Diabetic Retinopathy Classification using EfficientNet Convolutional Neural Networks

Project Report for ELEC-E8739 AI in Health Technologies D

Jieming You
Aalto University
jieming.you@aalto.fi

***Abstract*—Diabetic retinopathy (DR) is a common complication of diabetes and a leading cause of blindness in working-age adults [4]. Early detection and treatment of DR could potentially prevent vision loss. In this project, we train two EfficientNet [8] Convolutional Neural Networks (CNNs) for classifying DR into five categories: normal, mild, moderate, severe, and proliferative. As a baseline, we fine-tuned an EfficientNet-B3 model on the Kaggle APTOS-19 dataset, achieving a balanced accuracy (BACC) of 0.565 on the test set. This performance was further improved through an enhanced training procedure, resulting in a BACC of 0.594. The statistical significance of the improvement was tested using bootstrapping. The results suggest that the EfficientNet-B3 model can be used for classifying DR, with potential for further enhancement through optimization of the fine-tuning process.**

I. Introduction

Diabetic retinopathy (DR) is a common complication of diabetes which affects the blood vessels in the retina, and is a leading cause of blindness in working-age adults [4]. While the early detection and treatment of DR could potentially prevent vision loss, the manual diagnosis of DR is time-consuming and might be bottle-necked by the availability of clinical staff, especially in low-resource settings. Automated methods for classifying DR from retinal images could help in the early detection of DR and improve the efficiency of screening programs. Moreover, automated methods could potentially improve the accuracy and consistency of the diagnosis, and help in the prioritization of patients for further examination.

Convolutional neural networks (CNNs) have been the state-of-the-art for image-classifying tasks. A commonly used approach is to take a pre-trained CNN model, and fine-tune it on a domain-specific dataset. Pre-training CNN models on large-scale datasets, such as ImageNet [2], have shown to improve the performance of the models on domain-specific tasks. Thus, fine-tuning allows us to use the pre-trained weights as general feature extractors for the input images, while training the last layers teaches the model to adapt to the new domain.

In this project, we compare two deep learning-based method for classifying DR into five categories: normal, mild, moderate, severe, and proliferative. The dataset is from the Kaggle APTOS-19 competition, which contains 3662 sample images of diabetic retinopathy collected from participants residing in rural India, organized by the Aravind Eye Hospital [1]. The images are labeled with the hand-labeled severity of DR on a scale of 0 to 4.

The project report is organized as follows. Section 2 describes the method used in the project, including the dataset, preprocessing steps and the selectd model architecture. Section 3 presents the experiments, training setup, and results. Section 4 contains ablation studies and concludes the report.

II. Method

A. Dataset

The Kaggle APTOS-19 dataset contains 3662 sample images of diabetic retinopathy collected from participants residing in rural India. The images are of varying sizes and resolutions, and are labeled with the hand-labeled severity of DR on a scale from 0 (normal) to 4 (proliferative).

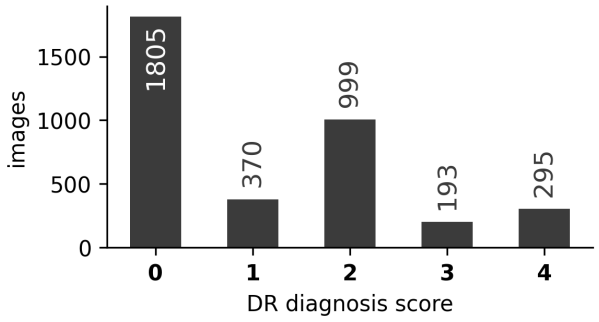

Fig. 1. Class balance of the APTOS-19 dataset. DR severity is labeled as 0 (normal), 1 (mild), 2 (moderate), 3 (severe), and 4 (proliferative).

The dataset is split between the training, validation and testing sets using the mapping provided by the original dataset. The dataset is slightly inbalanced towards the normal and moderate classes (Figure 1), and the similar distribution is seen in all splits. The training set has 2930 (80%) images, and the validation and testing sets both have 366 (10%) images each.

B. Preprocessing

The retinal images vary in resolution and aspect ratio, with some retinal images filling the whole frame, while others being cropped or being off-center.

The preprocessing steps are applied to the input images to enhance the retinal details and to standardize the input size for the model. The steps follow the procedure used in the original Kaggle competition [7] with additional steps of background

masking and data augmentation. The preprocessing steps are as follows:

1) The retina is masked using a square circular mask. The diameter of the mask is determined by the maximum distance from the center to the edge of the retina.
2) Retinal details are enhanced by subtracting a low-pass filtered image from the original image. This procedure is analogous to computing a scaled Laplacian of the original image, which highlights the regions of high frequency. Given an original retinal image I and a low-pass filtered image L, obtained using a Gaussian kernel with $\sigma = 10$, the enhanced image E is defined as:

$$E = \alpha I - \alpha L + \gamma$$

where the $\alpha = 4$ is the scaling factor and $\gamma = 128$ is the gamma correction coefficient.
3) The resulting squared image is downscaled to dimensions of 256x256 pixels.

The preprocessing steps are illustrated in Figure 2.

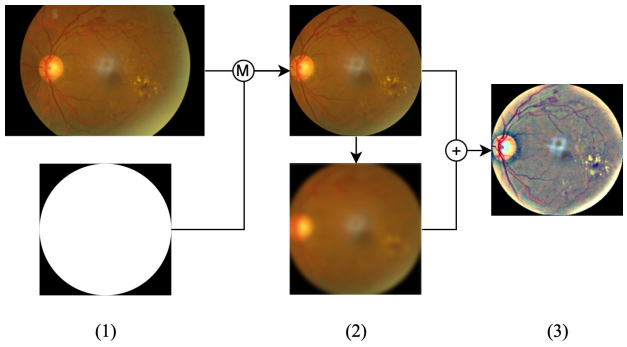

Fig. 2. The preprocessing workflow. The masked and the low-passed images at (2) are scaled with a scaling coefficient α and $-\alpha$, while the fused image is corrected by a γ coefficient.

Data augmentation is applied to the training samples to aid avoid overfitting and to improve the generalizability of the model. The following augmentations are applied to each image using the Solt data augmentation library [9]:

- Random brightness shift by $\pm 20\%$
- Random gamma correction by $\pm 20\%$
- Random contrast shift by $\pm 20\%$

Finally, the images are loaded into a PyTorch DataLoader for batched training.

C. Model

Both baseline and the improved models are based on the EfficientNet-B3 architecture which belongs to the EfficientNet model family [8]. EfficientNet is chosen as the baseline architecture since they are significantly smaller and faster at inference than the tradition ConvNet models, while achieving comparable or better performance. The EfficientNet-B3 has 12.2M parameters and is pre-trained on the ImageNet dataset.

Since the original EfficientNet-B3 model is trained on input images at resolution 300x300 and 1000 output classes, the last layer of the model is replaced with a fully connected layer with 1536 input and 5 output dimensions, corresponding to the smaller input resolution and the five classes of DR. A dropout layer of 0.5 is added before the fully connected layer to prevent overfitting.

III. Experiments

A. Training

Fine-tuning pre-trained models is generally used for domain-specific classification tasks [3], since the pre-trained weights can be used as general feature extractors for the input images, while the fine-tuning teaches the model to classify the images into the specific classes of interest. The pre-trained weights for the EfficientNet-B3 model is loaded using the TorchVision library [6] and fine-tuned on the retinal images.

The model is trained using the cross-entropy loss function, with additional weight applied to the minority classes to account for the class imbalance. The weights are calculated as the inverse of the class frequency in the training set

$$w = \frac{N^{\text{train}}}{N_c^{\text{train}}}$$

where N is the total number of samples in the training set, and N_c is the number of samples in class c.

$$\text{loss} = -\sum_c w_c \frac{\exp(x_c)}{\sum_{c'} \exp(x_{c'})} y_c$$

B. Baseline

For the baseline, the model was fine-tuned by freezing all the feature extraction layers and training only the last linear classification layer.

The model was trained using the AdamW optimizer [5] which is a variant of the Adam optimizer with decoupled weight decay. A learning rate and weight decay of 1×10^{-4}, and a batch size of 32 was used. The model was trained until convergence which took around 30 epochs.

The model achieved a balanced accuracy of 0.565 and a Cohen's Kappa score of 0.780 on the test set. The confusion matrix and the One-vs-All ROC curves are shown in Figure 3. The baseline was able to correctly classify the normal and mild classes with high accuracy, while the severe and proliferative classes were more challenging to classify.

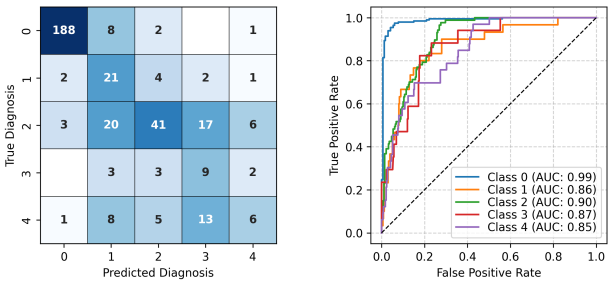

Fig. 3. Confusion matrix and One-vs-All ROC curves for the baseline model.

C. Improvement

For the improved model, we used the same architecture and training setup as the baseline, but we unfroze the feature extraction layers and trained the whole model end-to-end. This enabled the model to learn more domain-specific features from the retinal images.

Given the different complexities and training requirements of the feature extraction and classification layers, we used a different learning rate for the feature extraction layers and the classification layer. We used a smaller learning rate of 1×10^{-5} for the feature extraction layers, and 1×10^{-4} for the classification layer.

In addition, since we noticed that the model was prone to overfitting, we used a learning rate scheduler to reduce the learning rate by a factor of 0.1 after the initial 5 epochs of training which stabilized the training after the initial epochs.

The model was trained until convergence which took around 30 epochs. It achieved a balanced accuracy of 0.594 and a Cohen's Kappa score of 0.798 on the test set. The confusion matrix and the One-vs-All ROC curves are shown in Figure 4.

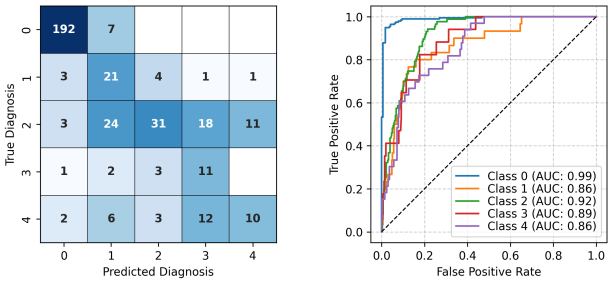

Fig. 4. Confusion matrix and One-vs-All ROC curves for the improved model.

The improved model was able to outperform the baseline model for all classes in the One-vs-All setting (Figure 4). A clear improvement from the baseline model can be seen from the normal and mild classes, while the severe and proliferative classes are still challenging to classify.

The training and validation curves for the loss are shown in Figure 5 for both the baseline and the improved model. The balanced accuracy and Cohen's Kappa scores for each epoch are shown in Figure 6.

D. Statistical testing

To test the statistical significance of the improvement in the balanced accuracy, we used bootstrapping to estimate the confidence interval of the difference in the BACC between the baseline and the improved model. 1000 samples were drawn with replacement from the test set, and the difference in the BACC was calculated for each sample. The 95% confidence interval of the difference was calculated from the bootstrapped distribution.

One-sample T-Test was used to verify the significance of the difference. The resulting p-value suggests that the improvement in the BACC is statistically significant and thus the null

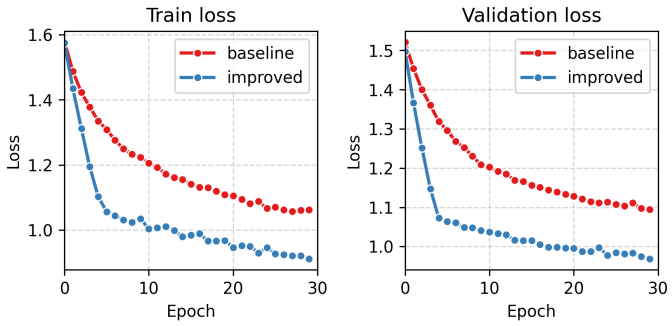

Fig. 5. Training and validation loss curves for the baseline and the improved model.

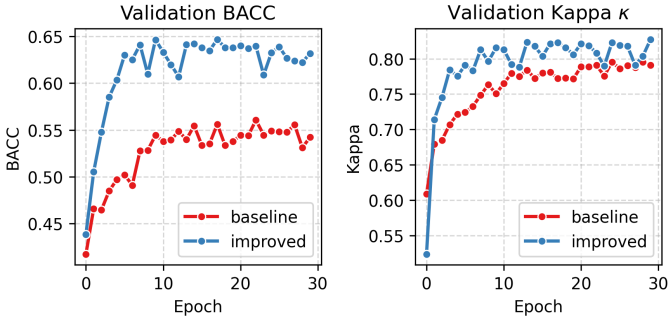

Fig. 6. Balanced accuracy and Cohen's Kappa scores for the baseline and the improved model.

hypothesis can be rejected. The bootstrapped distribution is shown in Figure 7.

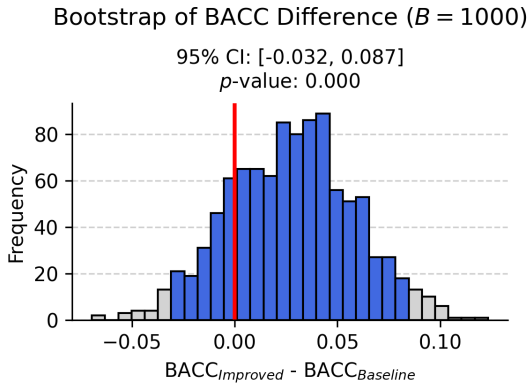

Fig. 7. Bootstrapped distribution of the difference in the balanced accuracy between the baseline and the improved model. The shaded are represents the 95% confidence interval.

IV. Ablation Study

Preprocessing playes a crucial role in the performance of the model. To investigate the effect of the preprocessing steps on the model performance, we trained a model without the preprocessing steps, and a model without the data augmentation using the training procedure adopted by the improved model. The results are shown in Table I.

TABLE I
ABLATION STUDY RESULTS FOR THE PREPROCESSING STEPS.

Model	BACC	Kappa
Baseline (improved model)	0.594	0.798
No augmentation	**0.601**	**0.814**
No preprocessing	0.580	0.801

Interstingly, the model without data augmentation performed slightly better than the improved model, while the model without preprocessing performed worse. Since data augmentation is used to prevent overfitting, the loss in accuracy with data augmentation might be a trade-off between the generalizability of the model. However, this hypothesis needs further investigation.

The model without preprocessing performed worse than the improved model, suggesting that the preprocessing steps are important for the model to learn the domain-specific features from the retinal images.

V. CONCLUSION

In this project, we trained two EfficientNet Convolutional Neural Networks for classifying diabetic retinopathy into five categories: normal, mild, moderate, severe, and proliferative. The baseline model achieved a balanced accuracy of 0.565, while the improved model achieved a balanced accuracy of 0.594.

While the model is not sufficient for clinical use, it was able to classify the normal and mild classes with reasonable accuracy. However, the severe and proliferative classes were more challenging to classify. There is potential for further enhancement through image pre-processing and optimization of the fine-tuning process.

REFERENCES

[1] *APTOS 2019 Blindness Detection Dataset*. Accessed: 2024-11-29. 2019. URL: https://www.kaggle.com/datasets/mariaherrerot/aptos2019/data.

[2] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[3] Hee E. Kim et al. "Transfer learning for medical image classification: a literature review". In: *BMC Medical Imaging* 22.1 (Apr. 2022). ISSN: 1471-2342. DOI: 10.1186/s12880-022-00793-7. URL: http://dx.doi.org/10.1186/s12880-022-00793-7.

[4] Martina Kropp et al. "Diabetic retinopathy as the leading cause of blindness and early predictor of cascading complications—risks and mitigation". In: *EPMA Journal* 14.1 (Feb. 2023), 21–42. ISSN: 1878-5085. DOI: 10.1007/s13167-023-00314-8. URL: http://dx.doi.org/10.1007/s13167-023-00314-8.

[5] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG]. URL: https://arxiv.org/abs/1711.05101.

[6] TorchVision maintainers and contributors. *TorchVision: PyTorch's Computer Vision library*. https://github.com/pytorch/vision. 2016.

[7] Duc Nguyen. *APTOS: Eye Preprocessing in Diabetic Retinopathy*. https://www.kaggle.com/code/ratthachat/aptos-eye-preprocessing-in-diabetic-retinopathy. Accessed: 2024-12-16.

[8] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *ArXiv* abs/1905.11946 (2019).

[9] Aleksei Tiulpin. *SOLT: Streaming over Lightweight Transformations*. Version v0.1.9. July 2019. DOI: 10.5281/zenodo.3702819. URL: https://doi.org/10.5281/zenodo.3702819.