Diabetic Retinopathy Classification Preprocessing Evaluation

Jordan Becci
beccijor@gmail.com

Rylee Albrecht
ryleemalbrecht@gmail.com

Abstract

Diabetic retinopathy (DR), caused by elevated blood sugar levels damaging retinal blood vessels, is a leading cause of blindness among working-age adults. This project explores the impact of different preprocessing techniques on the classification of DR severity using deep learning and computer vision methods. We fine-tune three distinct models multiple times, each with a different preprocessing approach applied to the same dataset, aiming to identify the optimal preprocessing technique across all models. Our findings suggest that while some preprocessing methods enhance performance for two models, there is no single preprocessing method that consistently improves results across all models tested.

1 Introduction

Diabetic retinopathy (DR) is the leading cause of blindness among working-age adults. It occurs when elevated blood sugar levels damage the blood vessels in the retina, causing them to swell and leak, which can lead to blurred vision [3]. By leveraging deep learning and computer vision techniques, automated DR classification and lesion localization can improve screening efficiency, provide explainable AI-driven lesion detection, reduce subjectivity in diagnosis, and ultimately improve patient outcomes.

Numerous preprocessing techniques have been applied to retinal images before training models for DR classification, but there is still no clear consensus on the most effective approach. Our goal was to identify the most effective preprocessing technique for retinal fundus images to enhance future diabetic retinopathy classification models.

2 Related Work

Recent advancements in automated diabetic retinopathy detection and classification include the use of convolutional neural networks (CNNs), transfer learning, Vision Transformers, and hybrid models. Additionally, various preprocessing methods have been developed and explored to help guide model attention toward important lesion types such as microaneurysms, hemorrhages, and exudates.

2.1 Deep Learning Models for DR Classification

CNNs have become the backbone of DR classification and detection architectures have shown significant performance increases, due to their residual learning framework that both enables deep network training and mitigates vanishing gradient issues. Jiwane et al. [6] used ResNet50 and transfer learning to achieve high accuracy in DR classification. ResNet has proved to be a used architecture for DR classification, as it is able to extract robust features while maintaining computational efficiency. EfficientNet and its variants (EfficientNetB0, etc) are widely used in DR classification because they are able to train with fewer parameters and still achieve high accuracy. Its efficiency would be ideal for clinical settings which have limited resources and computational power. DenseNet has also been used for DR classification tasks due to its ability to capture fine-grained features in retinal images.

In a study conducted by Akhtar and Aftab [5], DenseNet's accuracy on the APTOS dataset was competitive with that of other CNNs, especially when combined with transfer learning techniques. Additionally the use of hybrid models that utilize CNN backbones and transformers (ViTs), has seen significant improvements in DR classification accuracy and efficiency [9]. Hybrid models guide model attention to smaller, less-noticeable yet critical lesions.

Jordan Becci
beccijor@gmail.com

Rylee Albrecht
ryleemalbrecht@gmail.com

2.2 Preprocessing

Some of the preprocessing methods explored include contrast-limited adaptive histogram equalization (CLAHE) with a Gaussian filter [7], histogram equalization with a median filter [8], CLAHE applied to the green channel followed by a median filter [4], and Gaussian subtractive normalization introduced by Benjamin Graham, the winner of the EyePACS Kaggle competition [1]. These methods will be described in more detail in the Method section.

3 Dataset

For this project, we utilized two publicly available datasets: the APTOS 2019 Blindness Detection dataset [2] and the Diabetic Retinopathy Detection dataset (EyePACS) [1]. The APTOS 2019 dataset consists of 5,590 retinal fundus images labeled across five severity levels of DR, 0-4 (No DR to Proliferative DR).. The EyePACS dataset contains 35,126 images of retinal fundus with similar severity classifications. These datasets will serve as the foundation for training and evaluating our models to assess the effectiveness of our preprocessing methods.

For our experiments, we combined both datasets to create a more robust dataset. To isolate the impact of preprocessing methods, we balanced the classes by undersampling the majority class and oversampling the minority classes. This ensured that any differences in model performance were not influenced by class imbalance. As a result, each class contained 5,000 images.

4 Method

To effectively assess whether a particular preprocessing method outperforms others for diabetic retinopathy (DR) classification, we fine-tuned three different pretrained models, each trained five times using five distinct preprocessing techniques. A method that consistently achieves the best performance across all models would be a strong candidate for the most effective approach.

4.1 Models

The three models we tested our preprocessing methods on were ResNet-50, EfficientNet, and a hybrid DenseSwin model.

4.1.1 ResNet-50

The first model we trained on was the ResNet50 CNN backbone that was pre-trained on ImageNet and fine-tuned for DR classification. ResNet50 is a 50 layer CNN with residual connections that allow for robust feature extraction. The model is structured in stages in which there is the initial convolutional layer, max pooling, and four stages of residual blocks. The final stage includes global average pooling and a fully connected layer for classification.

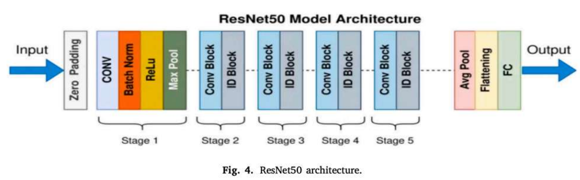

Figure 1: ResNet50 Architecture

4.1.2 EfficientNet

We used EfficientNet pre-trained on ImageNet, and fine tuned for DR classification tasks. 224x224 pixels RGB images are accepted by EfficientNet and move through a series of bottleneck blocks with squeeze-and-excitation modules. Next, it is followed by a layer of global average pooling and a fully connected layer. In fine-tuning, beginning layers were replaced with a new fully connected layer to predict the DR stages.

4.1.3 DenseSwin

The last model we fine-tuned was a hybrid model combining DenseNet-121 and shifted window (Swin) transformer.

Diabetic Retinopathy Classification Preprocessing Evaluation

Jordan Becci
beccijor@gmail.com

Rylee Albrecht
ryleemalbrecht@gmail.com

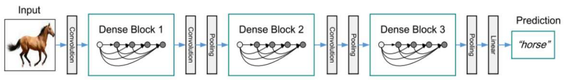

Figure 2: DenseNet-121 Architecture

The pretrained DenseNet-121 was used as the backbone of the model. As shown in Figure 2, the architecture consists of dense blocks alternating with convolutional and pooling layers. Within each dense block, every convolutional layer receives feature maps from all preceding layers, enabling efficient feature reuse and improved gradient flow.

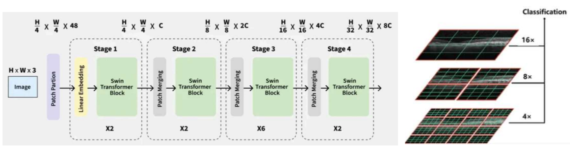

Figure 3: Swin Transformer Architecture

The Swin Transformer was used for fine-tuning. As illustrated in Figure 3, the input image is first divided into non-overlapping 4×4 patches, which are then embedded into vectors. The architecture consists of alternating Swin Transformer blocks and patch merging layers. Each Swin Transformer block updates the patch embeddings using shifted window-based self-attention and multilayer perceptrons (MLPs). Patch merging layers progressively combine 2×2 neighboring patches, building a hierarchical representation that captures local features in early layers and global features in deeper layers.

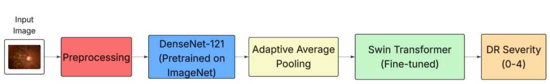

Figure 4: Hybrid DenseSwin Architecture

Together, shown in figure 4, these components form the hybrid DenseSwin architecture. The input image is first preprocessed and passed through DenseNet to extract local feature representations. These features are then reshaped via an adaptive average pooling layer to match the input format required by the Swin Transformer. The fine-tuned Swin Transformer then processes the features, capturing both local and global context through its hierarchical attention mechanism. This combination leverages the strength of DenseNet in capturing fine-grained local patterns and the Swin Transformer's ability to model broader contextual relationships for final prediction.

4.2 Preprocessing Methods

Each of the preprocessing methods we used was inspired by the related works described earlier because their models demonstrated strong performance. We chose to compare these methods to determine whether one consistently outperforms the others across multiple models, with the goal of providing a way to further enhance model performance beyond what has been previously achieved. An example of each preprocessing method applied to the same image is shown below in figure 5.

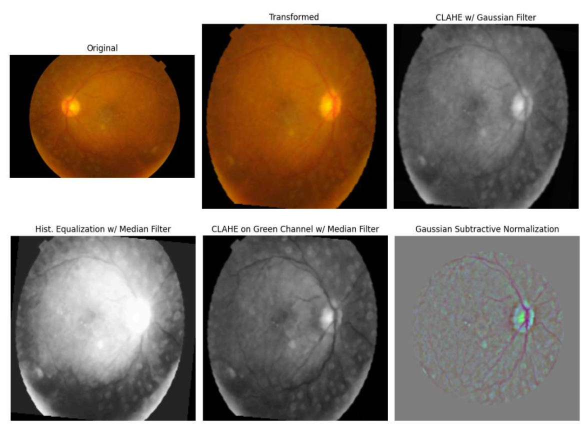

Figure 5: Example of all Preprocessing Methods

4.2.1 Regular

Our regular preprocessing is how we transformed the dataset before any other preprocessing, serving as the baseline. It involved resizing the images to 224x224 pixels, converted to RGB and normalized [0, 1]. Additionally, the training images underwent data augmentation including horizontal flips, rotations, and random resized crops; and the validation images were center cropped.

Jordan Becci **Rylee Albrecht**
beccijor@gmail.com ryleemalbrecht@gmail.com

4.2.2 CLAHE with Gaussian Filter

The goal of this method is to enhance local contrast with CLAHE and reduce noise while preserving edges with gaussian blur. First, images were converted to grayscale. Gaussian blur was implemented in a 3x3 kernel and zero sigma was applied to smooth the image. The processed images were converted back to RGB to match the model's input.

4.2.3 Histogram Equalization with Median Filter

The goal of this method is to improve local contrast with CLAHE and reduce noise with a median filter. The images were converted to grayscale, and then histogram equalization was applied to redistribute the pixel intensities. A 3x3 kernel median filter was applied, smoothie noise but keeping the edges. Finally, the processed grayscale image was converted back into RGB to match the model's

4.2.4 CLAHE on Green Channel with Median Filter

This method involves several steps. First, the image undergoes intensity conversion by selecting the high-contrast green channel and converting it into a grayscale image. Next, noise is reduced using a median filter to suppress isolated noise while preserving edges. Finally, contrast enhancement is performed using CLAHE, which provides efficient contrast enhancement without amplifying noise.

4.2.5 Gaussian Subtractive Normalization

This method first estimates the image's radius based on the middle row's intensity and resizes the image accordingly. Second, it subtracts the local average color using a gaussian blur and enhances contrast with weighted addition. Last, it applies a circular mask to preserve the central region and darken the outer areas.

5 Results

The performance of three deep learning models (ResNet, EfficientNet, and HybridDenseSwin) was evaluated across five preprocessing techniques for DR classification. We assessed accuracy, precision, recall, and F1 score summarized in Table 1. Regular preprocessing with data augmentations such as crop, flips, and rotations served as the baseline. The preprocessing technique CLAHE with the Gaussian filter achieved the highest accuracy for both EfficientNet (0.6961) and ResNet50 (0.7580).

As for ResNet50's results, CLAHE with the Gaussian filter was highest in all metrics. Gaussian Subtractive Normalization produced the lowest accuracy despite having the highest precision.

EfficientNet also performed best in all metrics, except precision, using CLAHE with Gaussian Filter. The baseline method had the highest precision. Histogram Equalization with Median Filter performed significantly worse than the other techniques.

The Hybrid DenseSwin model encountered issues close to the project deadline, preventing it from being trained on all preprocessing methods. However, among the methods it was tested on, it achieved the highest overall performance using only the regular preprocessing method. When looking at the other two models results, they both showed a decline in performance when trained with histogram equalization combined with a median filter and with Gaussian subtractive normalization. Since DenseSwin was not trained on these specific methods, we cannot definitively assess its performance under those conditions. However, given that DenseSwin appears to respond differently to input images compared to the other models, it's possible that it might have handled these preprocessing methods more effectively.

Diabetic Retinopathy Classification Preprocessing Evaluation

Jordan Becci
beccijor@gmail.com

Rylee Albrecht
ryleemalbrecht@gmail.com

Model	Metrics	Preprocessing Method				
		Regular	CLAHE w/ Gaussian Filter	Histogram Equalization w/ Median Filter	CLAHE on Green Channel w/ Median Filter	Gaussian Subtractive Normalization
ResNet-50	Accuracy	0.6709	0.7580	0.6767	0.7467	0.6184
	Precision	0.7503	0.6729	0.6155	0.7407	0.7567
	Recall	0.6709	0.7580	0.6767	0.7467	0.6184
	F1 Score	0.7033	0.7317	0.6344	0.7432	0.6698
EfficientNet	Accuracy	0.6762	0.6961	0.4840	0.6780	0.6526
	Precision	0.7501	0.7312	0.5480	0.6572	0.7335
	Recall	0.6762	0.6961	0.4840	0.6780	0.6526
	F1 Score	0.7036	0.7317	0.5070	0.6611	0.6869
Hybrid DenseSwin	Accuracy	0.7328	0.7068	-	0.6972	-
	Precision	0.7380	0.7243	-	0.7381	-
	Recall	0.7328	0.7068	-	0.6972	-
	F1 Score	0.7336	0.7121	-	0.7144	-

Table 1: Metrics for all Preprocessing Methods using ResNet-50, EfficientNet, and Hybrid DenseSwin.

Diabetic Retinopathy Classification Preprocessing Evaluation

Jordan Becci
beccijor@gmail.com

Rylee Albrecht
ryleemalbrecht@gmail.com

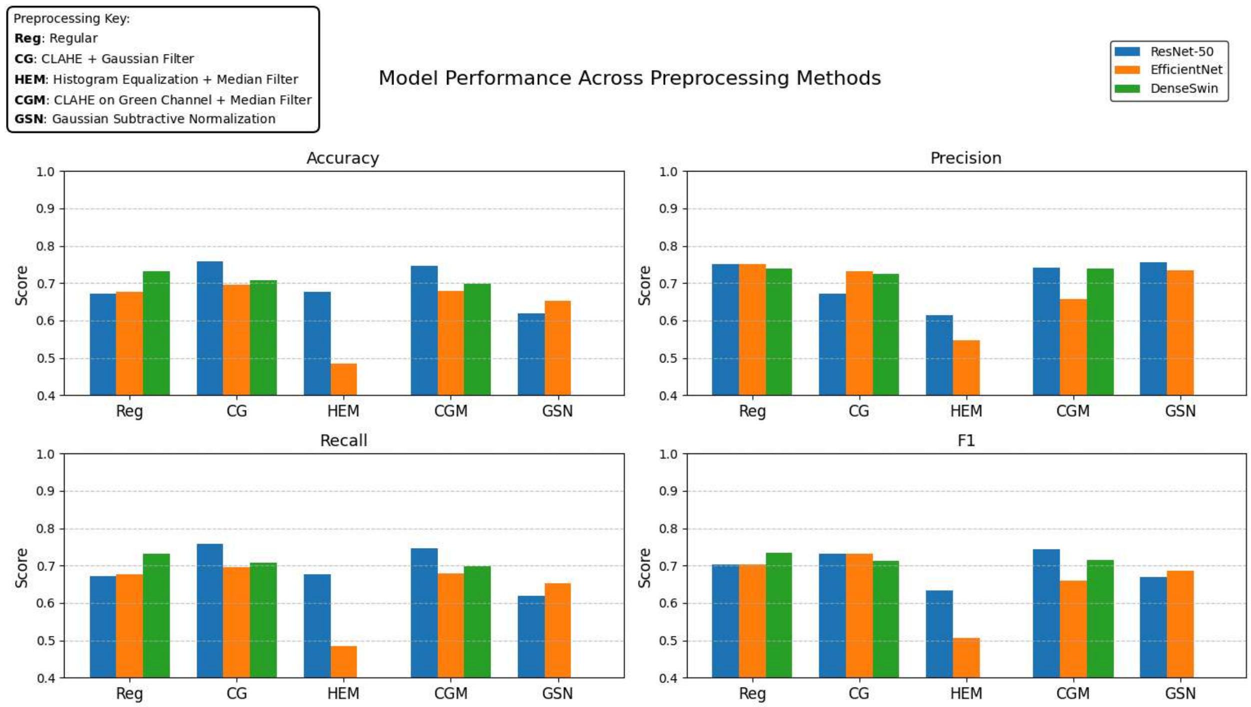

Figure 6: Performance comparison of ResNet-50, EfficientNet, and DenseSwin across five preprocessing techniques.

6 Conclusion

In this study, we evaluated the impact of five different preprocessing methods on diabetic retinopathy (DR) classification using three deep learning models–ResNet-50, EfficientNet, and a hybrid DenseSwin model. The objective was to identify the most effective preprocessing approach for enhancing model performance across different architectures. Our results revealed that certain preprocessing methods, such as CLAHE with Gaussian filter and CLAHE on the green channel with median filter improved performance for specific models, there was no single method that consistently outperformed others across all models. This suggests that the effectiveness of preprocessing techniques is model-dependent, and further research is needed to refine preprocessing strategies for optimal DR classification. The findings highlight the complexity of preprocessing choices and encourage a more tailored approach to model training based on specific architectural strengths.

6.1 Limitations

Due to computational restraints, Hybrid DenseSwin could not be tested on all 5 techniques (including regular preprocessing).

7 References

1. Emma Dugas, Jared, Jorge, and Will Cukierski. Diabetic Retinopathy Detection. https://kaggle.com/competitions/diabetic-retinopathy-detection, 2015. Kaggle.
2. Karthik, Maggie, and Sohier Dane. APTOS 2019 Blindness Detection. https://kaggle.com/competitions/aptos2019-blindness-detection, 2019. Kaggle.
3. Centers for Disease Control and Prevention, “Diabetes and Vision Loss,” U.S. Department of Health & Human Services. [Online]. Available: https://www.cdc.gov/diabetes/diabetes-c

Jordan Becci
beccijor@gmail.com
Rylee Albrecht
ryleemalbrecht@gmail.com

omplications/diabetes-and-vision-loss.html.

4. R. Navaneethan, H. Devarajan, Enhancing diabetic retinopathy detection through preprocessing and feature extraction with MGA-CSG algorithm, Expert. Syst. Appl 249 (2024) 123418.
5. Akhtar, S., Aftab, S. (2024). A Framework for Diabetic Retinopathy Detection using Transfer Learning and Data Fusion. *International Journal of Information Technology and Computer Science*, 16(6), 61–73.
6. Jiwane, S., Shinde, A., Gulhane, A., Pathak, P., Patel, A. (2022). Diabetic Retinopathy Detection using Deep Learning. *Journal of Medical Imaging*, 9(2), 024501.
7. Ratanapakorn, T., Daengphoonphol, A., Eua-Anant, N., & Yospaiboon, Y. (2019). Digital image processing software for diagnosing diabetic retinopathy from fundus photograph. *Clinical Ophthalmology*, 13, 641–648.
8. Qureshi, I., Ma, J., & Shaheed, K. (2022). Diabetic retinopathy detection and multi-stage classification using deep learning models: A quick review. *International Journal of Advanced Computer Science and Applications*, 13(6).
9. Huang, Y., Li, Z., Zhang, X., Wang, H. (2021). A Vision Transformer-Based Framework for Diabetic Retinopathy Grading. *arXiv preprint arXiv:2103.12345*