Diabetic Retinopathy Progression Recognition Using Deep Learning Method

Xu, Shunyao
shunyaox@stanford.edu

Huang, Zixin
alisa612@stanford.edu

Zhang, Yuhan
zhangyh@stanford.edu

Abstract

This project presents a deep-learning-based approach to automatically recognize Diabetic Retinopathy progression level. There are three baseline architectures used in our experiment, which are ResNet, EfficientNet, and Swin Transformer. We apply these models to solve our problem in transfer learning, with preprocessing, data augmentation and hyper-parameter tuning to make our models more robust. Besides, we compare the performance of these models mainly in terms of network size and accuracy, and use metrics such as confusion matrix to better visualize the results. The experiment results show that ResNet and EfficientNet has much less parameters than Swin Transformer and are more computational efficient, but Swin Transformer can achieve better accuracy and robustness in this classification task.

1. Introduction

Diabetic retinopathy(DR) is one of the leading causes of vision loss nowadays. It is a common and insidious microvascular complication of diabetes and can progress to the occurrence of vision loss asymptomatically, which makes it difficult to be detected and would delay the treatment. The current detection of DR is mainly manual-based, which is time consuming and requires great experience of clinicians. On the other hand, there is a large number of diabetic patients globally but resources such as advanced and modern medical facilities are limited in many areas, which makes it more difficult for many patients to discover the illness on time. If patients miss their best treatment opportunities, the illness may eventually cause irreversible visual damage or even blindness. As a result, it would be very helpful if the illness can be detected on its early phase with more efficient and low cost methods.

This project will investigate on the recognition of progression of diabetic retinopathy to provide a diagnosis of the progression level of the disease based on input retinal images using a deep-learning based approach. With advanced computer vision technology and AI algorithms, we hope this project can help to reduce the number of patients that are at risk of undetected diabetic retinopathy in the future. For our model, the input would be one or several images of the patient's fundus photos. These images would be passed into our CNN model and output a diagnosis of level of diabetic retinopathy which would be classified to be one of the 5 levels. For the diagnosis, "0" means that the patient doesn't have a risk of diabetic retinopathy, while "1"-"4" represents the level of severity of this disease. This classification of diagonosis is based on the standard of American Academy of Ophthalmology[1].

2. Related Work

Results from recent research show that deep neural networks can be trained, using large datasets and without having to specify lesion-based features, to identify diabetic retinopathy or diabetic macular edema in retinal fundus images with high sensitivity and high specificity[2][3]. We have explored several papers in this field to get useful information on data, model and details of training and evaluating the models. The commonly used public datasets include EyePACS[4], DRD, MESSIDOR 2 and E-Ophtha[5]. Some datasets with high data volume(such as EyePACS)[6] can be used for training and validating the model, others can be used for testing (e.g. MESSIDOR 2) since it is relatively small but contains data collected from real hospitals. However, some of the above datasets are not sufficient enough for our progression classification task, since they are only labeled as "0" and "1" indicating healthy or diagnosed, without specifying the progression level. Researches also conducted various preprocessing on the datasets such as scaling images into square shape with size of 512x512 pixels[4], 299x299 pixels[2] or 224x224 pixels[7]. Other implementations include performing normalization to set image pixel values to the range of 0 and 1[4] or subtracting the mean and dividing the variance from the train image datasets.[7] Rotation augmentation can also be performed

since retina fundus have circular shape and should be invariant to rotation. To enlarge dataset by data augmentation techniques, We can randomly rotating images, crop to square and adjust the brightness of before propagating them into our model.[4][7]

In the aspect of model, most of the researchers utilized CNN to address the problem. Some papers built their own customized CNN model, for example a 6-layer CNN.[4] Some model used lesion or red lesion detection to help with classification.[5] There are also some papers proposed transfer learning methods on existing modern CNN pretrained on ImageNET beforehand[7][Automatic-8][9][10]. These predefined CNN model include VGG-16, ResNet-18, GoogLeNet, DenseNet-121 and SE-BN-Inception. The optimization functions used are various as well. Some examples including RMSProp[11] and SGD[7] yield similar but precise accuracy. Learning rate used typically ranging from 1e-1 to 1e-4.[4][5][7][11] One specific weight decay as a hyperparameter we found is 4e-5[2]. We have also learned that sensitivity, specificity, accuracy and confusion matrix are pervasive and convincing evaluation metrics that are commonly used and we would evaluate some of them on our models as well.

3. Methods

In this project, Resnet, EfficientNet, and Swin Transformer are selected as our baseline models. More detailed information about these models are provided in section 3.2, 3.3, and 3.4.

3.1. Training Methods

Given a retino image, we aim to develop a robust deep learning model that can accurately recognize the progression level. Considering CNN's outstanding performance on image classification tasks, our plan is to build a CNN model which would probably be similar to some mentioned in "Diabetic retinopathy detection through deep learning techniques: A review"[5] or "Automated Identification of Diabetic Retinopathy Using Deep Learning"[4]. Experimental results in [11] and [12] have demonstrated transfer learning could achieve better accuracy than non-transferring learning methodology on DR image classification. So, we experiment on different CNN model structures with pretrained weights as transfer learning. Hyper-parameter tuning and fine-tuning will also be performed in our next step for further improving the performance of our model. The model architectures that we experiment with are Resnet 50, EfficientNet, and Swin Transformer. The reason why We choose Resnet 50 is that it has been highlighted in many related papers and are generally suitable for various classification tasks. We choose EfficientNet because it is proved to outperform Resnet in this particular task[13]. As last, we also choose Swin Transformer which is a very modern architecture that performed well in some other classification tasks.

3.2. Resnet

Resnet 50 is a very widely used neural network architecture nowadays. It is a convolutional neural network with 50 layers[14]. We adopted pretrained weights on the Imagenet so that the model has already learned various basic features. We also used starter code to build our pretrained Resnet50 model[13]. Resnet 50 contains 48 convolutional layers, 1 max pooling layer and 1 average pool layer. The residual connection in this architecture between layers help to mitigate the accuracy saturation problem. The shortcut connection added is to perform identity mappings. The basic idea of this shortcut connection is shown in Figure 1.

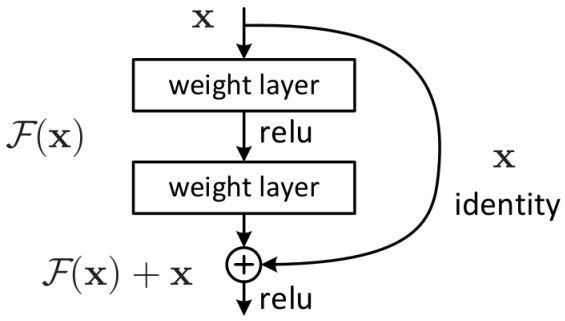


Figure 1: Basic Resnet 50 structure block

For Resnet 50, its shrotcut skips three layers and a 1*1 convolutional layer is added along the way. This architecture is widely used in many computer vision tasks including image classification, object localisation and object detection.

3.3. EfficientNet

EfficientNet is a CNN model that can achieve better performance in both accuracy and efficiency compared with previous ConvNets[15]. The core idea of EfficientNet is compound scaling method which uniformly scaling up on all depth/width/resolution dimensions with fixed scaling coefficients. The reason why we need to perform scaling on all dimensions is that, these scaling dimensions are not independent. More specifically, if the resolution of image is higher, a deeper network is desired to obtain larger receptive fields for capturing similar patterns, and a wider network to capture more fine-grained patterns. In compound scaling method, the value of depth, width, and resolution are defined as:

$$depth : d = \alpha^{\phi}$$
$$width : w = \beta^{\phi}$$

$$resolution : r = \gamma^{\phi}$$

where α, β, γ are constants obtained from a small grid search. By changing the value of ϕ and scaling up, we can obtain different versions of models in EfficientNets family.

The main building block of baseline network EfficientNet-B0 is MBConv (mobile inverted bottleneck convolution) inspired by [16], the architecture of EfficientNet-B0 is shown in Figure 2.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Figure 2: EfficientNet-B0 baseline network

In our experiment, we choose EfficientNet-B5 version as our baseline model, because it achieved a satisfying accuracy value on Imagenet dataset, with a reasonable parameter size as shown in Figure 3 from the original paper.

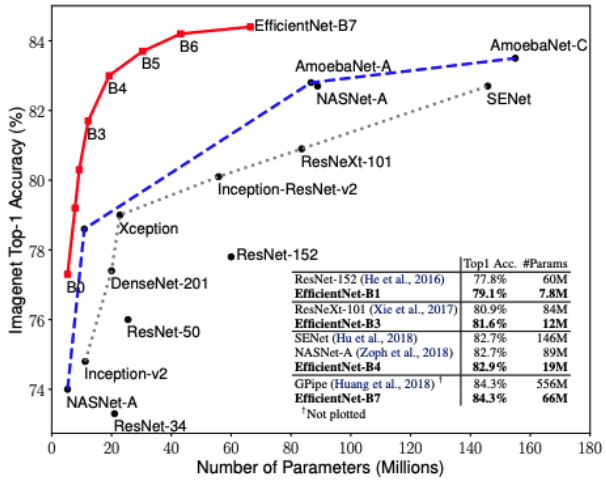
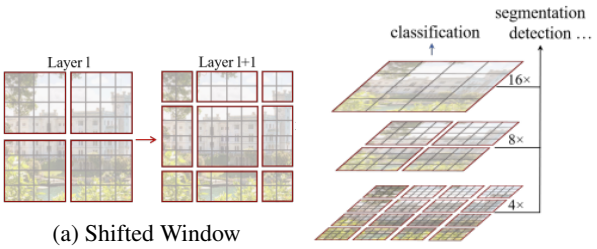


Figure 3: Comparison of different models on model size and Imagenet accuracy[15]

3.4. Swin Transformer

Recent works suggest that Vision Transformer (ViT) are more robust than CNNs in solving computer vision tasks [17], while Swin Transformer [18], introduced in 2021, is believed to outperform ViT with some more advanced features. Unlike the fixed scale word tokens that serve as the basic elements of NLP tasks, computer vision tasks elements can vary significantly in scale and resolution. Using ViT, the computational complexity is quadratic to image size. Due to this problem, the ViT model encounters difficulties when facing dense computer vision tasks. To overcome this issue, Swin Transformer proposed a modified transformer which computes self-attention locally within non-overlapping windows that partition an image, as shown in Figure 4. Through this way, the number of patches in each window is fixed, and thus the complexity increases linearly when image size rises. The second key design element of Swin Transformer is its shifted window partition between consecutive self-attention layers. As illustrated in Figure 4(a), the shifted windows in layer $l+1$ bridge the windows of the preceding layer l. These connections significantly enhance the representation power of the model. The third design element is the hierarchical feature maps, as shown in Figure 4(b). Gradually merging neighboring patches in deeper layers, Swin Transformer can conveniently leverage advanced techniques for dense prediction. All these merits make Swin Transformer suitable for our Diabetic Retinopathy Progression Recognition task.



(a) Shifted Window

(b) Hierarchical Feature Maps

Figure 4: Advanced Features of Swin Transformer [18]

The Swin Transformer model we decided to use follows the Swin-B architecture, as illustrated in Figure 5. Initially, an input 224×224 RGB image is split into patches by a patch splitting module. Following the original paper, we use a patch size of 4×4, which makes the dimension for each patch to be $4 \times 4 \times 3 = 48$ at the beginning. Then, this dimension is linearly transformed from 48 to $c = 128$ (Swin-B version). Swin Transformer blocks are applied afterwards into four stages to perform the feature transformation, where each stage contains 2, 2, 18, 2 transformer blocks respectively. The patch merging layer at the start of each stage is used to produce the hierarchical representation. For example, the merging layer in stage 2 concatenate 2×2 nearby patches together and then apply a linear transformation so that the output has shape $28 \times 28 \times 256$. As the neural network gets deeper, this procedure is repeated until the stage 4 outputs $7 \times 7 \times 1024$. Finally, the output is reduced to 1-dimension by applying a global average pooling

and linear transformed into vector scores for five classes we need, as shown in Figure 5.

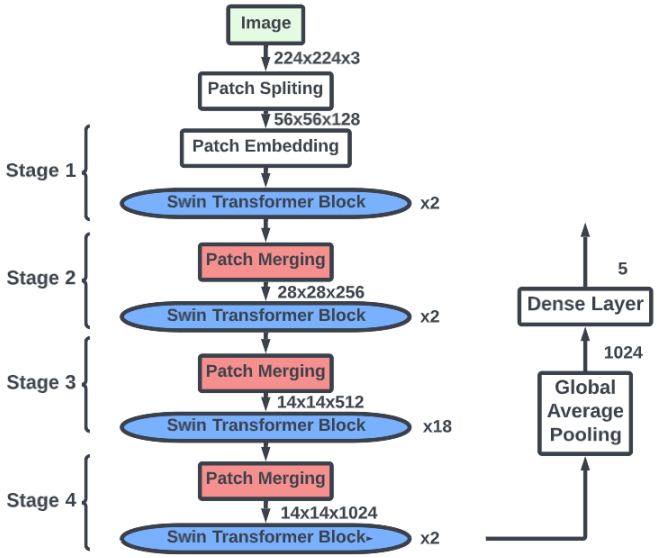


Figure 5: Architecture of Swin-B Transformer

4. Dataset and Features

We utilized the public datasets on Kaggle in our experiment instead of collecting new data by ourselves, for the reason that the current available data is adequate for our experimental use and our limit to clinical data access. The training dataset is from the Diabetic Retinopathy Detection(DRD) competition[19], and test dataset is from APTOS 2019 Blindness Detection competition[20]. The dataset classifies retinal images based on the severity level of diabetic retinopathy and there are adequate numbers of images for each level of the disease. The original dataset contains 25810 class 0 images, 2443 class 1 images, 5292 class 2 images, 873 class 3 images and 708 class 4 images. We observed that subsequently more class 0 image sources than class 3 or 4. The reason is probably that most sample tested are healthy. Such unbalanced dataset would probably result in the model only learns class 0 well, and the model may tend to classify most samples to class 0 while still maintains a "fake" high accuracy. As a result, we believe implementing data augmentation on class 2, 3 and 4 images are needed to increase their proportions. On the other hand, we need to (randomly) pick some class 0 samples to make the dataset more balanced.

To mitigate this problem, we have generated a augmented dataset and used it to train all of our models. We first randomly picked 1000 images from level 0, level 1 and level 2. Then, we performed random data augmentation which included horizontal flip, vertical flip and adjustment of brightness with coefficient 0.8 to 1.2 to these 3000 images along with the 873 level 3 images and the 708 level 4 images. We performed horizontal and vertical flip because the retino is a circle and flipping would not affect the capture of features. We adjusted the brightness randomly(a coefficient less than 1 means darken it while a coefficient larger than 1 means brighten it) because originally some of these images are brighter or darker compared to others. To make the training more efficient, we also center cropped all images to 224 * 224 pixels. As a result, we obtained a dataset of around 15000 images. Finally, we split it into training and validation set with rate 8:2. Some example images in our dataset are shown below.

Figure 6: Level 0 Example

Figure 7: Level 2 Example

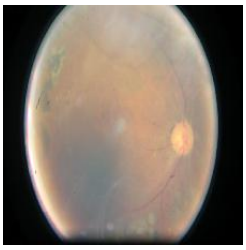


Figure 8: Level 4 Example

5. Experiments/Results/Discussion

5.1. Resnet

The hyper-parameters for the Resnet 50 model we trained is shown below.

All these hyper-parameters are chosen from results from different experiments. We started picking the hyper-

Batch Size: 8
Epoch: 30
Warmup Epoch: 2
Learning rate: 0.0001
Warmup Learning Rate: 0.001

Table 1: Hyper-parameters in Resnet 50 training

parameters by taking reference from the related works. Then, we adjusted them based on the training results and our condition. For example, due to limited access to the memory size, we can't set the batch size to be too large.

The final loss and accuracy plots we got are shown below. We have also plot the confusion matrix.

Figure 9: Loss Plot of Resnet 50 Model

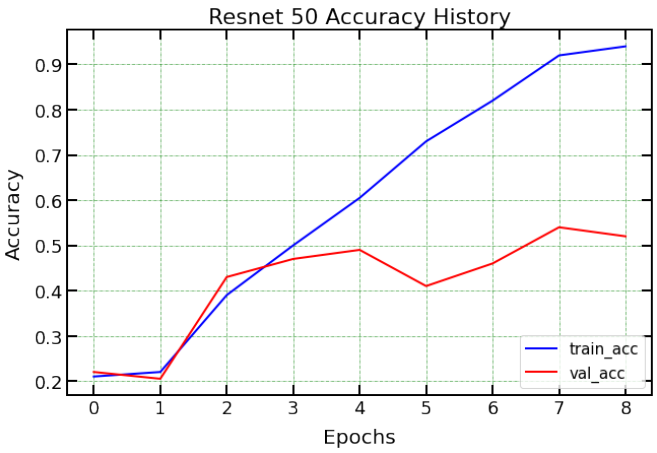


Figure 10: Accuracy Plot of Resnet 50 Model

The performance of Resnet 50 is not very satisfying. It obtains a maximum validation accuracy of about 54 percentage bafore the algorithm performing early stopping to prevent overfitting to the training data. We have also plot the diagnosis results of a small test sets (details are included in Comparison section) discovered that the model calssifies many level 1 and level 2 images to be level 0. We believe the primary reason would probably be that the difference between level 1, level 2's retino images and healthy retino images are small. It is hard for our model to capture some of the small variances and thus leading to a unsatisfying result.

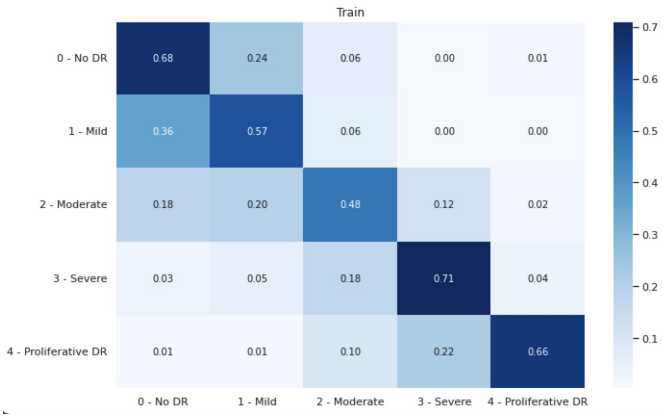


Figure 11: Confusion Matrix of Resnet 50

5.2. EfficientNet

We utilized the EfficientNet-B5 as our second baseline model, which has been proved can achieve a satisfying accuracy performance with a relative small network size[15]. We added one pooling layer, two drop-out layers after the original EfficientNet-B5 to mitigate overfitting problem, and used softmax classifier for classification[13].

The hyper-parameters used to train the EfficientNet-B5 model are shown in the Table 2:

Batch Size: 8
Epoch: 10
Warmup Epoch: 3
Learning rate: 0.0001
Warmup Learning Rate: 0.001

Table 2: Hyper-parameters in EfficientNet training

In the warmup session, we froze the top EfficientNet layers to train the last 5 classification layers. And in the training session, we fine-tuned the complete model. The loss and accuracy results are shown in Figure 12 and Figure 13.

From the loss and accuracy plots, it could be observed that EfficientNet model has a slight improvement in overall validation accuracy performance compared with Resnet, oscillating around 0.5. We also applied early-stop algorithms

Figure 12: Loss Plot of EfficientNet-B5 Model

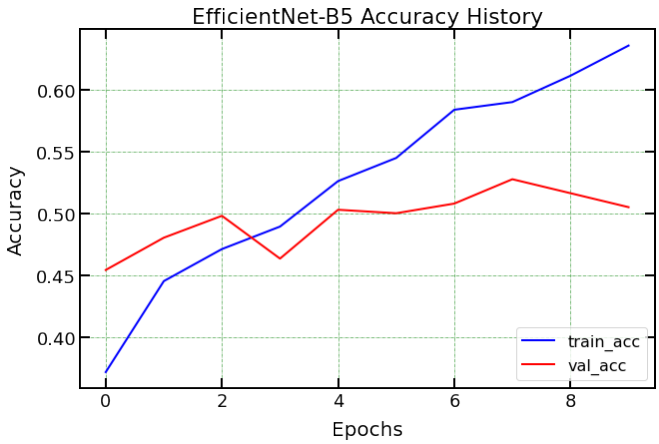


Figure 13: Accuracy Plot of EfficintNet-B5 Model

to this model's training to prevent overfitting, which makes the actual training process stop at 10th epoch. This model still cannot handle overfitting problem well, since after 6 epochs, the validation loss starts to increase, and the margin between training accuracy and validation accuracy is enlarged.

Based on the confusion matrix(Figure14), similar to Resnet model, the main errors occur among class 0, 1, and 2. Besides, EfficientNet model achieves a relative better performance in distinguishing class 3 and 4.

5.3. Swin Transformer

Here in the table below is the best combination of hyperparameters we choose after carefully tuning each of them across a wide range of values. Limited by the computation power we got, the batch size is selected to be 32. Also, we deployed an exponential learning rate schedule with decay

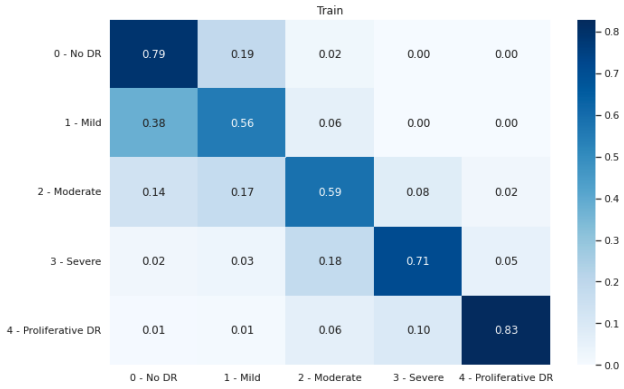


Figure 14: Confusion Matrix of EfficientNet-B5

step = 100 and decay rate = 0.95, meaning that the learning rate will be multiplied by 0.95 for every 100 steps.

Batch Size: 32
Learning Rate: 0.001
Epoch: 10
Exp Decay Steps: 100
Exp Decay Rate: 0.95

Table 3: Hyper-parameters in Swin Transformer training

Figure 15 and Figure 16 shows the loss and accuracy across the training and validation process. Compared with ResNet and EfficientNet, Swin Transformer achieves a much better validation accuracy at the final epoch. We stopped the training after 10 epochs because the validation accuracy stops increasing and stays around 0.8. Also, after 7 epochs, the validation loss starts to increase, which means that our model tends to overfit to the dataset. Therefore, stop at the 10th epoch should be a wise choice.

Figure 17 presents the confusion matrix for swin transformer when we evaluated it on validation set after the training. The accuracy distribution is kind of unbalanced, and it seems like the Swin Transformer model find it hard to distinguish between class 2, 3, and 4.

5.4. Comparison

We will compare our three models with respect to three metrics: network size(parameter numbers), test accuracy, and balanced accuracy. We tested models on 2000 samples from APTOS-2019 Blindness Detection dataset [20], whose data distribution between different classes is fairly unbalanced(around 9:2:5:1:1). The results are shown in Table 4:

Based on the data, we could observe that Resnet 50 is smallest in network size with least parameter numbers among three models, but is least robust to unbalanced data. EfficientNet-B5 model has the worst test accuracy perfor-

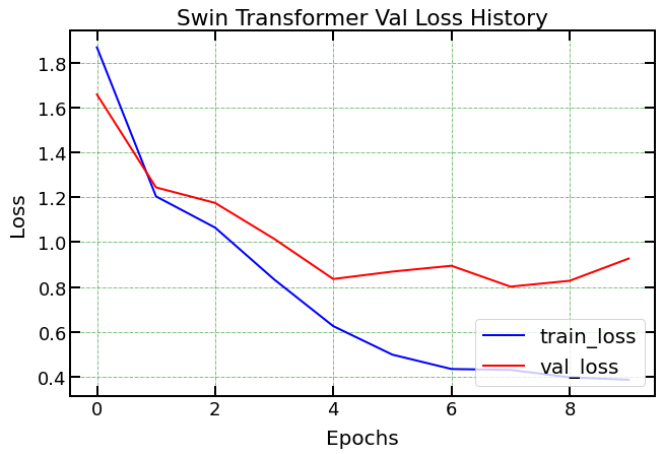


Figure 15: Loss Plot of Swin-B Transformer

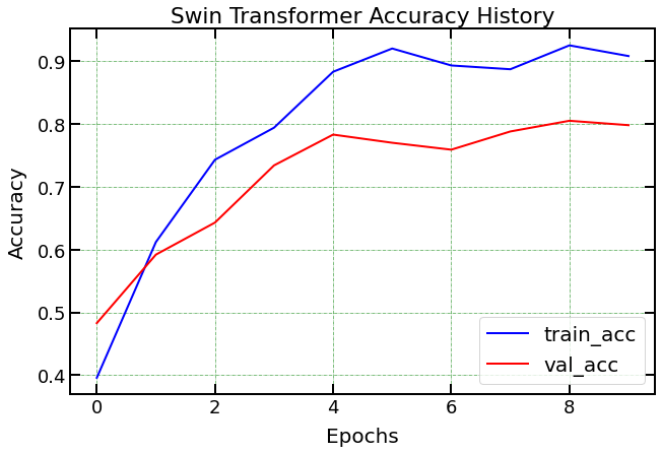


Figure 16: Accuracy Plot of Swin-B Transformer

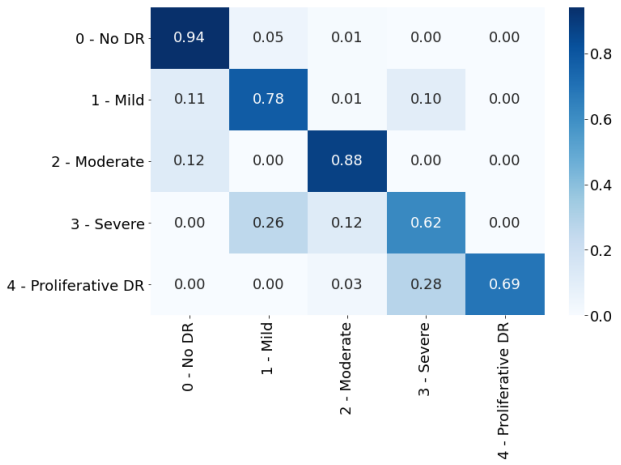


Figure 17: Confusion Matrix for Swin-B Transformer

	# param.	Test Acc	BAcc
Resnet 50	27,794,309	53.10 %	38.43 %
EfficientNet-B5	32,720,117	51.05 %	43.56 %
Swin-B Transformer	88,109,369	76.35 %	71.23 %

Table 4: Comparison

mance, but performs slightly better in balanced accuracy compared with Resnet 50. Swin Transformer has considerably more parameters and achieves highest accuracy among three models.

Even though Swin Transformer outperforms our CNN models in the test, we should aware that besides of the advantage of transformer model itself, the composition of test dataset may also contribute to the result. To be more specific, the majority of test data are from class 0, 1, 2, and Swin Transformer has relatively stronger ability in distinguishing these three classes. While the data from 3 and 4, on which Swan Transformer doesn't perform that well, has much less contributions to the results due to their small proportions in test set.

In summary, Resnet 50 is most computational efficient, followed by EfficientNet-B5, these two CNN models are less competitive with respect to accuracy compared with Swin Transformer, but transformer's better performance is at a cost of computational and time costs. There's a trade-off between accuracy and efficiency when choosing models.

6. Conclusion

Based on our experiment, Swin-B Transformer achieves the highest performance in terms of test accuracy and balanced accuracy. As described in section 5.4, swin transformer contains more parameters than ResNet and EfficientNet. More parameters generally means stronger representation power, which makes the model generalizes more easily to the training samples. Also, Swin Transformer integrates the advantages of CNNs in vision tasks with the powerful architecture of Transformer. It uses hierarchical representation to achieve scale-invariance and self-attention to model dependencies in data. Limited by time, we could only investigate the problem on three deep learning architectures. In the future, we might deploy more advanced architectures. We might also acquire more data and train the models for longer time to achieve an even higher performance.

7. Contribution

For this project, Zixin performs the data augmentation and trains the Resnet model, Yuhan trains and tests the EfficientNet model while Shunyao works on the Swin Transformer model. We work together to write the report and create the poster. This github repository provides some starter

code for building the Resnet and EfficientNet model[13].

References

[1] AmericanAcademyofOphthalmology. International Clinical Diabetic Retinopathy Disease Severity Scale Detailed Table. http://www.icoph.org/dynamic/attachments /resources/diabetic-retinopathy-detail.pdf.

[2] Voets, Mike, et al. "Reproduction Study Using Public Data of: Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs." PLOS ONE, Public Library of Science, https://journals.plos.org/plosone/article?id=10.13712-Fjournal.pone.0217541.

[3] "Detection of Diabetic Retinopathy Using Deep Learning Analysis." Retina Today, Bryn Mawr Communications, https://retinatoday.com/articles/2021-sept/detection-of-diabetic-retinopathy-using-deep-learning-analysis.

[4] T, Gargeya R;Leng. "Automated Identification of Diabetic Retinopathy Using Deep Learning." Ophthalmology, U.S. National Library of Medicine, https://pubmed.ncbi.nlm.nih.gov/28359545/.

[5] Alyoubi, Wejdan L., et al. "Diabetic Retinopathy Detection through Deep Learning Techniques: A Review." Informatics in Medicine Unlocked, Elsevier, 20 June 2020, https://www.sciencedirect.com/science/article/pii/ S2352914820302069.

[6] Bora, Ashish; "Predicting the Risk of Developing Diabetic Retinopathy Using Deep Learning." The Lancet Digital Health. https://www.thelancet.com/journals/landig/article/PIIS2589-7500(20)30250-8/fulltext.

[7] Li, Tao, et al. "Diagnostic Assessment of Deep Learning Algorithms for Diabetic Retinopathy Screening." Information Sciences, Elsevier, 5 June 2019, https://www.sciencedirect.com/science/article/pii/-S0020025519305377.

[8] "Automatic Screening of Fundus Images Using a Combination of Convolutional Neural Network and Hand-Crafted Features." IEEE Xplore, https://ieeexplore.ieee.org/document/8857073.

[9] Ayala, Angel; Figueroa, Thomas Ortiz; Fernandes, Bruno; Cruz, Francisco (2021-11). "Diabetic Retinopathy Improved Detection Using Deep Learning" (PDF). Applied Sciences. https://www.mdpi.com/2076-3417/11/24/11970

[10] Nguyen, Quang H., et al. "Diabetic Retinopathy Detection Using Deep Learning: Proceedings of the 4th International Conference on Machine Learning and Soft Computing." ACM Other Conferences, 1 Jan. 2020, https://dl.acm.org/doi/10.1145/3380688.3380709.

[11] Arcadu, Filippo, et al. "Deep Learning Algorithm Predicts Diabetic Retinopathy Progression in Individual Patients." Nature News, Nature Publishing Group, 20 Sept. 2019, https://www.nature.com/articles/s41746-019-0172-3.

[12] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017-05-24). "ImageNet classification with deep convolutional neural networks" (PDF). Communications of the ACM. 60 (6): 84–90. doi:10.1145/3065386. ISSN 0001-0782.

[13] dimitreOliveira. "Dimitreoliveira/aptos2019blindnessdetection: :3rd place medal: (Bronze Medal - 163rd Place Repository for the 'Aptos 2019 Blindness Detection' Kaggle Competition." GitHub, https://github.com/dimitreOliveira/APTOS2019 BlindnessDetection.git.

[14] Kaushik, Aakash. "Understanding Resnet50 Architecture." OpenGenus IQ: Computing Expertise amp; Legacy, OpenGenus IQ: Computing Expertise amp; Legacy, 21 July 2020, https://iq.opengenus.org/resnet50-architecture/.

[15] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." International conference on machine learning. PMLR, 2019.

[16] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V. MnasNet: Platform-aware neural architecture search for mobile. CVPR, 2019.

[17] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding Robustness of Transformers for Image Classification. CoRR. abs/2103.14586. 2021.

[18] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. International Conference on Computer Vision (ICCV), 2021.

[19] Kaggle Diabetic Retinopathy Detection competition. https://www.kaggle.com/competitions/diabetic-retinopathy-detection

[20] APTOS 2019 Blindness Detection competition. https://www.kaggle.com/competitions/aptos2019-blindness-detection/data