Eye Disease Detection using Swin Transformer

Omole Olakunle A.
Handong Global University
Policy Competency based on ICT Convergence
22347026@handong.ac.kr

Bai Jie Cao
Ede Christian University of A.S. (CHE)
Information, Communication and Technology
bjcao@student.che.nl

Maxine Olexa Phoa
Soegijapranata Catholic University
Information Systems
20n40010@student.unika.ac.id

Abstract

Nearly 2.2 billion people globally suffer from vision impairment, emphasizing the need for effective diagnostic tools. This project presents an approach utilizing the Swin Transformer to classify eye images as normal or affected by cataract, diabetic retinopathy, or glaucoma. With a dataset comprising 4,217 labeled images, the project evaluates the performance of Swin Transformer architectures, focusing on the Tiny variant, achieving an impressive 91.94% accuracy. The findings showcase the Swin Transformer's effectiveness in enhancing the accuracy of medical image classification, suggesting a viable alternative for automated diagnostic systems.

1 Introduction

Globally, almost half of the vision impairment can be either prevented or has not been addressed. According to the World Health Organization, approximately 2.2 billion people worldwide are affected by near or distance vision impairment. In various eye conditions, early detection is vital to prevent vision loss. Cataract, diabetic retinopathy, and glaucoma are amongst the leading causes of vision impairment and blindness [1].

This study aims to employ machine learning methods to categorize eyes into either normal or different eye condition classes. In 2021, the Swin Transformer, a general-purpose computer vision transformer backbone, was introduced. Notably, it builds hierarchical feature maps and exhibits linear computational complexity concerning image size. As highlighted by [2], a conventional CNN model achieved an 84% accuracy in distinguishing between a normal eye and one with diabetic retinopathy, cataract, or glaucoma, while a model with transfer learning achieved 94% accuracy. Considering CNNs as fundamental backbones for vision and recognizing the Swin Transformer as serving as a general-purpose backbone for computer vision [3], we advocate for the integration of Swin Transformer as a promising alternative.

1.1 Model Description

Swin (short form of Shifted Windows) Transformer is a type of vision transformer that creates hierarchical feature maps by merging image patches in deeper layers. It is suitable for image classification and dense recognition as it manages input image size with linear computation complexity. Unlike previous vision Transformers, it produces feature maps of varying resolutions and has a more efficient computation approach, which reduces complexity.

1.2 Project Aim

The primary objective of this project is to develop and evaluate a robust multi-class medical eye image classification system using the Swin Transformer architecture, aiming to enhance accuracy, and contributing to the research of automated diagnostic systems.

1.3 Data

The dataset utilized in this project is sourced from an opensource fundus dataset in Kaggle. It compromises cataract, diabetic retinopathy, glaucoma, and normal images for comparison, totaling 4.217 samples. Each image is labeled by folder and is publicly available on Kaggle [4]. The class distribution for cataract, diabetic retinopathy, glaucoma, and normal eyes, is as follows: 1.038 images, 1.098 images, 1.007 images, and 1.074 images.

2 Problem Statement

Accurate and timely classification is crucial for effective diagnosis and treatment planning. However, existing methods for multi-class classification of medical eye conditions have limitations. This study aims to explore an alternative approach that can overcome these limitations and improve the precision of classifying various medical eye conditions, ultimately contributing to more reliable diagnostic processes.

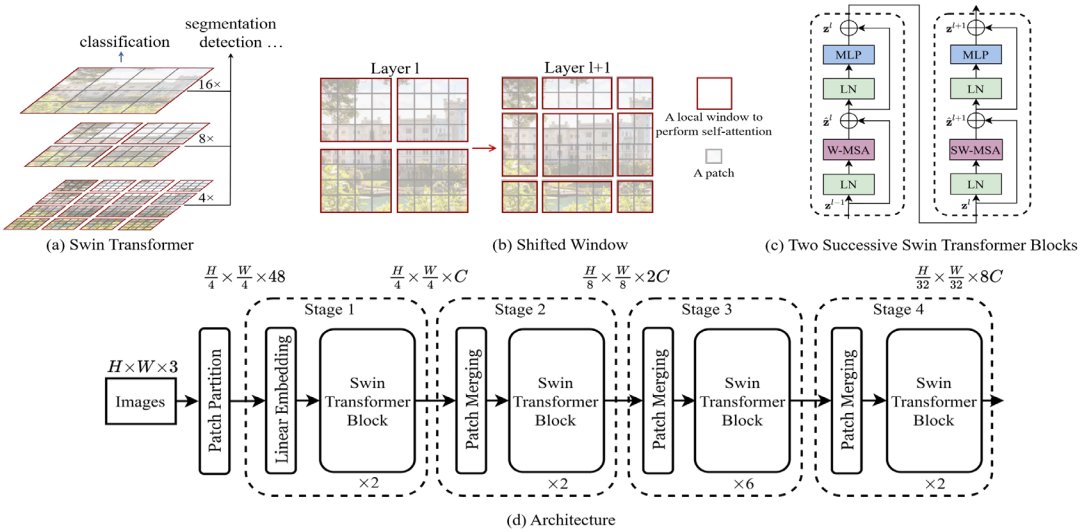

Figure 1: Swin Transformer architecture.

3 Technical Information

3.1 Swin Transformer Architecture

The Swin Transformer architecture is presented in Figure 1. It first splits an input RGB image into non-overlapping patches by a patch splitting module, like ViT. Each patch is treated as a "token" and its feature is set as a concatenation of the raw pixel RGB values. In our implementation, it uses a patch size of 44 and thus the feature dimension of each patch is 4 x 4 x 3 = 48. A linear embedding layer is applied on this raw-valued feature to project it to an arbitrary dimension (denoted as C). An innovative patch-based method is employed by the image classification model Swin Transformer. The input images are divided into fixed-size patches, which are then processed through numerous transformer layers in a hierarchical manner. Shifted windows are utilized to record dependencies between patches. Transformer blocks with self-attention mechanisms make up each layer, and positional embeddings and tokenization are used by the model to comprehend the spatial arrangement of patches. Swin Transformers efficiently captures both local and global characteristics by combining down-sampling and up-sampling techniques to manage varying spatial resolutions. The architecture of the model allows it to perform well in image categorization tasks.

3.2 Swin Transformer Variants

Different variants of the Swin Transformer exist; tiny, small, base, and large. These different variants are used for different use cases with different computational costs and needed performance for its task. All formats of the Swin Transformer have the same architecture seen in Figure 1. The main differences between the different variants are its model size, computational cost, performance, and the use of different hyperparameters to define the patch size for example. Furthermore, the number of layers may be different in the different variants of the Swin Transformer.

4 Preparations before Training

4.1 Use of Pre-Trained Model

This project utilizes a pre-trained model of the Swin Transformer in PyTorch. PyTorch makes the following Swin Transformers available for use, Swin Transformer Tiny, Swin Transformer Small, Swin Transformer Base and their respective V2 counterparts. The project will focus on the use of the original Swin Transformers.

These Swin Transformer architectures have been pre-trained on the ImageNet-1k-dataset. The weights from training on this dataset are thus like the one used in the original Swin Transformer paper [3]. The ImageNet-1k-dataset consists of 1.000 classes and consists of 1.281.167 images used for training, 50.000 images for validation and 100.000 images for testing.

This project will utilize the different available Swin Transformer architecture excluding the V2 counterpart to see the best fitting architecture on the dataset and training the best architecture further.

4.2 Dataset

To train the model, one Kaggle [4] dataset is used to train and test the model.

The following labels are used in the dataset; normal, cataract, glaucoma, and retina disease. The dataset consists of a total of 4.217 images.

These images will be augmented within the code to increase the number of images. The augmentation consists of the use of resizing, rotation, horizontal flip, cropping, and normalizing the images. The training dataset consists of 80% of the 4.217 images, the rest will be used for testing. Around 3.374 images will be used for training and 843 will be used for testing. the data augmentation will be performed on these numbers. Figure 2 shows an image from the dataset to provide some context on how the images look like in the dataset. The images will be resized to 224 x 224 to fit the Swin Transformer architecture better.

Figure 2: Normal eye image.

4.3 Evaluation Metrics

The evaluation metrics that are used to evaluate the model and its performance are training and validation loss, training and validation accuracy, F1-score, accuracy, precision, recall, and the use of a confusion matrix. The training and validation loss is mostly used during the training to evaluate the model while it is training for every epoch, the same goes for the training and validation accuracy. The F1-score is also measured for each epoch but is in this form not that important, the average or weighted F1-score will be used instead to evaluate the model.

The metrics of precision, recall and a confusion matrix will be important to use to evaluate the model's performance on the dataset. The precision metric identifies the proportion of positive classification that was correctly classified. To do this, the formula in Figure 3 is used to calculate it. *TP* stands for the true positives, and *FP* stands for the false positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Figure 3: Precision formula.

Recall is another metric that will be used to evaluate the model. Recall is a metric that calculates the proportion of the classification that were classified correctly. The formula to calculate recall is in Figure 4. With *TP* standing for the true positives and *FN* standing for the false negatives.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Figure 4: Recall formula.

At last, the confusion matrix. This will plot all the true positives and false positives and vice versa. The confusion matrix that will be used in this project will also show in what class it is falsely labeled to or correctly labeled to.

5 Results

5.1 Setup

To achieve these variant evaluation results the following setup has been used to train the different variants. The use of the learning rate 1e-3 and the use of 0.3 dropout with the choice of optimizer the AdamW algorithm. The batch size for the training and testing are both set to 64. The code is available on GitHub [5].

5.2 Swin Transformer Tiny

The Swin Transformer Tiny uses the following parameters to declare its architecture and layer properties.

Parameter	Value
Patch size	4 x 4
Embed dim	96
Depths	2, 2, 6, 2
Number heads	3, 6, 12, 24
Window size	7 x 7
Stochastic depth	0.2

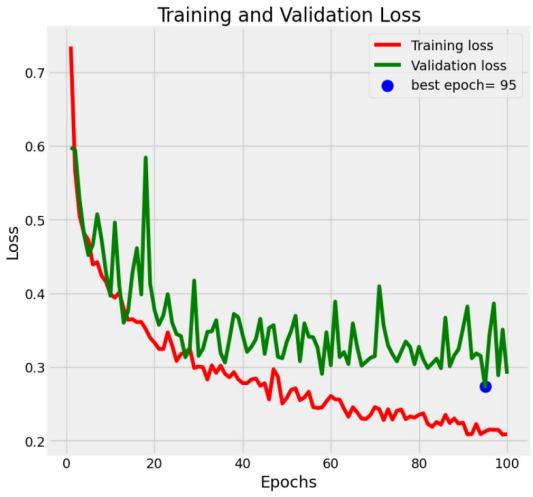

Figure 5: Training and Validation loss (Tiny).

The number of epochs is set to 100 to give the model a good chance of convergence when training. The model starts showing signs of overfitting around the 40th epoch in Figure 5. Despite the overfitting signs of the training and validation loss graph the model still outputs decent values.

The model outputs F1-score of 89.86% with an accuracy of 89.93%, precision of 0.90 and recall of 0.90. This variant can classify the images accordingly, Table 1 shows the corresponding accuracy per class.

Table 1: Accuracy per class (Tiny).

Class	Accuracy
Cataract	89.90%
Glaucoma	75.88%
Normal	95.65%
Retina	96.77%

5.3 Swin Transformer Small

The Swin Transformer Small uses the following parameters to declare its architecture and layer properties.

Parameter	Value
Patch size	4 x 4
Embed dim	96
Depths	2, 2, 18, 2
Number heads	3, 6, 12, 24
Window size	7 x 7
Stochastic depth	0.3

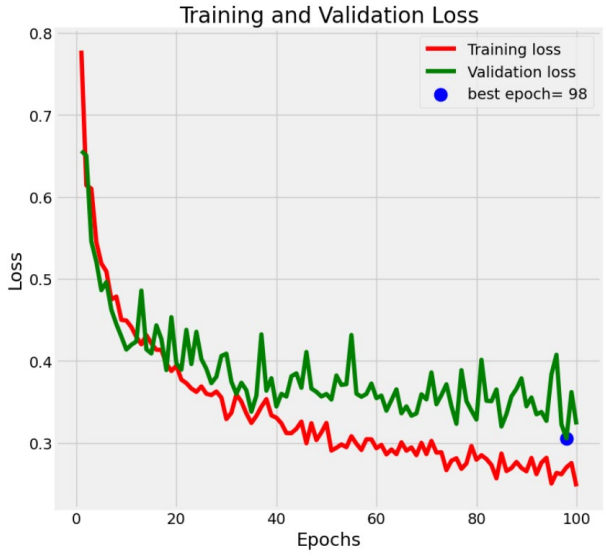

Figure 6: Training and Validation loss (Small).

The results of using the small pre-trained Swin Transformer variant are as follows. These results are achieved with a training of 100 epochs to give the model the chance to converge. The graph in Figure 6 also shows signs of overfitting after around epoch 40. Even though the model showed signs of overfitting, it still managed to score an F1-score of 88.21% with an accuracy of 88.27%, precision of 0.89 and recall of 0.88. Meaning that this model is still able to classify the images well. The accuracy of per class is depicted in Table 2.

Table 2: Accuracy per class (Small).

Class	Accuracy
Cataract	88.89%
Glaucoma	75.88%
Normal	91.74%
Retina	95.39%

5.4 Swin Transformer Base

The Swin Transformer Base uses the following parameters to declare its architecture and layer properties.

Parameters	Value
Patch size	4 x 4
Embed dim	128
Depths	2, 2, 18, 2
Number heads	4, 8, 16, 32
Window size	7 x 7
Stochastic depth	0.5

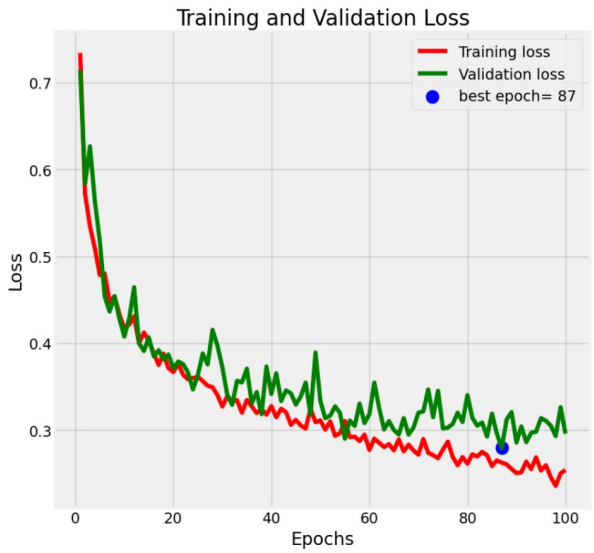

Figure 7: Training and Validation loss (Base).

When training the Swin Transformer Base on 100 epochs, the following results have been achieved. Figure 7 shows later signs of overfitting compared to the tiny and small variants; the base variant starts to overfit around 60 epochs. The model managed to score an F1-score of 89.01% with an accuracy of 89.10% and with both the

precision and recall of 0.89. The accuracy per class is depicted in Table 3.

Table 3: Accuracy per class (Base).

Class	Accuracy
Cataract	86.87%
Glaucoma	78.39%
Normal	90.87%
Retina	99.08%

5.5 Final Results

After evaluating the different variants of the Swin Transformer it can be concluded that each of the variants performs the same or within a ~1% margin of each other.

Nonetheless the Swin Transformer Tiny variant scored the best during evaluation and the choice was made to continue training using this architecture.

The final results for the Swin Transformer Tiny variant are as follows. The model has been trained for 300 epochs with the same parameters stated in paragraph 5.1.

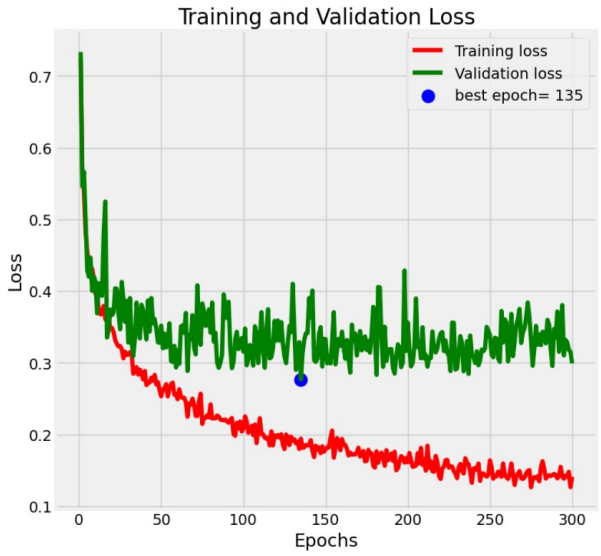

Figure 8: Training and Validation loss final (Tiny).

The final training result in Figure 8 shows overfitting after around 40 epochs, the same during evaluation. With the training loss improving over time, the validation loss keeps a stable value between 0.28 ~ 0.42. The accuracy graph in Figure 9 shows validation accuracy starting to stagnate around 100 epochs and improving slowly over time. The latter stage of the training sees the validation accuracy hanging around 87% ~ 92%.

The F1-score is 91.97%, around a 2% increase of the evaluation training. It further has an accuracy of 91.94% and for both the precision and recall a score of 0.92, both scores have been improved with 0.02 and the accuracy also saw an increase of around 2%.

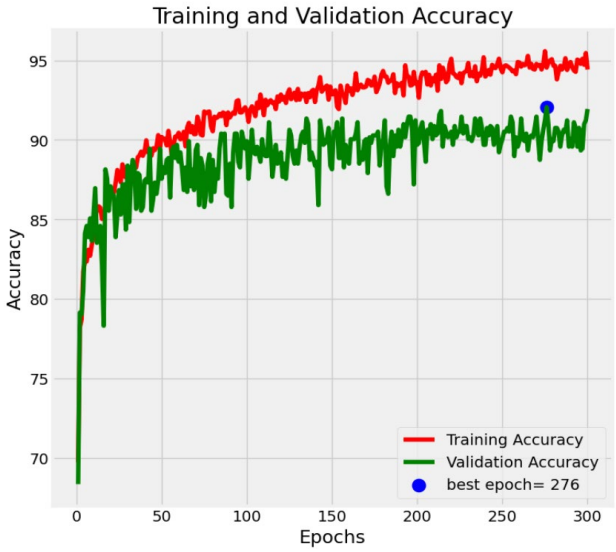

Figure 9: Training and Validation loss final (Tiny).

Accuracy for each class is depicted in Table 4, it shows an improvement compared against the evaluation result. With the glaucoma class almost improving 10%.

Table 4: Accuracy per class final (Tiny).

Class	Accuracy
Cataract	90.40 %
Glaucoma	84.92%
Normal	94.35%
Retina	97.24%

Finally, the confusion matrix in Figure 10 showing the model's performance in predicting the different classes and the number of times it predicted incorrectly. The matrix

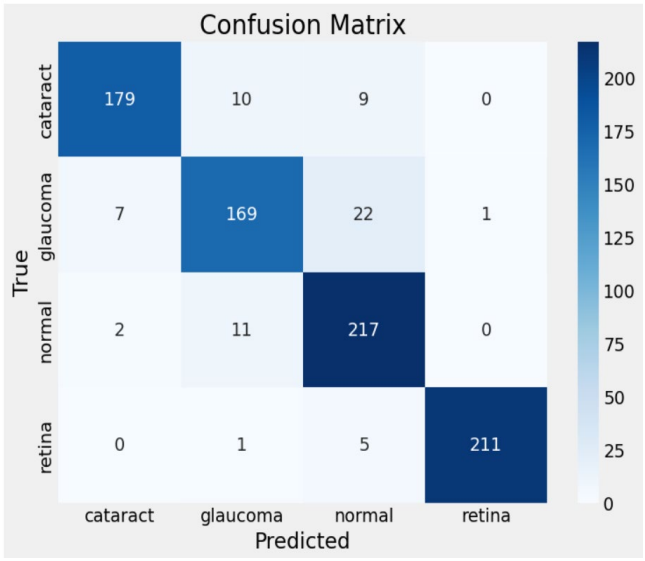

Figure 10: Confusion matrix final (Tiny).

shows better performance in predicting the different classes.

6 Result conclusion

Overall, the variants showed similar performance and results during the evaluation. The choice was made to continue training one, the tiny variant of the Swin Transformer because this variant showed the highest F1-score during evaluation.

With training the model with a higher number of epochs the model was eventually able to achieve a F1-score of 91.97%. With a similar accuracy of 91.94% and both the precision and recall on 0.92.

The Swin Transformer Tiny was thus the better architecture suited for the task of image classification of eye images, being able to capture details more precisely and making a more accurate prediction leading up to this score.

7 References

[1] Increasing eye care interventions to address vision impairment. (n.d.). https://www.who.int/publications/m/item/increasing-eye-care-interventions-to-address-vision-impairment

[2] Babaqi, T. (2023, 19 Juli). Eye disease classification using deep learning techniques. arXiv.org. https://arxiv.org/abs/2307.10501

[3] Liu, Z. (2021, 25 March). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv.org. https://arxiv.org/abs/2103.14030

[4] eye_diseases_classification. (2022, August 28). Kaggle. https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification

[5] BaiJie. (n.d.). GitHub - BaiJie90/Eye_Disease_Detection_Swin-Transformer_PyTorch: This repository contains an image classification model for the subject AI Convergence and Application held on Handong Global University (HGU). GitHub. https://github.com/BaiJie90/Eye_Disease_Detection_Swin-Transformer_PyTorch