Classification of Eye Diseases on Optical Coherence Tomography (OCT) Images using Swin Transformer

Viswanath Kasireddy
Department of Computer Science
Rice University
Texas, USA
Viswanath.Kasireddy@rice.edu

Somwipa Lotongkum
Department of Computer Science
Rice University
Texas, USA
Somwipa.Lotongkum@rice.edu

Jay Mapanao
Department of Computer Science
Rice University
Texas, USA
Jay.Mapanao@rice.edu

Nicholas Martin
Department of Computer Science
Rice University
Texas, USA
Nick.Martin@rice.edu

I. ABSTRACT

This paper introduces the application of Swin Transformers (ST) for detecting common eye diseases, such as Age-related Macular Degeneration (AMD) and Diabetic Retinopathy (DR). This work utilizes a Kaggle dataset of Optical Coherence Tomography (OCT) images in conjunction with Swin Transformers to provide an innovative approach aimed at improving the accuracy and efficiency of eye disease detection.

To evaluate the performance of an ST-based model, a comprehensive comparison with well-established deep learning-based image classification methods, including Convolutional Neural Networks (CNNs), Autoencoders, and Generative Adversarial Networks (GANs) was conducted. Evaluation metrics such as Accuracy, Precision, F1-score, and Recall were employed to assess the effectiveness of the proposed method both individually and in comparison to the aforementioned established methods.

The results of this study demonstrate the potential of Swin Transformers to significantly improve the diagnosis of eye diseases, offering a promising direction for future advancements in the field of medical image analysis. The efficiency and accuracy of detection of eye conditions is crucial for timely intervention and patient care, and this work contributes valuable insights into the application of state-of-the-art deep learning techniques for medical diagnosis in this area.

II. INTRODUCTION

The eyes are a vital part of the human body as they grant the ability to perceive visual stimuli, rendering them essential to the tasks needed for daily life. Since the eyes are a fairly complex organ, there are a variety of conditions affecting their health and functionality. Some of the most common eye-related conditions are Age-related Macular Degeneration (AMD), Amblyopia, Cataracts, Diabetic Retinopathy (DR), Glaucoma, Refractive Errors (RE), and Strabismus [1].

This paper focuses on the two of the most common conditions that lead to loss of vision: AMD and DR. As AMD has a variety of classifications, the authors will focus on Choroidal Neovascularization (CNV) and Drusen bodies diagnoses. For DR, the authors will focus on Diabetic Macular Edema (DME) diagnoses.

CNV is the abnormal growth of new blood vessels in the eye. It can cause visual acuity and distortion of vision[2] [3]. DME tends to be a chronic disease. Although spontaneous recovery is not uncommon, 24% of eyes with clinically significant DME and 33% of eyes with center-involving clinically significant DME will have moderate vision loss (15 or more letters on the ETDRS chart) within 3 years if untreated. Surgical therapy may be the best approach for the management of DME[4]. Drusen bodies, characterized by extracellular accumulations of lipids, proteins, and cellular debris, are observed within the layers of the retina. The sub-retinal pigment epithelium (RPE) deposits are a natural outcome of the aging process; however, the risk of developing AMD may increase based on factors such as size, number, location, and type of drusen present[5][6].

Eye disease detection has seen rapid growth in machine learning, from deep learning approaches to detect Glaucoma[7] to the use of CNNs to detect multiple eye diseases[8].

With the use of an Optical Coherence Tomography (OCT) image dataset from Kaggle[1], the authors are proposing to use an ST-based image classification model to detect and predict AMD and DR eye diseases. The results generated by the proposed model will be compared against those of other contemporary deep learning models incorporating CNNs, Autoencoders, and GANs. The evaluation process will utilize metrics such as Accuracy, Precision, F1-score, and Recall as the criteria for gauging the efficacy of the predictive outcomes.

[1]https://www.kaggle.com/datasets/ysnreddy/eye-disease-detection-dataset/data

III. Literature Review

New methods and approaches in deep learning-based image classification models have proven highly effective. [9]. Convolutional Neural Networks (CNNs) provide a foundational framework for deep neural network models in computer vision [10]. The introduction of AlexNet in 2012 marked a new era for CNNs [11]. Since then, a multitude of deep learning models have been introduced, including VGG [12], GoogleNet [13], Inception V3 [14], MobileNet [15], ResNet [16], DenseNet [17], HRNet [18], and EfficientNet [19]. Among the uses of deep learning-based image classification, a prominent application in the healthcare field is eye disease classification.

One such work takes a CNN-based approach to eye disease classification, employing the EfficientNetB3 model architecture, which is part of the EfficientNet model family. The EfficientNet architecture is centered around its compound scaling method, which optimizes the model's depth, width, and resolution, and incorporates convolutional, pooling, and fully connected layers for feature extraction and classification. The EfficientNetB3 model has been particularly promising in diagnosing eye disease through image classification, as it has demonstrated accuracy in classifying cataracts, diabetic retinopathy, glaucoma, and normal eye conditions [20].

Another work discusses various machine learning approaches for eye disease classification, including CNNs, VGG16, and Inception V3, as well as specific proprietary models such as CenterNet and DenseNet-100. In particular, the use of transfer learning was shown to improve accuracy in classifying eye conditions, and could prove to be a superior approach over conventional CNN based approaches [21].

The seminal work on ST, Swin Transformer: Hierarchical Vision Transformer using Shifted Windows by Liu et al introduced the Swin Transformer architecture. 2021 [22], with a subsequent expansion on enhancing both capacity and resolution presented in 2022 [10]. Swin Transformers provide an enhancement over traditional Vision Transformers, which are used in place of convolution layers for image processing, recognition, and object identification [23]. In particular, Swin Transformers create hierarchical feature maps by using patch merging on image patches in subsequent model layers [24]. Because the processing is limited to the local windows of the image as opposed to the whole image, the time complexity is linear and more efficient compared to other Vision Transformers.

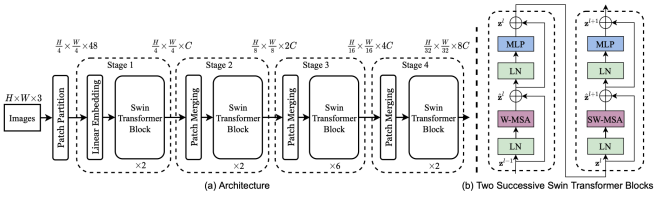

Fig. 1. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks by Liu et al. (2021) [22]

IV. Materials and Methodology

A. Image Acquisition

The dataset retrieved from Kaggle [25] was divided into train and validation sets. The train set was then divided into updated train and test sets with 80/20 splits respectively. The images are then resized to an optimal size per model. For ST, the resolution was changed to 224×224.

It is important to note that the work presented in this paper is entirely independent, and the authors have not drawn inspiration from or incorporated any results from previous Kaggle submissions or research papers utilizing this dataset. The Kaggle dataset only served as a source of diverse and real-world data to test the performance of the implemented models in a controlled environment.

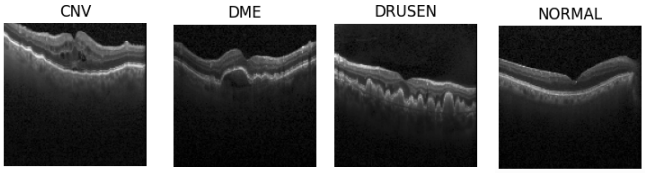

Fig. 2. Sample image of Eye Diseases [25]

B. Patch Partition and Embedding

The images are converted into Patch Embeddings with patch sizes of 4×4. The Number of features are calculated by the product of the patch size dimensions and the number of channels. With 3 as the number of channels, the number of features in one patch would be 48. With these results, the total number of patches in the whole image can be calculated through the followig equation:

$$\frac{m}{p} \times \frac{n}{p}$$

where $m \times n$ are the resolution of the image and $p \times p$ is the patch size. $\frac{224}{4} \times \frac{224}{4} = 3136$.

C. Patch Merging Layer

Modified self-attention computation, in the form of ST blocks, are employed on patch tokens in multiple stages. In the initial stage, named "Stage 1", Transformer blocks, along with linear embedding, maintain a token count of $H/4 \times W/4$. To create a hierarchical representation, subsequent stages involve reducing the number of tokens through patch-merging layers as the network deepens. The initial occurrence of patch merging and feature transformation constitutes "Stage 2", and that process is iterated again twice, resulting in "Stage 3" and "Stage 4" respectively.

In the initial patch merging step, the features of every 2×2 adjacent patches are combined, followed by the application of a linear layer on the resulting 4C-dimensional concatenated features. This results in a reduction of token count by a factor of $2 \times 2 = 4$ (equivalent to a 2× down sampling of image resolution), and the output dimension is adjusted to 2C, where

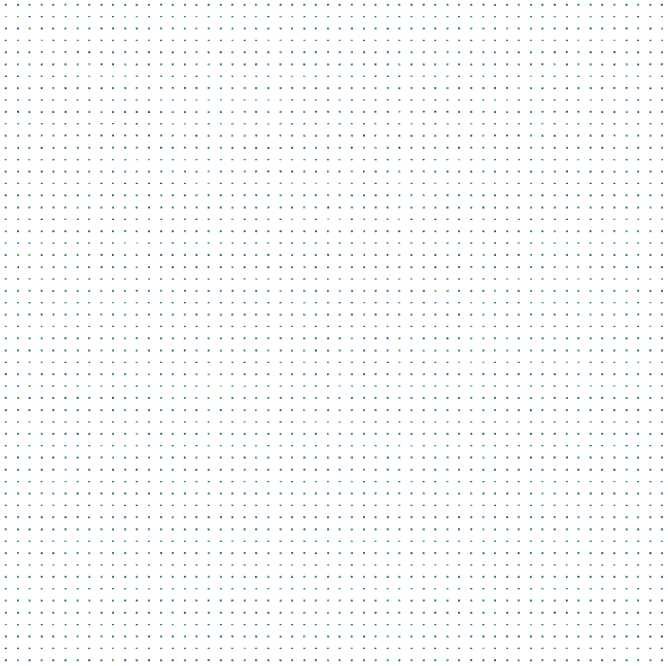

Fig. 3. Patch visualization of one image with value 3136

C is the number of channels or embedding dimensions. The number of channels used in the model is 96.

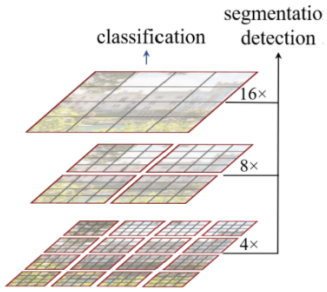

Fig. 4. Segmentation detection divided into three layers with the last layer being the classification layer [22]

The Patch-Merging layer combines four patches, leading to a successive reduction of both the height and width of the image by a factor of 2 with each merge. In Stage-1, the initial resolution is $(H/4, W/4)$, but after the patch merging process, the resolution transforms to $(H/8, W/8)$, serving as the input for Stage-2. Similarly, for Stage-3, the input resolution becomes $(H/16, W/16)$, and for Stage-4, the input resolution is $(H/32, W/32)$.

D. Swin Transformer Block

For each stage in our architecture, there are two consecutive ST blocks except in Stage-3, where there are six (6) ST blocks.

The ST is constructed by substituting the conventional Multi-head Self-Attention (MSA) module within a Transformer block with a module utilizing shifted windows, while keeping the other layers unchanged. The ST block used in this work comprises a shifted window-based MSA module, succeeded by a 2-layer Multilayer Perceptron (MLP) with a Gaussian Error Linear Unit (GELU) nonlinearity in between. Before each MSA module and each MLP, a LayerNorm (LN) layer is employed, and a residual connection is applied after each module.

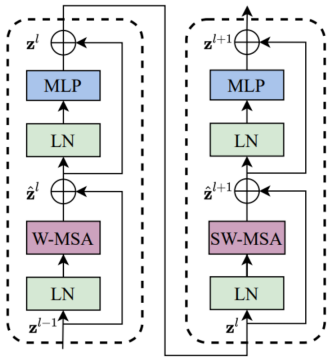

Fig. 5. Two Swin Transformer Blocks [22]

With the shifted window partitioning approach of the model, the ST-blocks are computed as:

$$\hat{z}^l = W - MSA(LN(z^{l-1})) + z^{l-1}$$

$$z^l = MLP(LN(\hat{z}^l)) + \hat{z}^l$$

$$\hat{z}^{l+1} = SW - MSA(LN(z^l)) + z^l$$

$$z^{l+1} = MLP(LN(\hat{z}^{l+1})) + \hat{z}^{l+1}$$

where $\hat{z}^l$ and z^l are the output features of Shifted Window Multihead Self-Attention (SW-MSA) module and the Multilayer Perceptron (MLP) module for block l respectively [22].

The $SoftMax$[22] function is utilized to compute the self attention of the each layer.

E. Swin Transformer v2 with Pre-trained Model

An extra model, derived from ST v2 and pre-trained using 'microsoft/swin-base-patch4-window7-224' [22], is incorporated for testing. This model has a resolution of 224×224. The dataset is trained on this pre-trained model, and the ensuing results will be compared with those obtained from the base ST model and other models.

F. Model Parameters

The model has a total parameters of 567, 252 with 554, 676 trainable parameters.

G. Evaluation Metrics

To evaluate the performance of the model, the authors use standard accuracy metrics to benchmark the performance of the Swin Transformer-based model against other models [26]. The following metrics were used to evaluate the performance of classification models [27]:

TABLE I
PARAMETERS FOR SWIN TRANSFORMER

Layer	Output Shape	Param Count
Input Layer	(None, 3136, 48)	0
Patch Embedding	(None, 3136, 96)	305,760
Swin Transformer	(None, 3136, 96)	87,224
Swin Transformer 1	(None, 3136, 96)	99, 768
Patch Merging	(None, 784, 192)	73,728
Global Average Pooling 1D	(None, 192)	0
Dense 10	(None, 4)	772

- **Accuracy**: This metric measures the ratio of correctly predicted observations to the total observations. It is given by the formula:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

- **Precision**: This metric measures the ratio of correctly predicted positive observations to the total predicted positive observations. It is given by the formula:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Recall**: Also known as Sensitivity or True Positive Rate, this metric measures the ratio of correctly predicted positive observations to all observations in the actual class. It is given by the formula:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **F1 Score**: This metric is the weighted average of Precision and Recall. It is useful when the costs of False Negatives and False Positives are different. It is given by the formula:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

V. OTHER MODELS

A. CNNs

Convolutional Neural Networks (CNNs) are a type of artificial neural network designed for processing and analyzing visual data, such as images [28]. They use convolution layers to adaptively learn features from the input data, making them well-suited for tasks like image recognition and classification. The authors used multiple architectures of Convolutional Neural Networks to provide baseline metrics for deep learning-based approaches to eye disease classification. The details of the CNNs used in this work can be found below:

TABLE II
CNN MODEL DETAILS

Model	Details
CNN	- Image resolution: 128x128 pixels - Color channels: 1 Grayscale - 2 convolution layers of 32 filters, kernel size of 7x7, Max Pooling, dropout of 0.2, and relu activation functions - 2 convolution layers of 32 filters, kernel size of 5x5, Max Pooling, dropout of 0.2, and relu activation functions - 2 convolution layers of 32 filters, kernel size of 5x5, Max Pooling, and relu activation functions - Output layer: flattened fully connected Dense Layer of output from convolution layers - Classes: Four distinct image classes
VGG19	- Image resolution: 224x224 pixels - Color channels: 3 RGB - 19 hidden layers in Keras application architecture [29].
InceptionV3	- Image resolution: 299x299 pixels - Color channels: 3 RGB - 48 hidden layers in Keras application architecture [30].
ResNet50	- Image resolution: 224x224 pixels - Color channels: 3 RGB - 50 hidden layers in Keras application architecture [31].

B. Autoencoder

Autoencoders are specialized neural networks used for unsupervised learning. They primarily focus on data encoding and dimensionality reduction. The encoder reduces the dimensionality of the data, capturing more prominent features in the process [32]. The autoencoder's feature extraction capabilities can be leveraged to perform feature classification. The details of the Autoencoder used in this work can be found below:

TABLE III
AUTOENCODER AND CLASSIFIER DETAILS

Aspect	Details
Data	- Image resolution: 128x128 pixels - Color channels: 1 Grayscale - Classes: Four distinct image classes
Parameters	- Buffer size: 16,000 - Batch size: 32 - Optimization Algorithm: Adaptive Moment Estimation - Loss Function: Categorical Cross Entropy - Activation Function: Sigmoid, ReLU, Softmax
Layers	- Conv2D (32, 64, and 128) - MaxPooling2D (2x2) - Dropout (.25) - Flatten (1D Array) - Dense (128 and 4 units, ReLU and Softmax, L2 regularization factor - 0.001)

C. GANs

Generative Adversarial Nets (GANs) are two models that are trained together: a generative model (G) that learns the data distribution, and a discriminative model (D) that predicts the likelihood of a sample being real or generated by G [33]. The goal of a GAN is to create data that D mistakenly

classifies as real, leading to a minmax game between the two. Eventually optimal point where G perfectly replicates the data distribution, and D cannot distinguish real from the generated images will be reached. In this study, the authors built G and D models to predict eye diseases with GANs conditioned on class labels [34]. The details of the GAN used in this work can be found below:

TABLE IV
GAN MODEL DETAILS

Aspect	Details
Data	- Image resolution: 128x128 pixels - Color channels: 1 Grayscale - Classes: Four distinct image classes
Parameters	- Buffer size: 16,000 - Batch size: 64 - Latent dimension: 100
Generator	- Architecture: Customized for grayscale images - Dense (None, 32768) - 5 LeakyReLU (None, 32768), (None, 16, 16, 256), (None, 32, 32, 128), (None, 64, 64, 64), (None, 128, 128, 32) - Reshape (None, 8, 8, 512) - 4 BatchNormalization (None, 8, 8, 512) 2048, (None, 16, 16, 256) 1024-param, (None, 32, 32, 128) 512, (None, 64, 64, 64) 256 - 4 Conv2DTranspose (None, 16, 16, 256), (None, 32, 32, 128), (None, 64, 64, 64), (None, 128, 128, 32) - Conv2D output (None, 128, 128, 4)
Discriminator	- Architecture: Customized for grayscale images - Conv2D: (None, 64, 64, 64), (None, 32, 32, 128), (None, 16, 16, 256) - LeakyReLU - BatchNormalization (Bat (None, 64, 64, 64) 256, (None, 32, 32, 128) 512 (None, 16, 16, 256) 1024 - Flatten and Dropout (None, 65536) - Fully connected layer Dense (None, 4)

VI. RESULTS

A. Convolutional Neural Network (CNN)

With the exception of VGG19, all attempted CNN models provided test accuracy of 80% or higher. The model that performed the best for all analyzed metrics was InceptionV3 with accuracy of 92.52%.

Each model required a specific method of preprocessing and resolution conversion. InceptionV3 had the highest resolution data of all CNN models and possibly only performed better than ResNet50 due to the higher resolution providing more information. Both models had a similar number of hidden layers and had similar preprocessing methods.

B. Autoencoder

The authors employed the Adam optimizer and utilized Categorical Cross-Entropy loss for training. The training process consisted of 20 epochs. The end result was an accuracy of 57.97%. The F1 score was 43.32%, revealing an imbalance between precision and recall. This indicates that the model had tendencies of missing cases with eye disease, while not falsely diagnosing normal results. A possible explanation for the relatively low accuracy of the model is that Optical Coherence Tomography (OCT) images are hard to extract features from, due to the complexity and detail of the image. It could also be the case that more complex models are needed over the relatively lightweight Autoencoder framework.

C. Generative Adversarial Networks (GANs)

The authors employed the Adam optimizer with a learning rate of 0.0003 and utilized Categorical Cross-Entropy loss for training. The training process consisted of 20 epochs. However, the model failed to produce meaningful results in its application of Generators and Discriminators. The F1-score remained consistently at 0.0 throughout the training process, indicating a lack of discriminating power and unsuccessful image generation. In particular the attempt to employ a Conditional GAN for image classification faced challenges related to vanishing gradients, slow convergence, and instability during training. Further investigation and modifications to the model architecture and training process may be necessary to address these challenges and improve performance in future work.

D. Swin Transformer

1) Base Model: With ten (10) epochs and four (4) classes, the results of train and test accuracy are defined in Figures 6 and 7. The train loss settled between 1.3 and 1.4 while the test loss remained constant at the 1.3 mark. Meanwhile the accuracy of train and test sets were in a constant scale of 25%. The base model did not involve any parameter tuning and default values were used for the embedding, patching and ST block layers.

As seen in the result, the embedding dimension of 96 and image resolution of 224×224 proved to be insufficient in terms of producing high accuracy on test data.

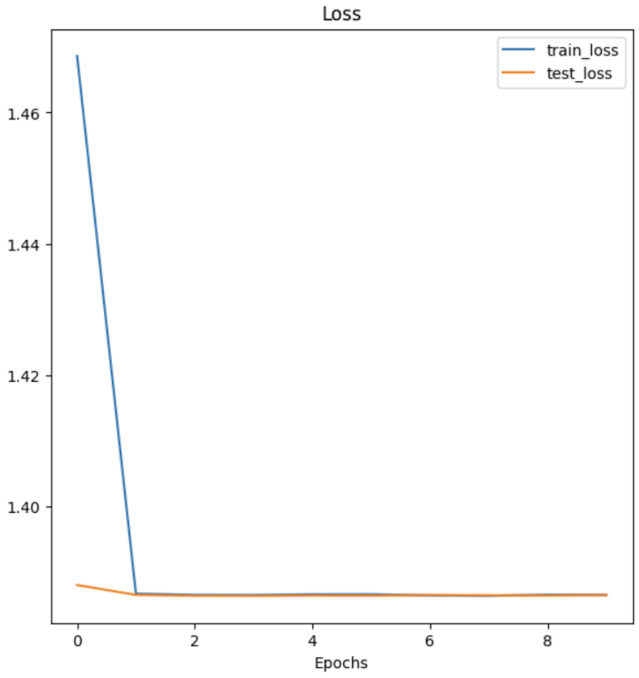

Fig. 6. Swin Transformer Base Training Results

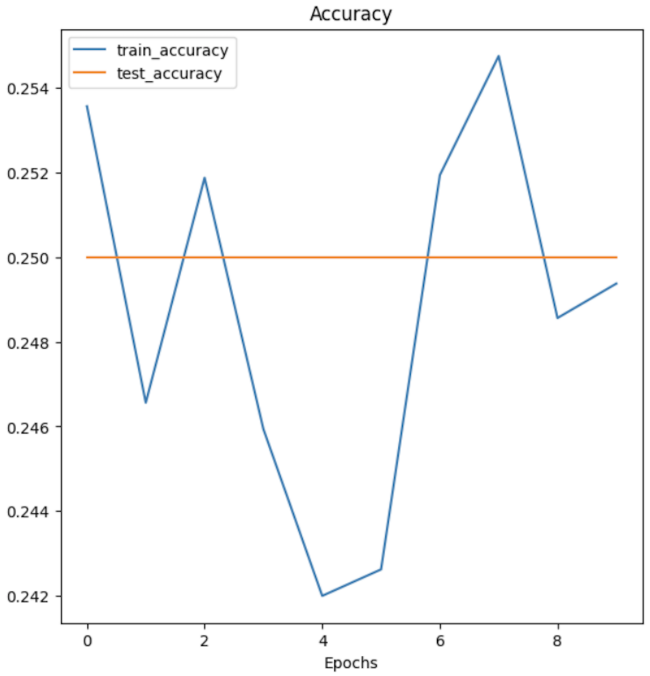

Fig. 7. Swin Transformer Base Training Results

2) With Pre-trained Model: With the pre-trained model, a significant reduction of runtime had been observed but the test accuracy remained the same as the base model.

3) Tuning the Base Model: The base model had been fine tuned through the increase of the patch size to 4×4, the dropout rate to 0.03, the MLP layer size on the ST-block increased to 256, the window size reduced to 2, and the epoch increased to 20. The training and validation scores saw a noticeable increase in performance at 96% and 80% accuracy rates respectively. The training and validation loss settled at 0.48 and 0.78. Figure 8 showed the decrease in training and validation loss as the epoch increased, an indication that the model was effectively learning and improving the ability to generalize for the eye diseases present in the dataset.

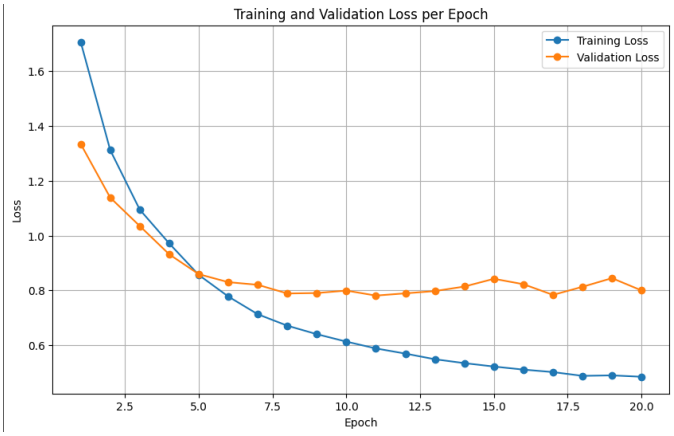

Fig. 8. Tuned Swin Transformer Loss per Epoch

The increase in accuracy for both training and validation sets as the epoch increased displays how the model improved in multi-classification of eye diseases in OCT images. The optimization of the base model proved effective as it significantly increased the accuracy of the model.

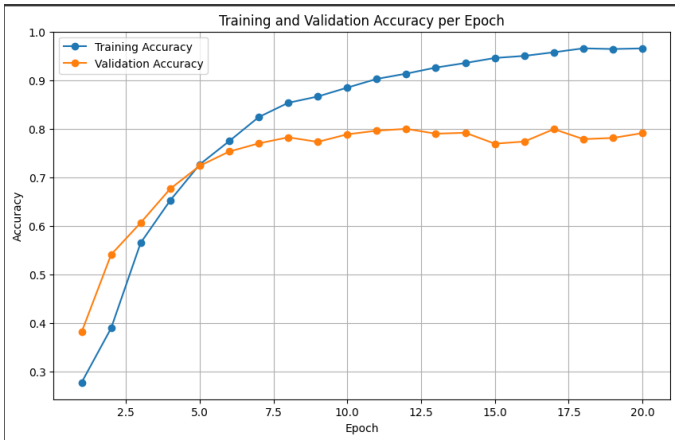

Fig. 9. Tuned Swin Transformer Accuracy per Epoch

VII. Discussion

A. Model Accuracy Comparison

The ST-model implemented in this endeavor has been compared with other novel deep learning architectures as shown in table V. Due to the relatively poor performances of the GAN and Autoencoder based approaches, we will center our discussion on the ST-based models, and the various CNN models.

TABLE V
Model Results

Model	Accuracy	F1	Precision	Recall	Loss
CNN	87.98%	87.90%	89.21%	86.68%	0.36
VGG19	73.95%	73.97%	74.12%	73.83%	1.84
InceptionV3	92.52%	92.52%	92.79%	92.27%	0.24
ResNet50	90.25%	90.27%	90.36%	90.18%	0.37
Autoencoder	56.97%	43.32%	73.14%	30.77%	1.03
GANs	-	-	-	-	-
Swin-T	25%	-	-	-	1.39
Swin-Tv2	25%	-	-	-	1.40
Swin-T Tuned	76.75%	-	-	-	0.86

For accuracy, the tuned ST performed relatively well compared to other models, although InceptionV3 performed the best with 92.52% accuracy. InceptionV3 provided the best accuracy result at epoch 15. ResNet50 and CNN also exhibited favorable performances.

This is possibly because the structure of these models employs the use of parallel convolutions and pooling layers, enabling them to use the high detail of the OCT images, which is something the ST-based models potentially struggled with. In particular, the InceptionV3 model used 48 layers, which is much deeper than the other models, and could be leveraged to capture both local and high level features.

Another reason for the comparatively poor peformance of the base Swin-T model could be that it used the default parameters across all the layers of partitioning, embedding, merging and ST blocks, resulting in the model not generalizing well. The chosen pretrained model for the Swin-Tv2 on the other hand, did not converge well with the provided dataset, resulting in around the same accuracy as the base model. When certain parameters were tuned correctly, the tuned Swin-T performed best compared to the other ST-based models, but possibly did not generalize well due to a lack of training epochs. This is due to the computational complexity of the model, which necessitates a trade off between time and accuracy, especially taking into consideration time spent in tuning parameters.

In terms of F1, Precision, Recall and Loss, InceptionV3 was the top-performing model. ResNet50 and CNN also demonstrated proficiency in learning and generalizing to the dataset.

With regard to Loss, the tuned Swin Transformer-based model displayed competence on par with the other models, and significantly reduced its loss compared to the base ST model by almost 50%. It is evident that the hyperparameter tuning for the ST model effectively optimized the model and improved the classification of various eye diseases.

B. Parameters

The authors compared a number of parameters of each model with its corresponding image sizes and accuracy to look for a significant relationship. The Swin-Tv2 model with pre-trained parameters had the largest number of parameters but possessed the lowest accuracy. This demonstrated that more parameters did not allow the model to capture any intricate patterns in the training data. However, this might be the result of the overfitting since the STv2 had complex pre-trained parameters.

ResNet50 and InceptionV3 exhibited exceptional accuracy with more parameters, proving that those models performed well in generalizing what it had learned with the training data against unseen data.

TABLE VI
TRAINED MODELS PARAMETERS

Model	Image Size	No. Param	Accuracy
CNN	128	187,108	89.21%
VGG19	224	20,057,156	73.95%
InceptionV3	299	21,835,556	92.79%
ResNET50	224	23,718,788	90.25%
GAN G	128	6,236,516	25%
GAN D	128	635,908	25%
Autoencoder	128	4,527,749	56.97%
Swin-T	224	567,252	25%
Swin-Tv2	224	87,768,224	25%
Swin-T tuned	224	554,676	76.75%

For less than 1M parameters, the CNN performed the best with less than 200000 parameters. This means that it was not only effective in terms of generalizing against unseen data and predicting eye diseases, but that it also had the capacity to produce good results in a short timespan.

The tuned ST had the potential to compete with other models in terms of accuracy vs. number of parameters as it displayed an accuracy of 76.7% with trainable parameters of 550000. With more epochs, the accuracy could have trended upwards.

C. Computational Capacity

The ST-related models took the most amount of time to run with 9, 8 and 13 hours to execute the ST-base, ST-v2 and ST-tuned models respectively. This was due to the complex patch-based processing mechanisms. ST divides the input image into non-overlapping patches and each patch had an interaction with the other patches. Moreover, the shifted window mechanism was the modified self-attention approach of some of the models. This mechanism enhanced the model's ability to capture dependencies within local regions of the input, promoting better spatial modeling. However, this came at the cost of adding complexity to the model, which resulted in increased time for model training and validation.

On the other hand, CNN and VGG19 executed the training and validation phases in 20 and 40 minutes with CNN having the better evaluation performance. InceptionV3 and ResNet50 both performed at 1.5 hours while Autoencoder and GAN came in relatively slower at 4.5 hours and 7 hours respectively.

VIII. CONCLUSION AND RECOMMENDATION

Swin Transformer-based models are generally effective in classifying the eye diseases described in this work. With the right tuning parameters and more epochs, they showed an improvement in terms of generalizing data and classifying multiple eye diseases.

The authors would recommend having multiple architecture variants for the ST models to tune the number of channels or embeddings. Conducting further analysis on the number of layers could potentially increase the accuracy rate. Increasing the epoch size on the fine-tuned parameters could potentially increase the accuracy of the model as well.

Since Swin Transformer-based models are considered as one of the highly expensive computationally, it is recommended to use a pre-trained model and tune the parameters accordingly with the dataset.

REFERENCES

[1] Centers for Disease Control and Prevention. *Common Eye Disorders*. https://www.cdc.gov/visionhealth/basics/ced/index.html. U.S. Department of Health & Human Services, n.d.

[2] J. R. Shapiro, P. H. Byers, and P. D. Sponseller. *Osteogenesis Imperfecta: A Translational Approach to Brittle Bone Disease*. Academic Press, 2014. DOI: 10.1016/C2011-0-07790-6.

[3] B. Bathija-Lala et al. "Choroidal Neovascularization in a Young, Healthy Eye after LASIK". In: *Optometry - Journal of the American Optometric Association* 81.12 (2010), pp. 632–637. DOI: 10.1016/j.optm.2010.04.093.

[4] G. H. Bresnick. "Diabetic Macular Edema: A Review". In: *Ophthalmology* 93.7 (1986), pp. 989–997. DOI: 10.1016/S0161-6420(86)33650-9.

[5] T.J. Heesterbeek et al. "Risk factors for progression of age-related macular degeneration". In: *Ophthalmic Physiol Opt* 40.2 (Mar. 2020), pp. 140–170.

[6] A.A. Bergen et al. "On the origin of proteins in human drusen: The meet, greet and stick hypothesis". In: *Prog Retin Eye Res* 70 (May 2019), pp. 55–84.

[7] B. Al-Bander et al. "Automated glaucoma diagnosis using deep learning approach". In: *2017 14th International Multi-Conference on Systems, Signals & Devices (SSD)*. IEEE. Mar. 2017, pp. 207–210.

[8] P. Muthukannan. "Optimized convolution neural network based multiple eye disease detection". In: *Computers in Biology and Medicine* 146 (2022), p. 105648.

[9] N. Motozawa et al. "Optical coherence tomography-based deep-learning models for classifying normal and age-related macular degeneration and exudative and non-exudative age-related macular degeneration changes". In: *Ophthalmol. Ther.* 8 (2019), pp. 527–539.

[10] Z. Liu et al. "Swin Transformer V2: Scaling Up Capacity and Resolution". In: *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems*. 2012, pp. 1097–1105.

[12] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *International Conference on Learning Representations*. May 2015.

[13] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1–9.

[14] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *arXiv preprint arXiv:1512.00567* (2015).

[15] Andrew G. Howard et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. https://doi.org/10.48550/arXiv.1704.04861. 2017. DOI: 10.48550/arXiv.1704.04861. arXiv: 1704.04861 [cs.CV].

[16] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[17] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4700–4708.

[18] Jingdong Wang et al. "Deep high-resolution representation learning for visual recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[19] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 6105–6114.

[20] Archana Saini, Kalpna Guleria, and Shagun Sharma. "An Efficient Deep Learning Model for Eye Disease Classification". In: *2023 International Research Conference on Smart Computing and Systems Engineering (SCSE)*. Vol. 6. 2023, pp. 1–6. DOI: 10.1109/SCSE59836.2023.10215000.

[21] Tareq Babaqi et al. *Eye Disease Classification Using Deep Learning Techniques*. 2023. arXiv: 2307.10501 [cs.CV].

[22] Z. Liu et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows". In: *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[23] Chen Wei et al. *High-Resolution Swin Transformer for Automatic Medical Image Segmentation*. 2022. arXiv: 2207.11553 [cs.CV].

[24] M. Puttagunta, R. Subban, and N. K. B. C. "SwinIR Transformer Applied for Medical Image Super-Resolution". In: *Procedia Computer Science* 204 (2022), pp. 907–913. DOI: 10.1016/j.procs.2022.08.110.

[25] Y. S. N. Reddy. *Eye Disease Detection Dataset*. Kaggle. [Online; accessed 01-November-2023]. Year of Dataset Release or Access. URL: https://www.kaggle.com/datasets/ysnreddy/eye-disease-detection-dataset/data.

[26] A.R. Wahab Sait. "Artificial Intelligence-Driven Eye Disease Classification Model". In: *Appl. Sci.* 13.20 (2023), p. 11437. DOI: 10.3390/app132011437.

[27] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[28] Rikiya Yamashita et al. "Convolutional neural networks: an overview and application in radiology". In: *Insights Imaging* (2018).

[29] Keras. *VGG*. https://keras.io/api/applications/vgg/. Keras, n.d.

[30] Keras. *inceptionV3*. https://keras.io/api/applications/inceptionv3/. Keras, n.d.

[31] Keras. *ResNet and ResNetV2*. https://keras.io/api/applications/resnet/#resnet50-function. Keras, n.d.

[32] G. E. Hinton and R. R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786 (2006), pp. 504–507.

[33] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[34] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: *arXiv preprint arXiv:1411.1784* (2014).