce the dimensionality of the dataset as distances are an ineffective clustering technique in high-dimensional spaces.

2) Performance metrics: The Adjusted Rand Index (ARI) was chosen as the primary performance metric for this investigation, as it provides a quantitative assessment of the agreement between the true clustering (or ground truth) and the predicted clustering produced by a clustering algorithms. The equation of Rand Index is shown by equation 1, and it is computed for every pair of clusters. Where a+b is the number of agreements between true and predictive clustering and c+d is the total number of disagreements in true and predicted clustering.

$$RI = \frac{a+b}{a+b+c+d} \tag{1}$$

The equation for the ARI score is given by equation 2, where $RI_{expected}$ is the expected agreement if the clusters were randomly placed and RI_{max} is the maximum possible RI-score for a given pair of clustering. This means that ARI score is typically normalised between 0 and 1, where 0 is agreement expected if the clusters were placed randomly and 1 is expected if clusters perfectly aligned with the ground-truth labels.

$$ARI = \frac{(RI - RI_{expected})}{(RI_{max} - RI_{expected})} \tag{2}$$

3) Comparison of results: Spectral clustering had a higher ARI score compared to K-means suggests that it captured the underlying structure of the data better. This can be attributed to Spectral clustering's ability to detect complex patterns and relationships beyond what K-means can achieve solely based on distances. While ARI is an important metric for evaluating clustering performance, it's crucial to consider other evaluation metrics as well, such

as true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), and false negative rate (FNR). In the context of diagnosing MS, minimising the false negative rate is of paramount importance because misclassifying a patient with MS as healthy could delay critical treatment. Spectral clustering's lower false negative rate indicates its effectiveness in correctly identifying patients with MS, ensuring they receive prompt medical attention.

Algorithm	ARI	True-P	False-P	False-N	True-N
K-Means	0.0878	62	22	49	71
Spectral	0.127	76	30	35	63

TABLE III: Performance metrics of K-means and Spectral Clustering

4) Visualisation of results: Figures 7 and 8 show visualisations of the performances of k-means and spectral clustering algorithms compared to the ground truth. Spectral clustering seems to perform slightly better on edge-cases which are further away from the main distribution of data-points as it is able to leverage connectivity to find patterns.

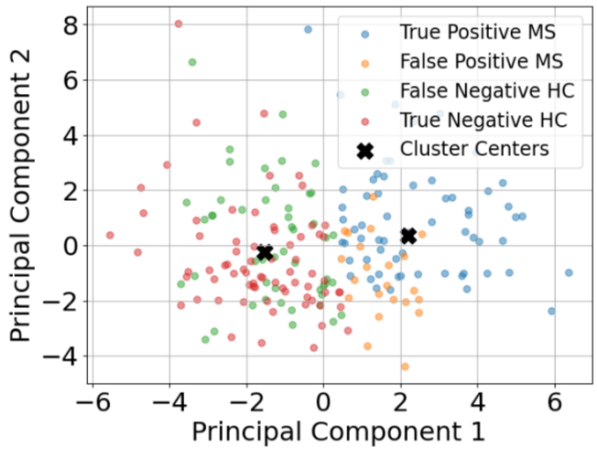

Fig. 7: Visualisation of K-means clustering in two dimensions

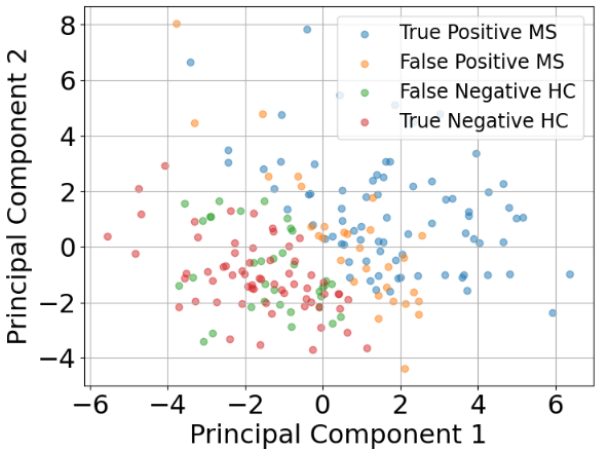

Fig. 8: Visualisation of spectral clustering in two dimensions

II. Convolutional Neural Networks

A. Introduction

Convolutional Neural Networks (CNNs) are pivotal in image classification tasks due to their ability to extract intricate features from images, often crucial indicators of diseases like Multiple Sclerosis. Typically, CNN architectures comprise stacked convolutional layers, pooling layers, and activation functions, culminating in a fully connected ANN. Convolutional layers employ filters that traverse the image, generating feature maps capturing specific edges, textures, or shapes. Subsequently, pooling layers diminish the spatial dimensions of each feature map, managing the CNN's complexity and size. Activation functions introduce non-linearity, enhancing the model's capacity to discern complex feature interactions. In the realm of MS diagnosis, CNNs learn to associate structures present in eye scans with MS-related pathologies, aiding in accurate disease detection and classification.

B. Data Preprocessing

The data preprocessing initially involved flipping retina scans to ensure consistent orientation for easier interpretation by the CNN, based on information from the SLO Excel file. Subsequently, unique patient IDs were organized into training, validation, and test sets to prevent data leakage and ensure model validity. The split was 70% for training, 15% for validation, and 15% for testing, prioritizing a larger training set for effective learning, while enabling hyperparameter tuning on the validation set and assessing generalisability on the test set.

C. Customised CNN

Hyperparameter optimisation involves fine-tuning the external settings of a machine learning model to enhance its performance, distinct from its internal parameters. These settings, known as hyper-parameters, influence the learning process. Optuna, a package for hyper-parameter optimization, was chosen for this task. The first step was to select accuracy as the performance metric due to the balanced distribution of classes in the dataset, providing a comprehensive comparison of each model's overall performance. Initially, a baseline architecture with two convolutional layers offered the best performance for the minimum computational cost. Then, the Optuna algorithm ran the CNN for a given search space and the best accuracy was found using a Gaussian algorithm. The optimised CNN architecture, illustrated in Figure 9, is designed to process grayscale images with dimensions of 256x256 pixels. The model comprises two convolutional layers, each utilizing 3x3 pixel filters. The first convolutional layer uses 5 filters, while the second employs 15 filters, as deeper layers capture more abstract features. The kernels for the first and second convolutional layers are depicted in Figures 10 and 11 respectively. When comparing the kernels from layer 1 and 2, it's evident that layer 2 captures more complex patterns, whereas feature map one represents simple edges which demonstrates the customised CNN performing hierarchical feature extraction.

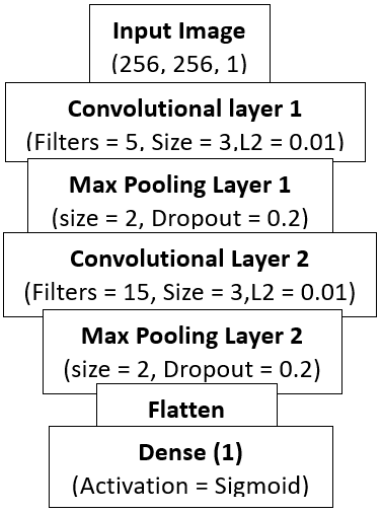

Fig. 9: Architecture of custom CNN

Fig. 10: Visualisation of the kernels in the first convolutional layer

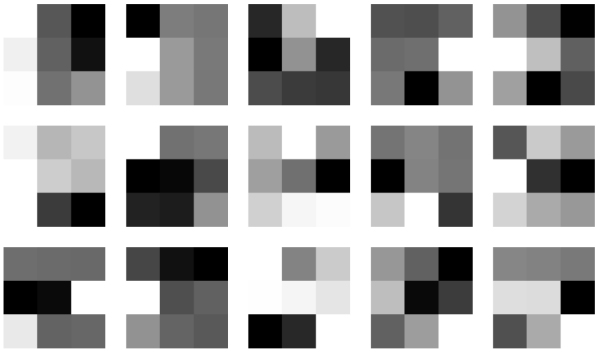

Fig. 11: Visualisation of the kernels in the second convolutional layer

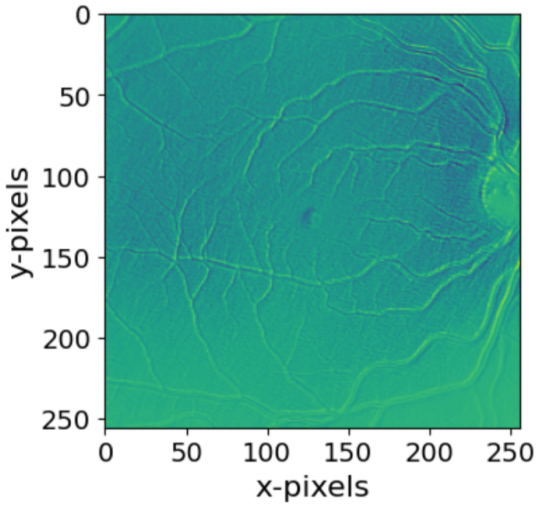

Fig. 12: Visualisation of the kernels in the second convolutional layer

The first kernel of convolutional layer 1 shown in figure 10 detects horizontal lines, the resulting activation map on a retina scan images shows is shown by figure 12. The activation map detects the basic edges of the structure, then deeper layers form more abstract activation maps that detect features specific to MS diagnosis. After Optuna optimisation, L2 regularisation was added to convolutional layers 1 and 2 to prevent large weights in the CNN from over-fitting to training data. Dropout was added after each convolutional layer to prevent co-adaptations between neurons which detect features specific to the training set. The resulting loss curve is shown by figure 23. Unfortunately, the loss curve still shows signs of over-fitting which could be due to the dataset lacking depth and diversity.

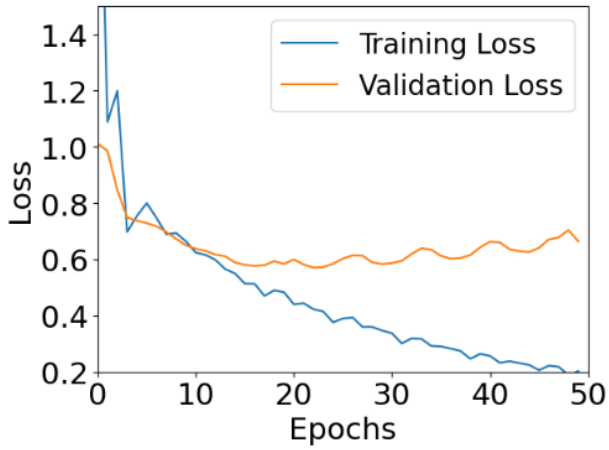

Fig. 13: Visualisation of the kernels in the second convolutional layer

D. Data Augmentation

Data augmentation is a technique used to artificially expand and enrich the dataset. In medical image classification, especially with datasets like retina scans limited by privacy concerns. Several types of data-augmentation techniques are shown by Figure 14.

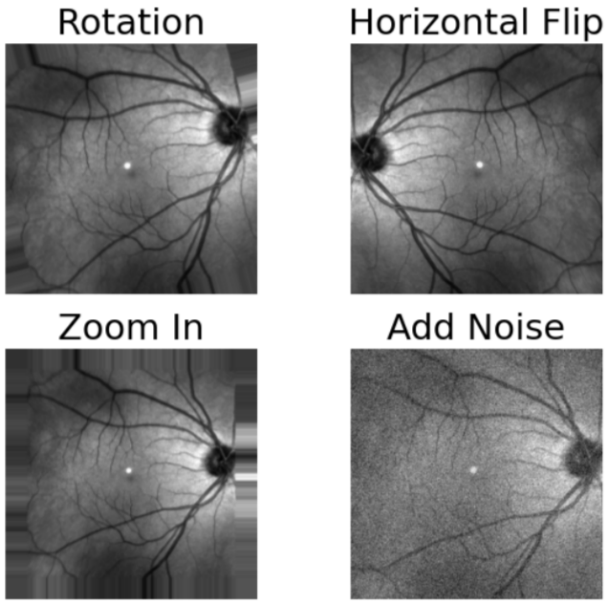

Fig. 14: Data Augmentation Techniques

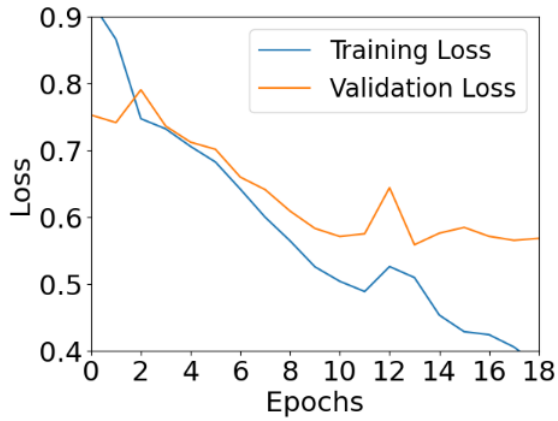

Fig. 15: Range of 2° of rotation.

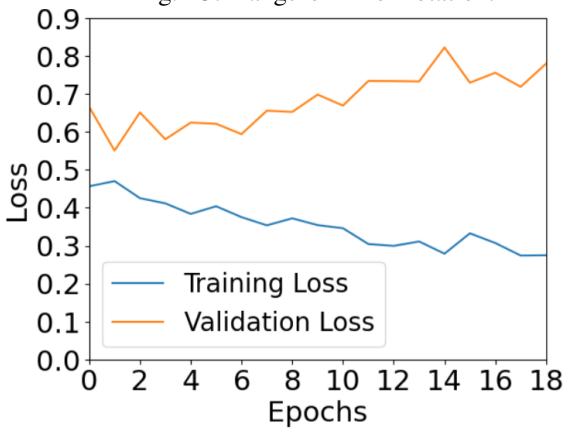

Fig. 16: Horizontal flipping.

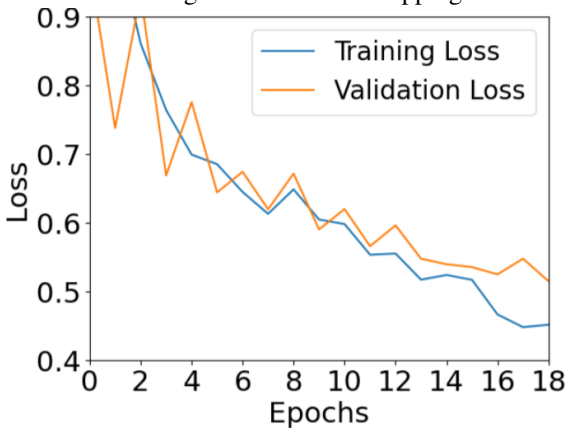

Fig. 17: 10% zoom range.

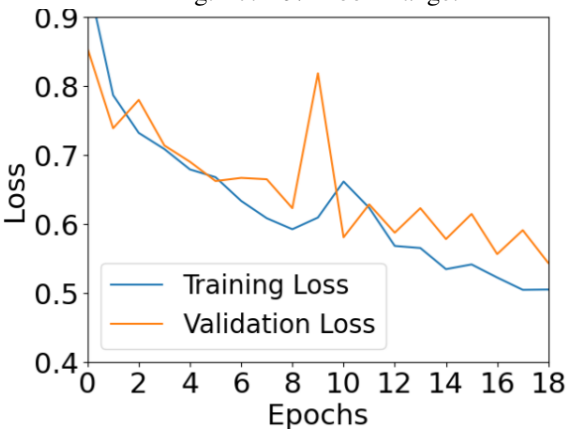

Fig. 18: Combination of flipping, 10% zoom and 2° of rotation.

Augmentation	Precision	Recall	F1-Score	Accuracy
Original	0.691	0.881	0.775	0.701
Rotation (2)	0.842	0.844	0.843	0.826
Rotation (5)	0.817	0.845	0.826	0.817
Flip (Horizontal)	0.798	0.928	0.857	0.823
Flip (Vertical)	0.749	0.732	0.731	0.728
Zoom (0.1)	0.776	0.927	0.845	0.802
Zoom (0.2)	0.743	0.921	0.827	0.777
Gaussian (10%)	0.571	0.627	0.594	0.566
Combined (Selected)	0.857	0.929	0.881	0.863

TABLE IV: Performance metrics for each augmentation technique

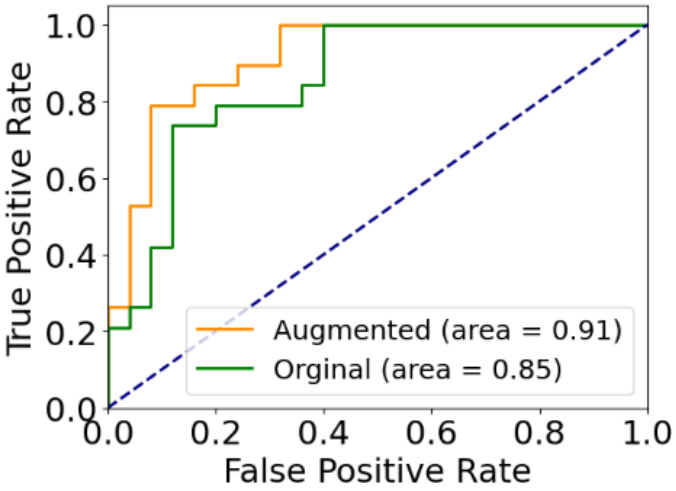

Fig. 19: ROC curve of combined augmentation and original dataset

Good augmentation techniques make models more robust to variations within test data by applying transformations which reflect the natural variation of data. Figure 15 illustrates the loss curve for 2° of rotation. Rotation is a good augmentation technique as retina scans will not always be taken at the same angle and the retina structure will vary across sample population. Augmentation with 2° of rotation increases the training loss however, decreases the validation loss significantly showing better generalisation. When the angle increases to 10° the training looses specificity to the task and the test performance decreases. Figure 16 shows that the flipping transformation causes the model to overfit to the augmented dataset, CNN models are non-linear interpolators and require more original images to predict the variation in data. Figure 17 shows the augmentation technique with 10% range of variation in zoom, this had the greatest impact on the loss curves as the validation curve aligned well with the training curve. This is because zoom not only makes the CNN more robust to variation in the images, it also allows the CNN to see an increase resolution of the retina structure to detect important features. Figure 18 shows the loss-curve of the CNN with a combination of rotation flip and zoom, the loss characteristic is similar to zoom. Table IV shows the key performance metrics from the best of five trials using each augmentation. Heavy Gaussian noise adversely affects model performance by obfuscating fine details of the retina structure. Such noise is artificial and not

representative of typical test samples with high-resolution scans. The combination of every augmentation technique had a very positive impact on precision, recall and f1 score as the CNN was able to extract better predictive features from augmentation. Figure 19 shows a comparison of the ROC curve of the original model and model with an augmented dataset, the augmented model has a better true positive prediction rate over a wide range of thresholds making it a versatile model.

E. Transfer Learning

Transfer learning involves leveraging a pretrained model on a large and diverse dataset to accelerate training and improve performance on a related task by transferring learned knowledge and patterns. This section will discuss the performance of the VGG16 pretrained model on the MS retina scan classification task. VGG16 has 16 convolutional layers and 3 fully connected layers, VGG16 was trained on 3-channel rgb data so the grey-scale data must be tripled to be used as input to the VGG16 model. Figure 20 shows the learning curve of a frozen VGG16 for the retina classification task when all of the layers are frozen. The learning loss approaches zero whilst the validation loss diverges away, this is because the output layer is able to memorise the training set whilst learning no valuable features, this means that the model struggles to predict any unseen data. The obvious next step is to unfreeze some of the top layers such that the bottom layers still retain the advantage of detecting simple shapes and edges, whilst the top layers can extract more abstract features that are specific to the retina dataset. Figure 21 shows the loss curve with 14 frozen layers and 2 trainable layers. The validation and training losses are very closely aligned however and the performance of the model is quite good. The next step should be unfreezing more layers to allow more parameters to fit to the dataset. Figure 22 shows the loss curve with 10 frozen layers and 6 trainable layers. It is clear that the validation curve follows the training curve up to about 15 epochs and then begins to diverge which suggests over-fitting. Table V shows that VGG16 with all frozen layers predicted had very good recall but poor precision, suggesting that it predicted that every single patient had MS. With 14 frozen layers and 2 free layers, the f1-score improved to 0.79 and had a good balance between precision and recall suggesting that the model was gaining more specificity to the task. Increasing the number of trainable layers further causes overfitting and a decrease in f1-score and less consistency on the test set.

No. frozen layers	Precision	Recall	F1-Score	Accuracy
16	0.573	1.00	0.721	0.571
14	0.834	0.761	0.796	0.775
12	0.716	0.882	0.795	0.727
10	0.921	0.486	0.637	0.721

TABLE V: Effect of frozen depth of unet on performance

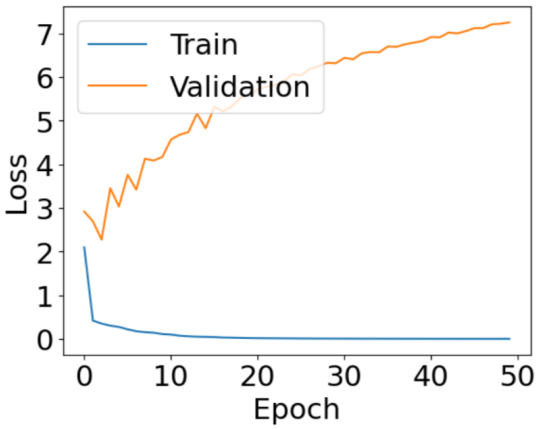

Fig. 20: Fully frozen VGG16 model

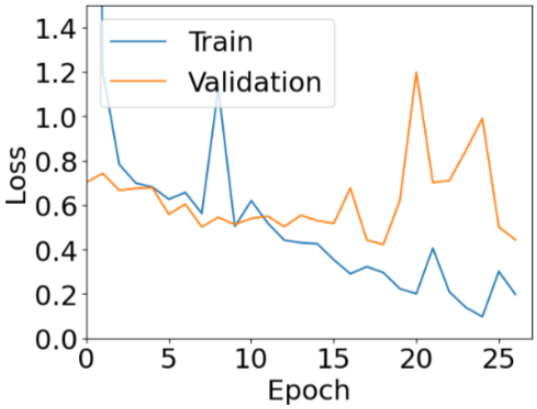

Fig. 21: VGG16 with two trainable layers

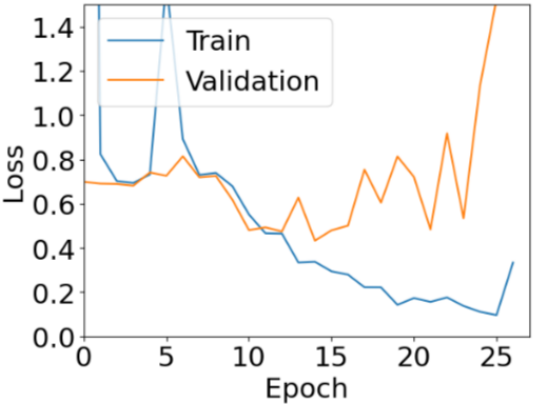

Fig. 22: VGG16 with four trainable layers

F. Conclusion

In conclusion, hyperparameter optimisation is a simple way of optimising the parameters of a model to a given task and regularisation techniques such as L2 regularisation and dropout can reduce the risk of over-fitting. The optimisation of these networks involves a combination of loss-curve analysis and performance metric evaluation. Dataset augmentation found the greatest performance benefit, by adding different transformations to the training data, the model became more robust to over-fitting and had a greatly improved generalisation performance. Transfer learning shows that careful investigation is required to determine the depth of the model which should be frozen. The retina classification task is a very specific dataset and as such the benefits of transfer learning for basic image recognition must be balanced with network specificity. The optimal number of frozen layers is about 14 for this task.

III. Retinal Vessel Segmentation

A. Methodology

During the initial preprocessing stage, the path to each directory which contained the images were defined. Then a function was defined to preprocess each image and mask pair. Preprocessing involved a conversion from 3-channel color to single-channel gray-scale and were subsequently resized to 256x256 pixels. This ensures that the model learns on a dataset which is the same format as the SLO images on which this model will be tested. Finally, the pixel values were standardised to stabilise the learning process. The pixel values were then converted to floats and scaled between 0 and 1. The masks were also standardised between 0 and 1. A map function was then applied to both train and test data simultaneously to maintain consistency. This task is an example of binary segmentation, where the U-net's objective is to classify each pixel in the scan as either part of the retina body or not. To achieve this, the sigmoid activation function was used to scale the output values between 0 and 1. Furthermore, the binary cross-entropy loss function was utilised to quantify the disparity between predicted and actual pixel classifications.

B. Performance metric

The Dice coefficient, often used in image segmentation evaluation, measures the similarity between predicted and ground truth masks. It quantifies overlap, with 0 indicating no overlap and 1 indicating perfect overlap. It's calculated as the ratio of twice the intersection of the masks to the sum of their sizes. Here, $|A \cap B|$ is the size of the intersection between predicted (A) and ground truth (B) masks, while $|A| + |B|$ is the total total size of predicted and ground truth labels.

$$Dice = \frac{2 \times |A \cap B|}{|A| + |B|} \tag{3}$$

C. Initial Model

The architecture was structured with four encoding blocks, each reducing the input dimensions from the original image. Each layer had 64, 128, 256 and 512 feature maps respectively, extracting increasingly abstract features through convolutional layers with a kernel size of 3 and ReLU activation functions. Dropout layers with a proportion of 0.2 were strategically placed to mitigate overfitting concerns. Following the encoding path, a bottleneck layer with 1024 nodes encapsulated the most critical features. Subsequently, the decoding path comprised four blocks mirroring the encoding configuration, but in reverse order, aimed at reconstructing spatial information. This symmetrical design facilitated the restoration of finer details lost during the encoding process. The integration of skip connections between corresponding encoding and decoding layers enabled the preservation of spatial information and facilitated gradient flow during training. The model was trained over 90 epochs, the loss curve is shown by figure 23, the validation loss follows the training loss which indicates that generalisation performance is very good. Figure 24 displays the dice coefficient performance increasing with the number of training epochs. Training is slow to begin with the initial selection of weights is quite poor but rapidly increases from between 10 and 40 epochs. The final dice coefficient in the test set was about 0.72, indicating the model's proficiency in delineating key features within the retina scans. Figure 25 shows the original image, ground-truth mask and predicted mask for two images. The model is able to predict the mask when the retina body is on the right or left side of the eye, this is a good indication that the model is robust to real-world variations in data. Figure 26 shows the model being tested on the SLO dataset, the mask aligns very well with the original image.

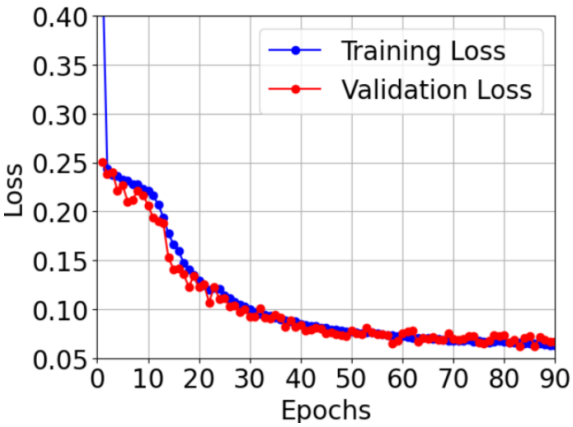

Fig. 23: Loss curve for training and validation (1st model)

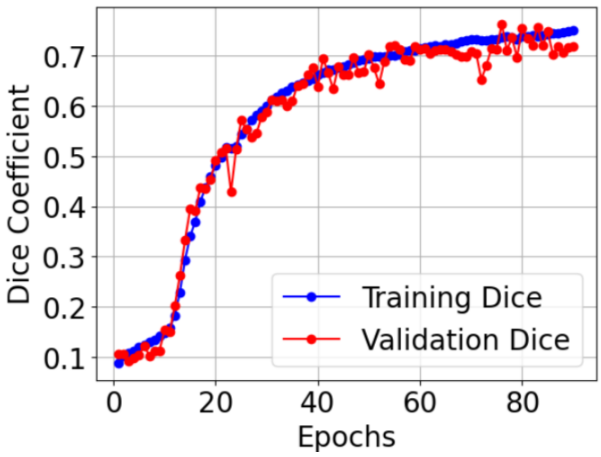

Fig. 24: Dice curve for training and validation (1st model)

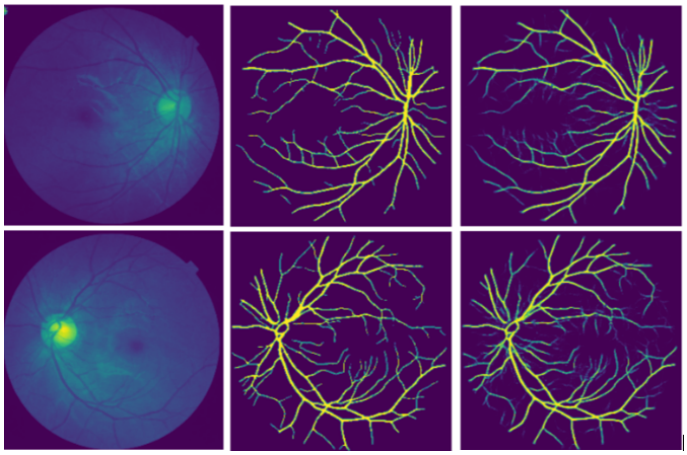

Fig. 25: Plot showing retina segmentation of fives dataset. Left: Original image; Middle: Ground Truth; Right: Prediction;

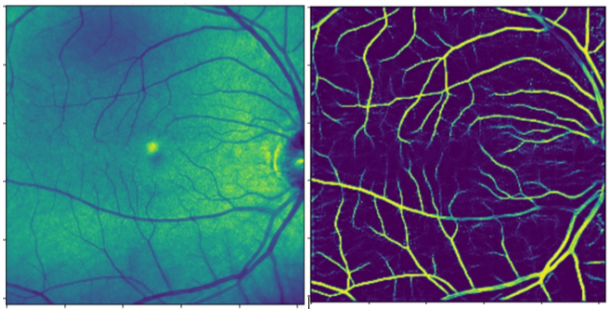

Fig. 26: Plot showing retina segmentation of SLO dataset. Left: Original image; Right: Prediction; SLO

D. Model Development

Although the original unet performed very well, it could benefit from a greater capacity as the training curves indicate underfitting due to the proximity of validation and training loss. The unet is that it takes a long time to train so it is not feasible to blindly apply a grid-search algorithm to optimise the architecture. The first step was to increase the number of convolutional layers to two per encoder/decoder block, this will allow the unet to capture deeper representations in each layer and reduce underfitting. The next step was to replace dropout with batch normalisation, because the dataset is quite small the randomness introduced by dropout prevents the network from learning effectively. Batch normalisation was applied after each convolutional layer to normalise the activation's of each layer to promote more stability in training. Furthermore, batch normalisation introduces some noise into the network to maintain some regularisation effect. The final alteration to the network was to increase the learning rate from 0.0016 to 0.01, the greater learning rate allows the network to stop getting stuck in 'local-minima' and find the best solution. Figure 27 shows the updated loss curve for the new network, the training loss approaches zero whilst the validation loss shows a lot more noise. This might be a consequence of batch normalisation not having enough regularisation effect on the deeper network. Figure 28 shows the dice-coefficient curve for training and validation data. Again, the validation curve is much more noisy than the training curve indicating a slight lack of regularisation, however the test Dice score was 82% indicating a significant improvement over the last iteration. Although the binary-cross entropy validation loss does not reach as low a point the initial model, the loss curve is mainly a tool to optimise hyper-parameters. The validation and test performance with the dice-coefficient are a more representative measure of segmentation accuracy. This improvement is validated by Figure 29 which show that the predicted masks in the updated model capture a slightly greater resolution of the retina bodies than before in Figure 25. This performance is carried over to the SLO dataset in Figure 30 for which the model more accurately represents the original image. In summary, accurate segmentation enables the identification and delineation of anatomical structures, lesions, tumors, and other abnormalities in medical images. Segmentation has been improved by using a slightly more complex network, however increased regularisation is nesesary to reduce overfitting on the more complex model.

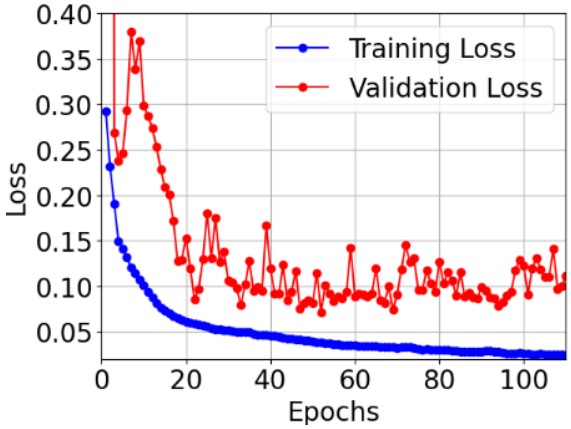

Fig. 27: Loss curve for training and validation (2nd model)

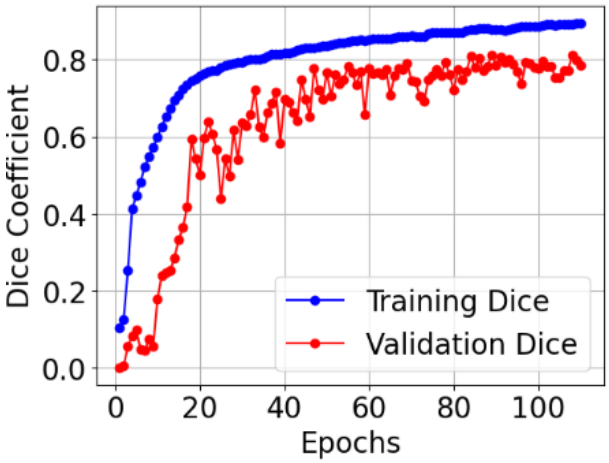

Fig. 28: Dice curve for training and validation (2nd model)

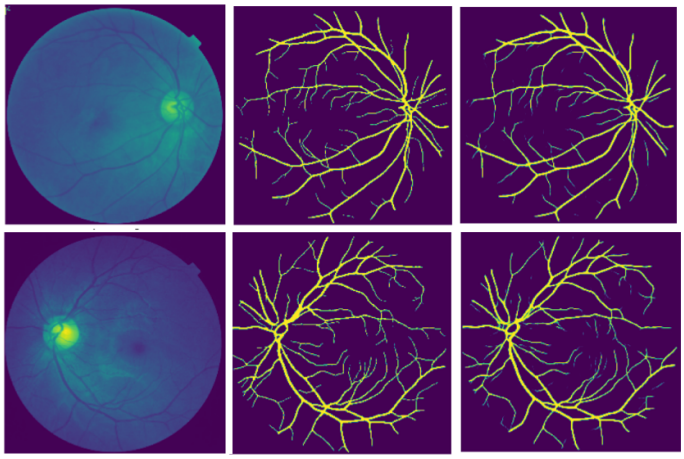

Fig. 29: Plot showing retina segmentation of fives dataset (2nd model). Left: Original image; Middle: Ground Truth; Right: Prediction;

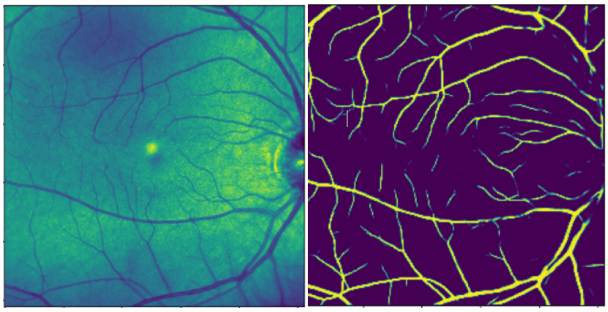

Fig. 30: Plot showing retina segmentation of SLO dataset (2nd model). Left: Original image; Right: Prediction;