CHAPTER IV
RESULTS AND DISCUSSION

4.1 Results of Image Screening

4.1.1 Optical disc and macula detection

In Table 4.1, the performance of the proposed method reveals that the Euclidean distance (ED) error for the optic disc and macula is 19.9 and 20.3 pixels, respectively. When expressed as a percentage of the image width, these errors are approximately 1.55% and 1.59%, respectively. Additionally, we evaluate the proposed method using various average precision (AP) scores, including AP at an IOU threshold of 0.50 (AP_{50}), AP at an IOU threshold of 0.75 (AP_{75}), and the mean AP score between IOU thresholds of 0.50 and 0.95 with a step size of 0.05. The results indicate that macula detection is highly sensitive to the IOU threshold, as evidenced by the AP score dropping from 0.847 to 0.650. Conversely, optic disc detection shows less sensitivity when comparing AP_{50} with AP, indicating that the template matching method for optic disc detection is relatively stable. Moreover, we calculate the mAP to serve as the representative AP score for the proposed method. The mAP is widely used to compare the performance of object detection algorithms, providing a comprehensive measure of accuracy across different IOU thresholds. Generally, to compare the proposed method with other approaches, the Messidor dataset is used to measure the performance of macula detection algorithms. This measurement is conducted by comparing the predicted macula location with the ground truth location across various sizes of agreement areas and then computing the score, called the R-criterion score, where R represents the radius of the optic disc, being roughly 208 pixels each image. As shown in Figure 4.1, an increase in the denominator leads to a reduction in the size of the agreement area. Thus, the R-criterion effectively demonstrates the accuracy, precision, and stability of the algorithm in predicting the macula's location as the agreement area decreases.

Table 4.2 presents the outcomes of the proposed method across different agreement areas. It is notable that the values for R and R/2 are relatively close, indicating that over 90 percent of the predicted macula locations fall within the radius of R/2, which equals 104 pixels, around the ground-truth macula location. Similarly, the

Table 4.1 The general performance of proposed method on IDRiD dataset.

Ocular name	ED (pixels)	AP_{50}	AP_{75}	AP	mAP	runtime (s)
Optic disc	19.9	0.938	0.840	0.740	0.659	0.102
Macula	20.3	0.847	0.650	0.578		

predicted macula is distributed around the ground-truth location within the radius of 52 pixels at 84.2 percent and within 26 pixels at 38.7 percent. This demonstrates the proposed method's capability to accurately predict the macula's location with varying levels of precision.

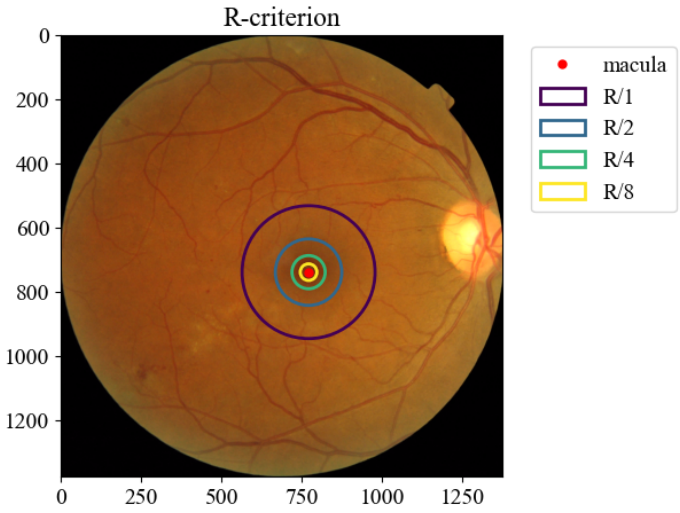

Figure 4.1 The agreement area of each R-criterion on Messidor dataset.

Table 4.2 The macula detection on Messidor dataset.

Ocular name	R/8	R/4	R/2	R
Macula	0.387	0.842	0.916	0.922

The qualitative results of the proposed method are illustrated in Figure 4.2 and Figure 4.3. Sub-figures (a-c) demonstrate high-quality predictions, whereas sub-figures (d-f) show poor-quality predictions. Notably, the IDRiD dataset results highlight frequent false detection of the optic disc, frequently caused by bright lesions and white fibers that obscure the optic disc's location. Additionally, false macula detection is

influenced by factors such as medium-sized hemorrhages, dark spots, and uneven illumination in the retinal image because these dark-like regions can resemble the macula on grayscale images during the matching process. Furthermore, erroneous prediction in optic disc detection can adversely impact macula detection, as the predicted location of the optic disc is used as a reference to delineate the ROI for macula detection. Furthermore, in the Messidor dataset, the proposed method frequently struggles to detect the macula due to the challenges in distinguishing blood vessels from the macula. Eventually, these qualitative results demonstrate the effectiveness of the method under optimal conditions and simultaneously expose its vulnerability to image noise and pathological obstructions.

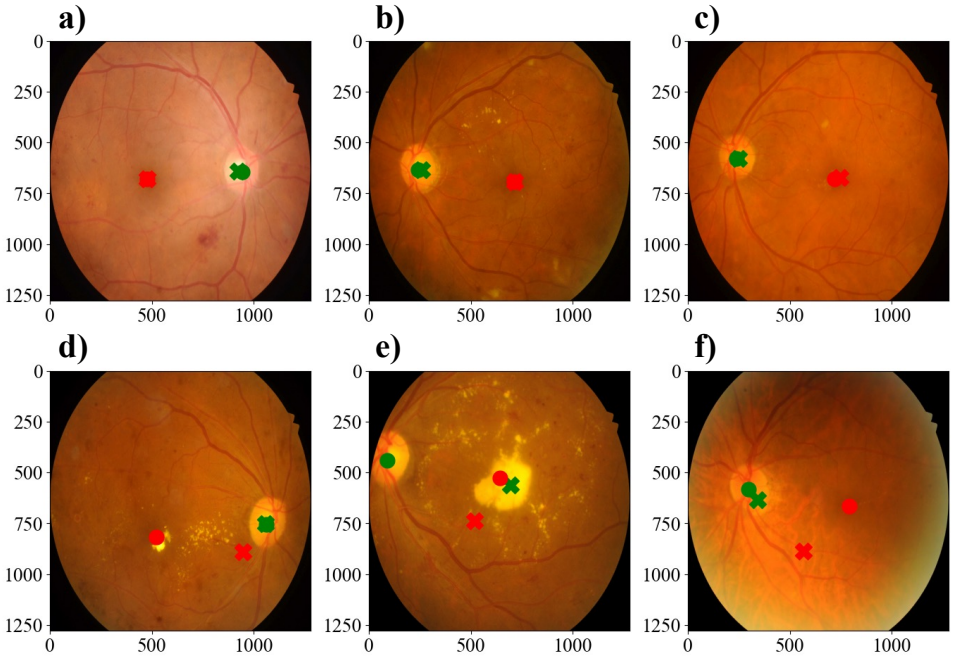

Figure 4.2 The detection results of the proposed method on the IDRiD dataset. The three images above (a–c) showcase good quality predictions, whereas the images below (d–f) exhibit poor quality predictions. Where, cross sign (✕) is a predicted location, the dot sign (●) is a ground-truth location, the green color represents the optic disc, and the red color represents the macula.

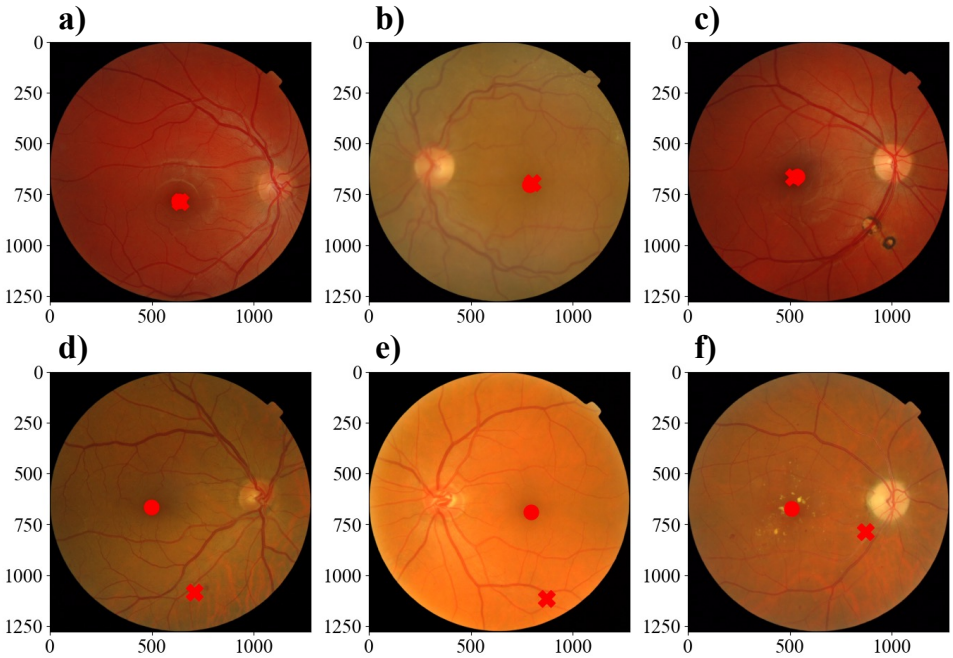

Figure 4.3 The detection results of the proposed method on the Messidor dataset. The three images above (a–c) showcase good quality predictions, whereas the images below (d–f) exhibit poor quality predictions. Where, cross sign (✕) is a predicted location, the dot sign (●) is a ground-truth location, and the red color represents the macula.

4.1.2 Screening algorithm

The performance of the proposed method is demonstrated through the confusion matrix presented in Figure 4.4. This matrix illustrates that the algorithm exhibits superior ability in correctly classifying positive images as opposed to negative images. This is evidenced by a lower incidence of false positives compared to false negatives. This discrepancy suggests that the features employed by the algorithm might not be sufficiently robust for accurately distinguishing negative images. Table 4.3 provides further insight, indicating the high precision score and reliable false discovery rate of the proposed method, which closely align with the goal score of 0.05. Moreover, the method achieves a high recall of 0.906, which can tackle the established goal. In practical terms, the proposed method successfully reduces the proportion of negative images in the dataset to roughly 7 percent while retaining 90 percent of the positive images after screening.

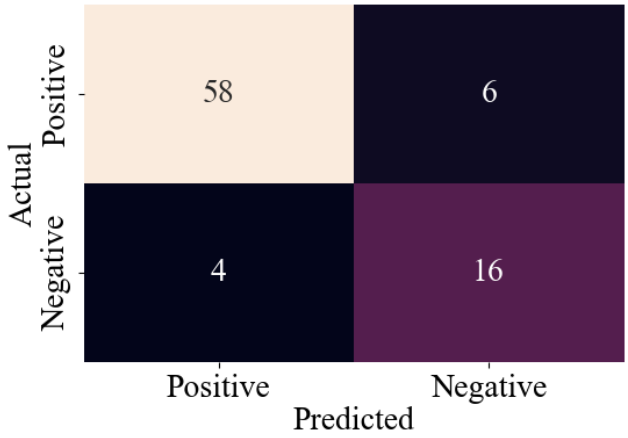

Figure 4.4 The confusion matrix of proposed image screening method.

Table 4.3 The performace of proposed image screening

	Accuracy	Precision	Recall	FDR
Goal Coyner et al., 2018; Fleming et al., 2006	-	0.950	0.900	0.050
Proposed method	0.881	0.935	0.906	0.065

The qualitative results of the proposed method, as illustrated in Figure 4.5, reveal distinct characteristics associated with each type of prediction. Notably, false-positive cases frequently occur from false detections where lesions or haemorrhages are incorrectly identified as resembling the macula. Similarly, false negatives result from misdetections, as depicted in Figure 4.5.e. Conversely, the case shown in Figure 4.5.f indicates a scenario where image overcropping causes the macula to be located outside the acceptable region, leading to false negative predictions. These observations suggest that enhancing the stability of the retinal fundus field could potentially improve the accuracy of the screening process. In the true positive case, the proposed method demonstrates the ability to correctly classify an image as positive, even in the presence of abnormal retinal conditions. Additionally, images of the nasal field are accurately identified as true negatives, further validating the method's effectiveness.

In the further study of the ML model for the screening task, as presented in Table 4.4, most models demonstrate high performance, exhibiting reliable precision, recall, and false discovery rate (FDR) scores. While no single model consistently outperforms others across all metrics, certain models exhibit optimal performance for specific objectives. Based on this dataset, the Histogram Gradient Boosting (HGB) model

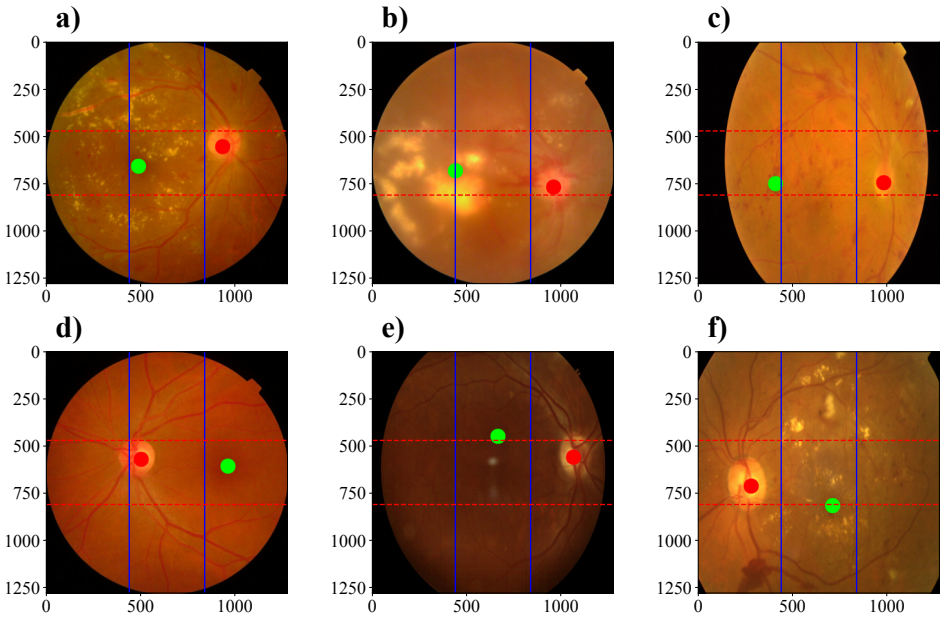

Figure 4.5 Example of image screening results: (a) True positive; (b-c) False positive; (d) True negative; and (e-f) False negative.

is most effective for identifying negative images, achieving a precision of 0.908 and an FDR of 0.092, reflecting strong screening capability. In contrast, DecisionTree and LightGBM models demonstrate the highest recall scores of 0.938, making them suitable for preserving positive images. Ultimately, if a single model were to be selected for real-world deployment, the HGB model would be preferred due to its superior precision and sufficiently high recall, which exceeds the target threshold of 0.900.

Table 4.4 The performace of machine learning in image screening. **Bold** represents the best score and underline represents the second best score.

Model name	Accuracy	Precision	Recall	FDR
Logistic Regression	0.845	0.905	0.891	0.095
Decision Tree	0.857	0.882	0.938	0.118
SVC	0.857	0.871	**0.953**	0.129
Random Forest	0.857	0.906	0.906	0.094
HistGradientBoosting	**0.869**	**0.908**	0.922	**0.092**
XGBoost	0.833	0.868	0.922	0.132
LightGBM	**0.869**	0.896	0.938	0.104

4.2 Ablation Study of Image Screening

In this section, we empirically investigate the impact of the components in the algorithm. The metric considered for this entire study is an AP score at an IOU threshold of 0.50 due to ease and fairness.

4.2.1 Template size and sampling amount

First, we investigate the influence of optic disc template size and the number of sampled templates using a grid search strategy. Template sizes are varied from 200 to 400 pixels, incorporating both symmetric and asymmetric configurations, while the number of samples ranged from 5 to 50 in increments of 5. Higher performance is represented by brighter colors on the corresponding color bar. As depicted in Figure 4.6, the optimal template dimensions lie within the range of 250–350 pixels in width (x-axis) and 300–400 pixels in height (y-axis). Due to the variability in optimal sizes, we apply a 3x3 average kernel to smooth the score distribution and identify an optimally representative size which corresponds to a size range of (300, 350). Conversely, templates that are excessively small or disproportionate yield suboptimal results due to insufficient structural representation. Furthermore, to determine the appropriate number of samples, we utilize a line plot, revealing that performance improves with an increasing number of templates up to approximately 30, after which the gains plateau. Based on this observation, we select 35 as the optimal number of samples, as it resides within the performance plateau while avoiding unnecessary computational overhead.

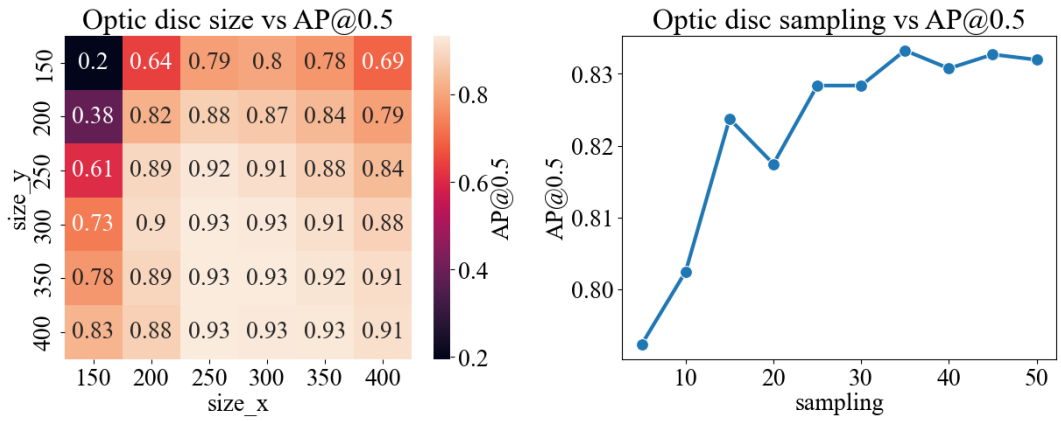

Figure 4.6 The template parmeter of optic disc. Heatmap depictes the sensitivity of template dimensions via the AP score. The line plot shows the impact of sampling.

Similarly to the optic disc, we conduct an investigation into the optimal parameters for macula template generation, focusing on both template size and the number of samples. Given the smaller anatomical size of the macula, the template size search space is restricted to 100–350 pixels, while the number of samples ranges from 5 to 50 in steps of 5. As illustrated in Figure 4.7, the heatmap indicates that template widths (x-axis) between 150–250 pixels and heights (y-axis) between 250–350 pixels yield the highest performance. To address the variability in optimal size—similar to the optic disc case—we apply an average kernel, resulting in an optimal template size of (200, 300). Moreover, further analysis demonstrates very small or excessively large templates degrade performance due to inadequate or overly diffuse anatomical representation. The corresponding line plot reveals a rapid increase in performance as the number of samples rises to 15, beyond which the performance stabilizes. Only minimal gains are observed beyond 35 samples. Ultimately, considering resource efficiency similar to the approach taken for the optic disc, we select 35 as the optimal number of samples, that balance efficiency and accuracy in macula detection.

Based on our investigation, we determined the optimal templates as depicted in Figure 4.8. For the optic disc, the optimal template size was determined to be (200, 300), and the sampling amount was set to 20 images. As same manner, for the macula, we selected a template size of (200, 200) with a sampling amount of 35 images.

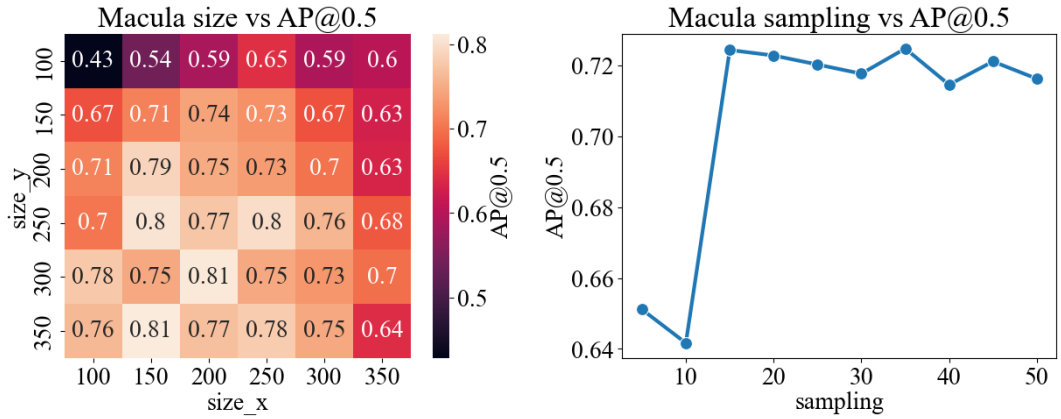

Figure 4.7 The template parmeter of macula. Heatmap depictes the sensitivity of template dimensions via the AP score. The line plot shows the impact of sampling.

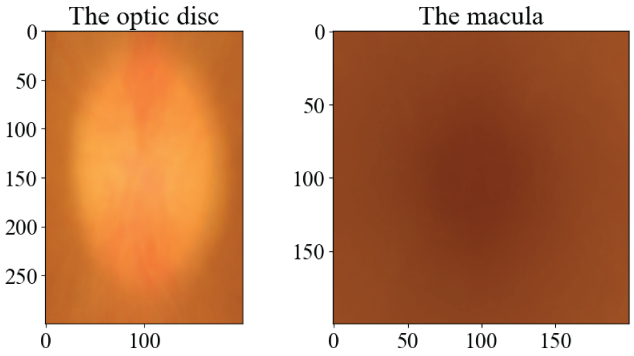

Figure 4.8 The optimal template for optic disc and macula.

4.2.2 Matching functions

Further study concerns the influence of the different matching functions on an AP50 score. These functions are shown below:

- **Sum of square differences (SQDIFF)** and **Normalized sum of square differences (SQDIFF_NORMED)**

  SQDIFF function:

$$R(x,y) = \sum_{x',y'} \left(T(x',y') - I(x+x',y+y')\right)^2 \tag{4.1}$$

  SQDIFF_NORMED function:

$$R(x,y) = \frac{\sum_{x',y'} \left(T(x',y') - I(x+x',y+y')\right)^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \tag{4.2}$$

- **Cross correlation (CCORR)** and **Normalized cross correlation (CCORR_NORMED)**

  CCORR function:

$$R(x,y) = \sum_{x',y'} \left(T(x',y') \cdot I(x+x',y+y')\right) \tag{4.3}$$

CCORR_NORMED function:

$$R(x,y) = \frac{\sum_{x',y'}\left(T(x',y') \cdot I(x+x',y+y')\right)}{\sqrt{\sum_{x',y'}\left(T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \tag{4.4}$$

- **Correlation coefficient (CCOEFF)** and **Normalized correlation coefficient (CCOEFF_NORMED)**

CCOEFF function:

$$R(x,y) = \sum_{x',y'}\left(T'(x',y') \cdot I'(x+x',y+y')\right) \tag{4.5}$$

CCOEFF_NORMED function:

$$R(x,y) = \frac{\sum_{x',y'}\left(T'(x',y') \cdot I'(x+x',y+y')\right)}{\sqrt{\sum_{x',y'}\left(T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}} \tag{4.6}$$

where,

$$T'(x',y') = T(x',y') - \frac{1}{w \cdot h} \cdot \sum_{x'',y''} T(x'',y'') \tag{4.7a}$$

$$I'(x+x',y+y') = I(x+x',y+y') - \frac{1}{w \cdot h} \cdot \sum_{x'',y''} I(x+x'',y+y'') \tag{4.7b}$$

In Table 4.5, we observe that the normalized function consistently yields superior scores compared to the non-normalized function. This discrepancy likely occurs due to variations in luminosity and contrast present in retinal fundus images while the template remains in fixed conditions. Consequently, these variations can lead to false detections when using non-normalized functions. Conversely, the normalized function is specifically designed to normalize both the target image and the template, thereby mitigating the influence of these variations and enhancing algorithm stability and accuracy. Ultimately, we choose CCOFF_NORMED to be the matching function for ocular structure detection, as it can outperform the other normalized functions.

Table 4.5 CCOEFF_NORMED achieves the highest AP_{50} score for both detection task.

	Optic disc	Macula		Optic disc	Macula
SQDIFF	0.386	0.087	**CCORR_NORMED**	0.542	0.438
SQDIFF_NORMED	0.420	0.090	**CCOEFF**	0.381	0.000
CCORR	0.387	0.000	**CCOEFF_NORMED**	0.948	0.830

4.2.3 Region of interest (ROI)

Further on, we examine the influence of the ROI technique on the overall algorithm performance, as detailed in Table 4.6. The AP_{50} score shows a improvement, from 0.811 to 0.847. However, when measured using the ED error, macula detection with ROI yields a higher value compared to without ROI. This increase is occured to the ROI technique constraining the detection algorithm to identify the macula location within the specific area, which occasionally leads to the prediction of locations that merely resemble the macula. As a result of the quantitative results, the ROI technique appears to be slight improvement for detection performance. Addtionally, upon examining the qualitative results in Figure 4.9, reveals that the ROI technique significantly enhances detection accuracy by effectively guiding the macula detector to focus within the appropriate anatomical region.

Table 4.6 The quantitative result of ROI technique.

Method	ED (pixels)	AP_{50}
Macula w/o ROI	19.4	0.811
Macula w ROI	20.4	0.847

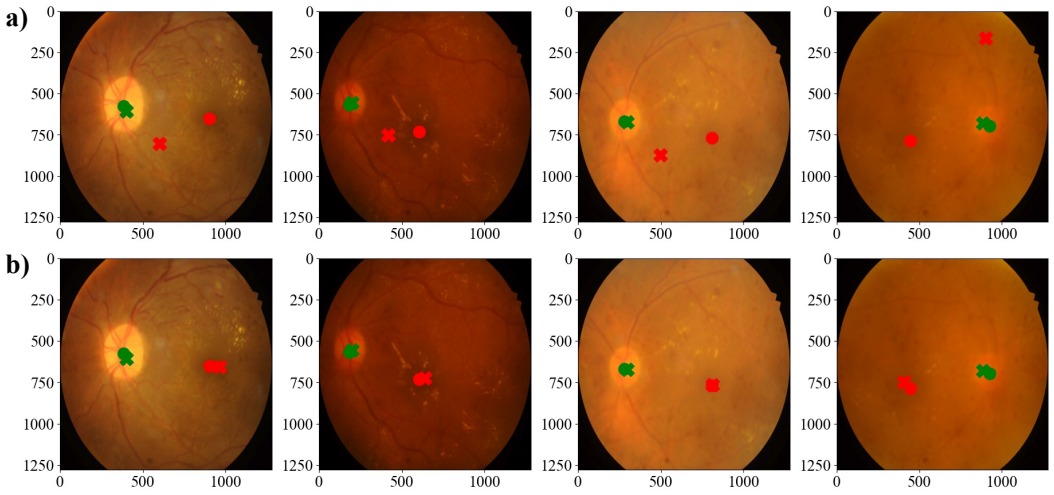

Figure 4.9 The qualitative result of macula detection, a) without ROI, b) with ROI. Where, cross sign (✕) is a predicted location, the dot sign (●) is a ground-truth location, the green color represents the optic disc, and the red color represents the macula.

4.2.4 The generalization of the proposed method

In this section, we investigate the generalizability of our approach using an Out-of-Distribution (OOD) testing framework. In this testing, models are trained on one dataset and evaluated on a different dataset, as summarized in Table 4.7. This form of testing presents a significant challenge, as it introduces distribution shift—an inherent issue in real-world applications that non-generalized models often struggle to address. The overall results suggest that our proposed method can effectively manage this challenge. Although it does not achieve the top performance in the second testing scenario, it still delivers competitive results. Within the machine learning context, inherently well-generalized models such as logistic regression and random forest exhibit stable performance across unfamiliar distributions. In particular, random forest, which employs a bagging algorithm, benefits from reduced variance and improved adaptability to diverse data distributions. Additionally, the observed asymmetry in generalization performance may occur from dataset characteristics: the SUTH dataset is balanced, whereas the Maharaj dataset is imbalanced. This discrepancy can significantly affect model transferability and the reliability of screening outcomes.

Table 4.7 The performace of machine learning in image screening. **Bold** represents the best score and underline represents the second best score.

Train	Test	Model name	Accuracy	Precision	Recall	FDR
Maharaj	SUTH	**LogisticRegression**	0.755	0.742	0.854	0.258
		DecisionTree	0.670	0.650	0.872	0.350
		SVC	0.685	0.651	**0.926**	0.349
		RandomForest	**0.767**	0.739	0.893	0.261
		HistGradientBoosting	0.737	0.703	0.908	0.297
		XGBoost	0.745	0.706	0.923	0.294
		LightGBM	0.714	0.679	0.914	0.321
		rulebase	0.757	**0.758**	0.821	**0.242**
SUTH	Maharaj	**LogisticRegression**	0.758	0.821	**0.875**	0.179
		DecisionTree	0.765	0.891	0.791	0.109
		SVC	**0.815**	0.906	0.847	0.094
		RandomForest	0.765	0.940	0.741	0.060
		HistGradientBoosting	0.787	0.939	0.772	0.061
		XGBoost	0.799	**0.947**	0.781	**0.053**
		LightGBM	0.772	0.934	0.756	0.066
		rulebase	0.748	0.932	0.725	0.068

4.3 Results of DR Grading

Currently, the architecture of the proposed model includes Swin s as the backbone network for feature extraction, followed by three fully connected layers with 2208, 64, and 5 nodes, respectively, designed for the grading task. In terms of hyperparameters, the input data is an RGB retinal images with a resolution of 512x512 pixels. The model is trained with a learning rate of 0.0002, a batch size of 8, and using the AdamW optimizer with default parameters from the PyTorch library. During training, data augmentations such as random horizontal flips, random equalization, and random rotations are applied to enhance model robustness and generalization.

The performance of the proposed model is illustrated by the confusion matrix in Figure 4.10, which reveals a substantial disparity in predictive accuracy between majority and minority classes. Specifically, the model correctly classifies 356 out of 361

instances in class 0, while achieving only 15 correct predictions out of 38 for class 3. Misclassification patterns are particularly evident in minority classes, especially classes 3 and 4, underscoring the difficulty in distinguishing advanced stages of diabetic retinopathy (DR), where lesion characteristics tend to overlap. This indicates that while the model demonstrates strong performance in detecting early-stage DR, further enhancement is needed for accurate classification of advanced stages. In addition to overall prediction trends, the confusion matrix also reveals instances of severe misclassification. For example, 22 images with a ground truth of class 4 were misclassified as classes 1 and 2. Therefore, analyzing these misclassifications could provide insights into the potential areas for model improvement, particularly in advanced DR detection.

Figure 4.10 The confusion matrix of proposed model.

In the same manner as the confusion matrix, the classification report in Table 4.8 indicates that the model has a better understanding of retinal images in the majority class compared to the minority class. The F1-scores for classes 1 and 2 are 0.993 and 0.785, respectively, while the average F1-score for the minority class is approximately 0.561. This significant gap underscores the model's performance discrepancy between the majority and minority classes. Additionally, the average F1-score across these classes shows a substantial difference between the F1 macro score of 0.693 and the micro score of 0.843. This disparity occurs from the model's poor performance on the minority classes, as evidenced by the precision and recall scores. Additionally, in the practica term, the precision macro score of 0.730 indicate that the model is confident to predict the severity of DR level as each class around 73 percent of the

time, while the recall macro score of 0.693 indicate that the model can detect only 69 percent of all positive images. Consequently, to improve the model's performance, it is imperative to address the imbalance issue, achieved by either balancing the amount of data in each class or by modifying the model's architecture or input data to better define the boundary conditions between classes, particularly the minority classes.

Table 4.8 The classification report of proposed model on the APTOS 2019 dataset.

class	Precision	Recall	F1-score	Support
0	1.000	0.986	0.993	361
1	0.615	0.793	0.694	74
2	0.757	0.814	0.785	199
3	0.441	0.395	0.412	38
4	0.839	0.441	0.578	59
Macro avg	0.730	0.687	0.693	731
Micro avg	0.853	0.845	0.843	731

Conversely, Figure 4.11 illustrates the ROC curve, which demonstrates satisfactory results due to the favorable shape of the curve and the high AUC across all classes. For instance, class 0 has an AUC of 1.00, and class 3 has an AUC of 0.92, which starkly contrasts with the findings from the confusion matrix and classification report. This conflict arises due to the imbalance issue, where a large amount of true negative data leads to consistently high AUC values on the ROC curve. Therefore, in the context of imbalanced datasets, the ROC curve may be non-informative and unable to accurately reflect the model's true performance. Consequently, to better visualize the model performance in this context, we introduce the Precision-Recall (PR) curve, which offers a more appropriate representation despite having similar logical foundations as the ROC curve as shown in Figure 4.12. However, in the table summarizing the model's general performance, we still have to report the AUC score of the ROC curve instead of the AUC of the PR curve, referred to as the average precision (AP) score, as the AUC score is commonly utilized for performance comparison in numerous research papers.

Additionally, we comprehensively investigate the model's performance on a balanced test dataset. In this experiment, 38 images are randomly selected from each class, matching the number of images in the minority class, to eliminate the influence of class imbalance. The results, summarized in Table 4.9 and visualized in Figure 4.13,

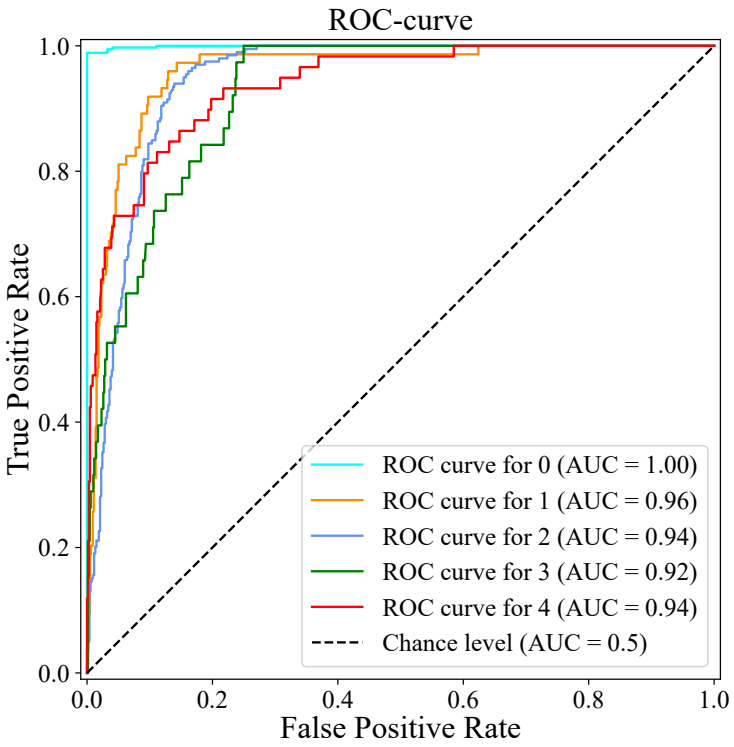

Figure 4.11 The figure illustrates the ROC curve and AUC for each class, indicating that the proposed model outperforms in classifying class 1 across a variety of threshold values. However, the other classes exhibit constraints related to the threshold value, impacting their lower classification performance.

surprisingly reveal the outstanding performance for class 0 (F1 = 0.993) and class 1 (F1 = 0.786). This contrasts with the performance observed on the imbalanced test dataset, where the model is expert on class 0 and class 2. Furthermore, analysis of the confusion matrix shows that the model continues to struggle with accurately classifying classes 3 and 4. These stages are frequently misclassified as class 2, resulting in a high false positive rate for class 2. Actually, this misclassification pattern also appears in the imbalanced test set, it is less apparent due to the disproportionately large number of class 2 samples, which masks the underlying issue. Further insights are provided by the ROC curves in Figure 4.14 and the PR curves in Figure 4.15. Class 0 achieves perfect classification, while classes 1, 3, and 4 also demonstrate acceptable performance. However, class 2 consistently underperforms across both evaluation metrics, indicating its inherent difficulty due to overlapping feature characteristics with adjacent severity levels. Eventually, these results confirm the model's robustness in detecting early DR

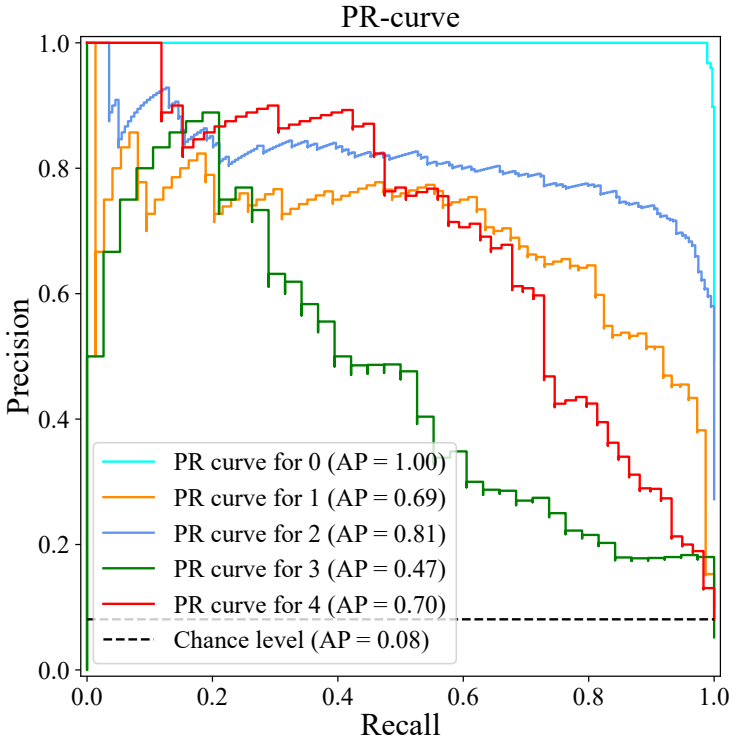

Figure 4.12 The PR curve and AUC for each class, which is more informative than the ROC curve in context of imbalance dataset.

stages and its potential for reliable performance in balanced conditions. However, the limited accuracy in classifying advanced stages (particularly classes 3 and 4) remains a challenge. To improve grading performance, it is essential to enhance the model's capacity to distinguish advanced DR stages. This may be accomplished through balancing the training dataset or by modifying the model architecture to better define the boundary conditions between classes, especially for minority classes.

Table 4.9 The classification report of proposed model on the APTOS 2019 dataset.

class	Precision	Recall	F1-score	Support
0	1.000	0.921	0.959	38
1	0.717	0.868	0.786	38
2	0.441	0.790	0.566	38
3	0.714	0.395	0.509	38
4	0.850	0.447	0.586	38
Macro avg	0.745	0.684	0.681	190
Micro avg	0.745	0.684	0.681	190

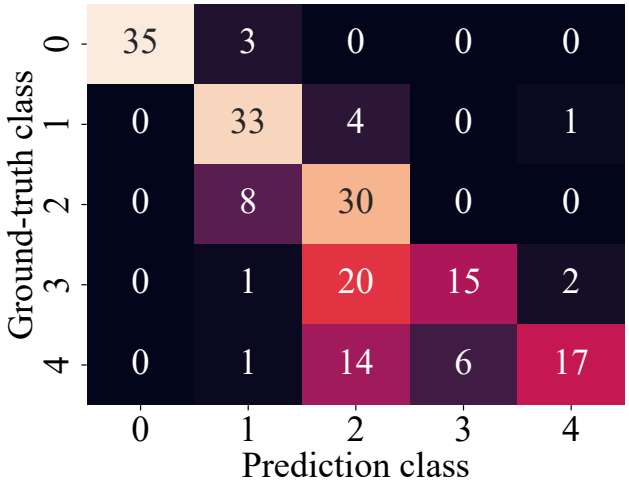

Figure 4.13 The confusion matrix of proposed model.

Ultimately, Table 4.10 compares the performance between the proposed model and previosu related work, which is from difference vision architecture such as CNN and Vision-mamba, on the same dataset, indicating that the proposed model can outperforms the other models. This comparison reveals the backbone network based on Vision Transformer (ViT) architecture, can provide a significant improvement in the grading task. The QWK score of the proposed model is 0.903, which is higher than the previous work, suggesting that the proposed model can achieve a comparable grading performance as ophthalmologist. Nevertheless, the significant disparity between the F1 scores and accuracy reveals the proposed model's inferior performance in the classification task compared to the ophthalmologists. Consequently, to achieve our criteria,

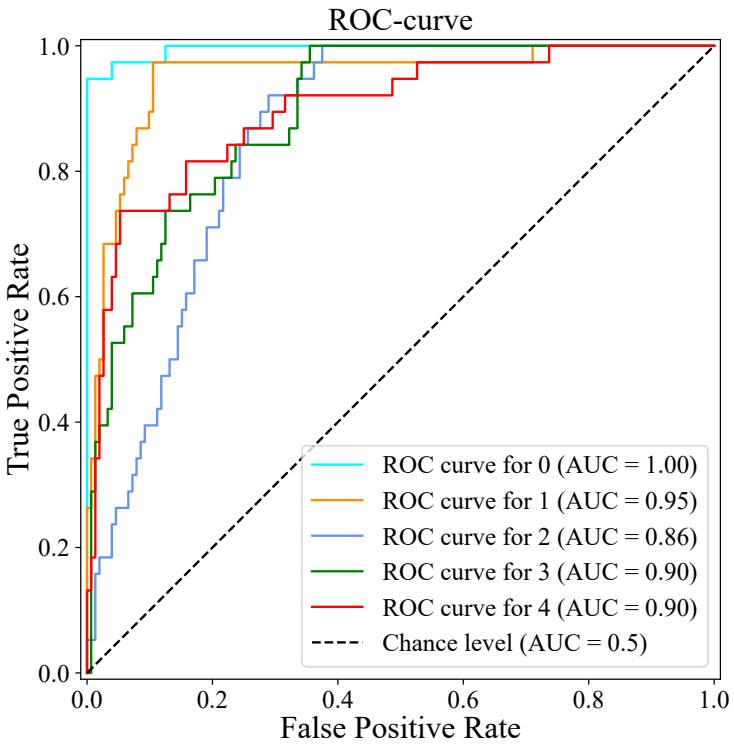

Figure 4.14 The figure illustrates the ROC curve and AUC for each class, indicating that the proposed model outperforms in classifying class 1 across a variety of threshold values. However, the other classes exhibit constraints related to the threshold value, impacting their lower classification performance.

substantial improvements are necessary for the classification task.

Table 4.10 The general performance of proposed model compared the human performance. **Bold** represents the best score.

Dataset	Method	Acc.	Micro		Macro		QWK
			F1	AUC	F1	AUC	
EyEPACS-2	Ophthalmologists Krause et al., 2018	0.895	0.895	-	0.714	-	0.871
APTOS 2019	VGG16+Xception+CNN Bodapati et al., 2021	0.827	0.818	-	0.663	-	0.864
	DenseNet169+CBAM+INS Farag et al., 2022	0.822	0.825	-	0.685	-	0.888
	VMamba-m Xue et al., 2024	0.786	0.786	-	0.661	-	0.784
	Proposed model	**0.845**	**0.843**	**0.967**	**0.693**	**0.952**	**0.903**

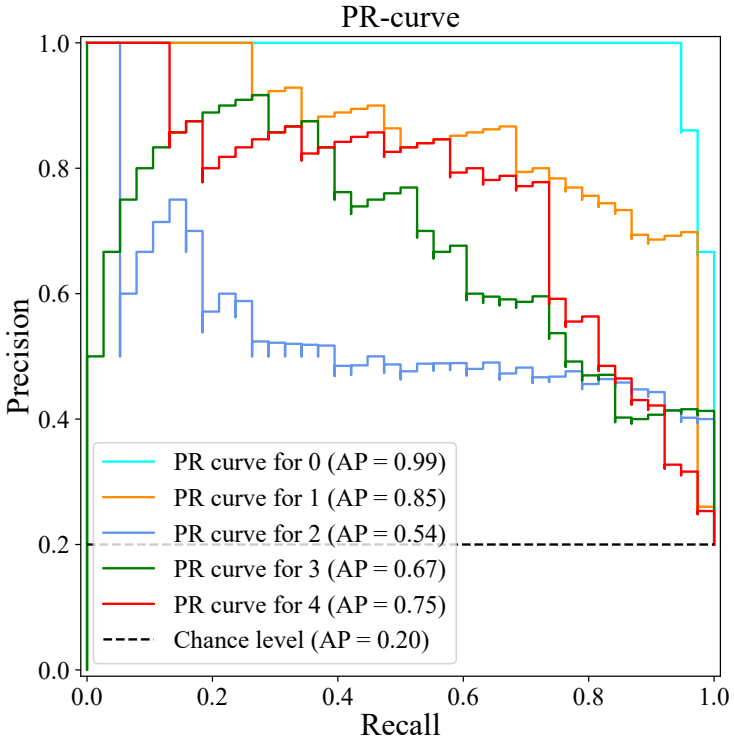

Figure 4.15 The PR curve and AUC for each class, which is more informative than the ROC curve in context of imbalance dataset.

4.4 Ablation Study of DR Grading

In this section, we examine the influence of various training improvement techniques on the validation set of the APTOS 2019 dataset. The DR severity grading task inherently faces the challenge of class imbalance. Therefore, appropriate metrics for this task are macro scores, including the F1 macro score and the AUC-ROC macro score. These metrics are suitable because they measure performance across all classes with equal weight, in contrast to micro scores, which balance each class according to its own size.

4.4.1 Backbone model selection

First, we investigate the backbone networks to determine the optimal feature extraction network for our task. Given the numerous networks currently proposed for DR grading, it is impractical to thoroughly evaluate them. Therefore, we emphasize five of the most commonly used networks: VGG 19 (Simonyan and Zisserman, 2014), Inception

V3 (Szegedy et al., 2016), ResNet 50 (K. He et al., 2016), DenseNet 161 (G. Huang et al., 2017), and Swin Transformer (Liu et al., 2021). As shown in Table 4.11, DenseNet 161 achieves the highest scores for both macro-metrics, with an F1 score of 0.579 and an AUC of 0.933, and ranks second for the other metrics, which is impressive for a non-tuned network. However, this is a grading task, so the Quadratic Weighted Kappa (QWK) score must also be considered. On this metric, Swin Transformer significantly outperforms the other networks and also ranks first or second for the other scores. As demonstrated, DenseNet 161 and Swin Transformer exhibit impressive performance, presenting a dilemma in network selection. To address this challenge effectively, we consider employing both networks as backbone models.

Table 4.11 The general performance of each backbone model. **Bold** represents the best score and underline represents the second best score.

Model name	Acc.	Micro		Macro		QWK
		F1	AUC	F1	AUC	
VGG19	0.726	0.726	0.866	0.333	0.731	0.747
Inception V3	0.761	0.761	0.943	0.512	0.888	0.785
ResNet50	0.762	0.762	0.953	0.526	0.910	0.809
DenseNet161	0.792	0.792	0.963	**0.579**	**0.933**	0.822
Swin s	**0.793**	**0.793**	**0.964**	0.515	0.931	**0.843**

4.4.2 Data sampler

For further study, we investigated the data sampler strategy for effective network training. We compared conventional data samplers, typically sequential or random, with a sampler designed to address the imbalance issue, termed the imbalance data sampler. This sampler attempts to distribute data evenly across classes within each training batch, as illustrated in Figure 4.16. Consequently, the network gradually and equally comprehends each class, leading to a slight improvement in performance. Table 4.12 illustrates that the imbalanced data sampler notably enhanced the performance of DenseNet 161, particularly in the F1 macro score and QWK, which increased from 0.579 to 0.662 and 0.822 to 0.840, respectively. Similarly, the Swin Transformer also exhibited improvement in several metrics, particularly in the F1 macro score, which increased by approximately 0.14 points, and in the QWK, which improved from 0.843

to 0.869. In conclusion, the imbalanced data sampler demonstrates potential for improving network performance by optimizing the data sampling process.

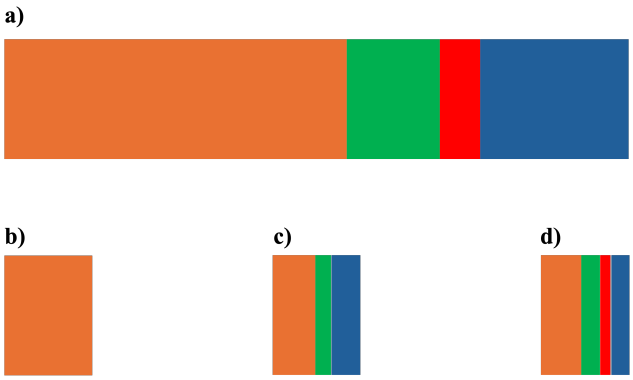

Figure 4.16 The figure illustrates the data sampler strategies: a) the entire training dataset; b) the training data in a batch with a sequential sampler strategy, which sequentially selects the data from beginning to end; c) the training data in a batch with a random sampler strategy, which randomly selects the data from the entire dataset; and d) the training data in a batch with an imbalance sampler strategy, which attempts to select the data from each class. Each color represents data from a different class.

Table 4.12 The influence of imbalance data sampler to DenseNet 161 and Swin Transformer. **Bold** indicates the best score.

Model name	Acc.	Micro		Macro		QWK
		F1	AUC	F1	AUC	
DenseNet161	0.792	0.792	**0.963**	0.579	**0.933**	0.822
+Imbalance sampler	**0.798**	**0.798**	0.956	**0.662**	0.928	**0.840**
Swin s	0.793	0.793	**0.964**	0.515	**0.931**	0.843
+Imbalance sampler	**0.810**	**0.810**	0.962	**0.657**	0.927	**0.869**

4.4.3 Loss functions

In our third study, we investigated different loss functions employed in training neural networks, as selecting the appropriate loss function is crucial for guiding the network towards optimal performance.

- **Cross-entropy loss (CE):** Cross-entropy loss is used to measure model performance by quantifying the difference between the predicted distribution **Q** and the ground truth distribution **P**, which can be described in mathematical form as (Eq. 4.8).

$$CE(P, Q) = -\sum_{i=1}^{C} P(x_i)\log(Q(x_i)) \tag{4.8}$$

  where, C is the number of classes, x_i is the data point for i^{th} class.

- **Weighted cross-entropy loss (WCE):** Weighted cross-entropy loss is the extension of the standard cross-entropy loss by adding a weighted parameter for each class to mitigate the impact of imbalanced data during training, as shown in (Eq. 4.9).

$$WCE(P, Q) = -\sum_{i=1}^{C} w_i P(x_i)\log(Q(x_i)) \tag{4.9}$$

  where, C is the number of classes, x_i is the data point and w_i is the weight for i^{th} class, which typically computed as N/N_i. Here, N_i represents the number of data points in the i^{th} class, and N is the total number of data points.

- **Focal loss (FL) (Lin et al., 2017):** Focal loss is an extension of the cross-entropy loss, specifically designed to mitigate the imbalance between foreground and background classes in object detection tasks. Therefore, due to our imbalanced dataset, we decided to utilize this loss function. The focal loss is defined as follows:

$$FL(P, Q) = -\sum_{i=1}^{C} P(x_i)\left(1 - Q(x_i)\right)^{\gamma}\log(Q(x_i)) \tag{4.10}$$

  where, C is the number of classes, x_i is the data point for i^{th} class. γ is the focusing parameter $\gamma \geq 0$, influencing the model 's attention on the misclassified example or minority data point.

Table 4.13 illustrates that the CE loss yields the best performance across multiple metrics, particularly the F1 scores, which highlight the superior classification capabilities among the evaluated loss functions. In contrast, the FL loss exhibits better performance for the grading task, achieving roughly 0.853 for the QWK score. Interestingly, the WCE and FL losses do not perform as well as anticipated across various metrics. This issue might arise from the use of an imbalanced data sampler, which assists every batch to become a balanced batch, thereby causing the focusing parameter γ in FL loss to negatively affect the loss value during training. Additionally, the weight parameter in WCE loss is a fixed, global parameter that does not adapt to the class distribution within each batch, resulting in an ineffective weight for the loss value during training. Eventually, based on the results presented in the table, we can conclude that CE loss is the optimal loss function for training the model in conjunction with the imbalanced data sampler.

Table 4.13 The influence of loss functions on DenseNet 161 and Swin Transformer. **Bold** represents the best score.

Model name	Loss function	Acc.	Micro		Macro		QWK
			F1	AUC	F1	AUC	
DenseNet161	**CE**	**0.798**	**0.798**	0.956	**0.662**	**0.928**	0.840
	WCE	0.778	0.778	0.950	0.633	0.921	0.842
	FL (γ = 1)	0.796	0.796	0.956	0.639	0.924	0.857
	(γ = 2)	0.795	0.795	0.956	0.656	0.925	0.851
	(γ = 3)	0.792	0.792	**0.958**	0.635	0.926	**0.856**
Swin s	**CE**	**0.810**	**0.810**	**0.962**	**0.657**	0.927	**0.869**
	WCE	0.791	0.791	0.957	0.630	0.925	0.851
	FL (γ = 1)	0.803	0.803	0.960	0.643	**0.930**	0.858
	(γ = 2)	0.808	0.808	0.961	0.637	**0.930**	0.846
	(γ = 3)	0.780	0.780	0.955	0.600	0.924	0.840

4.4.4 The impact of SMOTE

In this investigation, we explore the influence of the Synthetic Minority Oversampling Technique (SMOTE) through the model's performance and K-nearest neighbour performance. Table 4.14 illustrates the performance of Swin model with and without SMOTE. The result indicate that model is trained with synthetic data from

SMOTE can achieve a higher score on both severity level and the overall performance, as shown that the F1 macros score increase from 0.637 to 0.659 in Swin s model. As a same manner, SMOTE also imrpove the performance of DenseNet 161 model, as illustrated in Table 4.15, where the F1 macro score increases from 0.637 to 0.657. This improvement is particularly significant for the minority classes such as class 3 and 4, while compensating with performance to classify the class 1. However, the majority class (class 1) also experiance a slight increase in performance, which from roughly 0.7 to 0.72. This indicates that the model can learn more robust features from the synthetic data generated by SMOTE, leading to better generalization and classification performance across various classes.

Table 4.14 The classification report of Swin Transformer model with and without SMOTE.

class	Support	Witout SMOTE			With SMOTE		
		Precision	Recall	F1	Precision	Recall	F1
0	361	0.949	0.983	0.966	0.962	0.983	0.973
1	74	0.535	0.730	0.617	0.577	0.662	0.616
2	199	0.753	0.643	0.694	0.729	0.729	0.729
3	38	0.421	0.421	0.421	0.400	0.474	0.434
4	59	0.604	0.492	0.542	0.768	0.424	0.544
Macro avg	731	0.652	0.654	0.648	0.685	0.654	0.659
Micro avg	731	0.799	0.796	0.794	0.814	0.810	0.808

To further investigate the models' understanding, we analyze the quality of the feature vectors extracted from the backbone networks by applying the K-Nearest Neighbors (KNN) algorithm within the feature space. As shown in Table 4.16, the KNN classifier trained with SMOTE achieves higher F1 macro scores on both the training and validation sets. Particularly, the feature vectors extracted from the Swin model show a performance gain of approximately 5.7% with the application of SMOTE. Therefore, the results ensure the effectiveness of SMOTE in improving the model's performance by addressing class imbalance.

An additional investigation into the impact of SMOTE on model performance using a balanced test dataset is presented in Table 4.17. The results indicate that the F1 macro score of the Swin model increases from 0.648 to 0.680 when trained with

Table 4.15 The classification report of DenseNet161 model with and without SMOTE.

class	Support	Witout SMOTE			With SMOTE		
		Precision	Recall	F1-score	Precision	Recall	F1-score
0	361	0.973	0.986	0.979	0.952	0.989	0.970
1	74	0.505	0.743	0.601	0.612	0.554	0.582
2	199	0.739	0.668	0.702	0.708	0.779	0.742
3	38	0.360	0.474	0.409	0.500	0.421	0.457
4	59	0.808	0.356	0.494	0.684	0.441	0.536
Macro avg	731	0.677	0.646	0.637	0.691	0.637	0.657
Micro avg	731	0.817	0.798	0.797	0.806	0.814	0.807

Table 4.16 The classification performance of KNN model with different pretrained models. Notably, we use **macro** score for comparison

Model name	Without SMOTE		SMOTE	
	F1 (Train)	F1 (Val)	F1 (Train)	F1 (Val)
DenseNet161	0.858	0.535	0.950	0.552
Swin s	0.752	0.489	0.895	0.546

SMOTE, demonstrating a positive effect. In contrast, DenseNet161 exhibits a decline in performance, with its F1 macro score decreasing from 0.627 to 0.601 under the same conditions. This decline is arised to reduced classification accuracy in classes 1 and 4, as well as an increased false positive rate in class 2 when trained with SMOTE 4.18. As a result, we face with a dilemma because SMOTE generally enhances model performance, except for the DenseNet161 on the balanced dataset. To further evaluate the robustness of SMOTE, we apply a KNN classifier on features extracted from both models and test it on a balanced dataset, as shown in Table 4.19. The KNN results demostrate consistent improvements, with performance gains of approximately 0.6 for DenseNet161 and 0.9 for Swin, confirming the effectiveness of SMOTE in enhancing classification ability. In conclusion, the SMOTE technique proves beneficial in improving model performance on imbalanced datasets, particularly for the Swin architecture.

Table 4.17 The classification report of Swin Transformer model with and without SMOTE.

class	Support	Witout SMOTE			With SMOTE		
		Precision	Recall	F1	Precision	Recall	F1
0	38	0.900	0.947	0.923	0.878	0.947	0.911
1	38	0.674	0.763	0.716	0.750	0.711	0.730
2	38	0.433	0.684	0.531	0.446	0.763	0.563
3	38	0.643	0.474	0.546	0.750	0.474	0.581
4	38	0.790	0.395	0.526	0.792	0.500	0.613
Macro avg	190	0.688	0.653	0.648	0.723	0.679	0.680
Micro avg	190	0.688	0.653	0.648	0.723	0.679	0.680

Table 4.18 The classification report of DenseNet161 model with and without SMOTE.

class	Support	Witout SMOTE			With SMOTE		
		Precision	Recall	F1-score	Precision	Recall	F1-score
0	38	0.857	0.947	0.900	0.783	0.947	0.857
1	38	0.700	0.737	0.718	0.714	0.526	0.606
2	38	0.426	0.605	0.500	0.389	0.737	0.509
3	38	0.593	0.421	0.492	0.700	0.421	0.525
4	38	0.630	0.447	0.523	0.714	0.395	0.509
Macro avg	190	0.641	0.632	0.627	0.659	0.605	0.601
Micro avg	190	0.641	0.632	0.627	0.659	0.605	0.601

Table 4.19 The classification performance of KNN model with different pretrained models. Notably, we use textbfmacro score for comparison

Model name	Without SMOTE		SMOTE	
	F1 (Train)	F1 (Val)	F1 (Train)	F1 (Val)
DenseNet161	0.858	0.499	0.950	0.561
Swin s	0.752	0.455	0.895	0.564

4.4.5 The impact of fine-tuning

In this section, we examine the impact of fine-tuning on the performance of backbone models. Fine-tuning serves as a critical step in adapting pre-trained models to task-specific domains by leveraging representations learned from large-scale datasets. We evaluate the performance of both pre-trained and fine-tuned models on the APTOS 2019 dataset, as summarized in Table 4.20. Both DenseNet161 and Swin s show impressive improvements following fine-tuning, achieving F1 macro scores of 0.680 and 0.693, respectively, compared to 0.657 and 0.659 in their pre-trained model. As a result, this fine-tuning provide the substantial benefit, especially in this challenging task, which involves imbalanced class distributions and overlapping lesion features in advanced stages (classes 2, 3, and 4). In such a context, even a 5% gain in F1 macro score is a significant enhancement in model performance.

Table 4.20 The classification performance of DenseNet161 and Swin s models before and after fine-tuning. Pretrained models are trained on ImageNet-1K dataset. Fine-tuned models are pretrained model and then, tuned on APTOS 2019 dataset. Notably, we use only **macro** scores for comparison.

Model name	Pretrained			Fine-tuned		
	Precision	Recall	F1	Precision	Recall	F1
DenseNet161	0.691	0.637	0.657	0.710	0.662	0.680
Swin s	0.685	0.654	0.659	0.730	0.687	0.693

To assess the model's ability to understand and represent data, we extract feature vectors from both pre-trained and fine-tuned backbone models. We check these representations in two ways: by using a K-Nearest Neighbors (K-NN) classifier for quantitative evaluation and t-distributed stochastic neighbor embedding (t-SNE) for qualitative evaluation. The K-NN results reveal that fine-tuning substantially enhances classification performance for both RGB and grayscale images, with average improvements of approximately 9.2% for DenseNet 161 and 16.1% for Swin s, as shown in Table 4.21. Notably, the fine-tuned Swin Transformer using RGB images achieves the highest validation score of 0.716, demonstrating its strong representational capacity. This result underscores the model's suitability for downstream classification tasks, as it effectively captures meaningful and discriminative features from the input data. The t-SNE visualizations in Figures 4.17 and 4.18 visualize the feature spaces of Swin and DenseNet161,

demonstrating the influence of fine-tuning and input modality. Fine-tuning significantly improves class separability in both models, with a clear improvement in classifying abnormal classes, corresponding to diabetic retinopathy (DR) severity levels greater than 1. In the fine-tuned models, abnormal classes form well-defined clusters, whereas in the pre-trained models, these classes are more intermingled and thus more challenging to classify. In conclusion, these studies confirm that fine-tuning enhances the quality of learned representations, enabling better identification of diabetic retinopathy stages.

Table 4.21 The classification performance of KNN model with different image types and pretrained models. Notably, we use only **macro** score for comparison

Model name	Image type	Pretrained		Fine-tuned	
		F1 (Train)	F1 (Val)	F1 (Train)	F1 (Val)
DenseNet161	Gray	0.968	0.593	0.978	0.647
	RGB	0.950	0.552	0.983	0.682
Swin s	Gray	0.935	0.532	0.977	0.684
	RGB	0.895	0.546	0.984	0.716

To further examine the impact of fine-tuning on model performance under balanced test conditions, the results are summarized in Table 4.22. The fine-tuned DenseNet161 model achieves an improved F1 macro score of 0.657, compared to its pre-trained model. In contrast, the Swin Transformer exhibits only a marginal improvement of 0.01 in F1 macro score after fine-tuning. These results show lower performance than those model obtained on the imbalanced test set, suggest that the models still face challenges in accurately classifying DR severity levels, although after fine-tuning. Furthermore, the KNN result highlights that RGB input images yield greater performance gains than grayscale images, especially in the fine-tuned setting 4.23. The highest validation F1 score of 0.701 is achieved by the KNN classifier using features extracted from the fine-tuned Swin Transformer trained on RGB inputs. These results confirm the importance of fine-tuning and color information in boosting the discriminative capacity of vision models for DR grading.

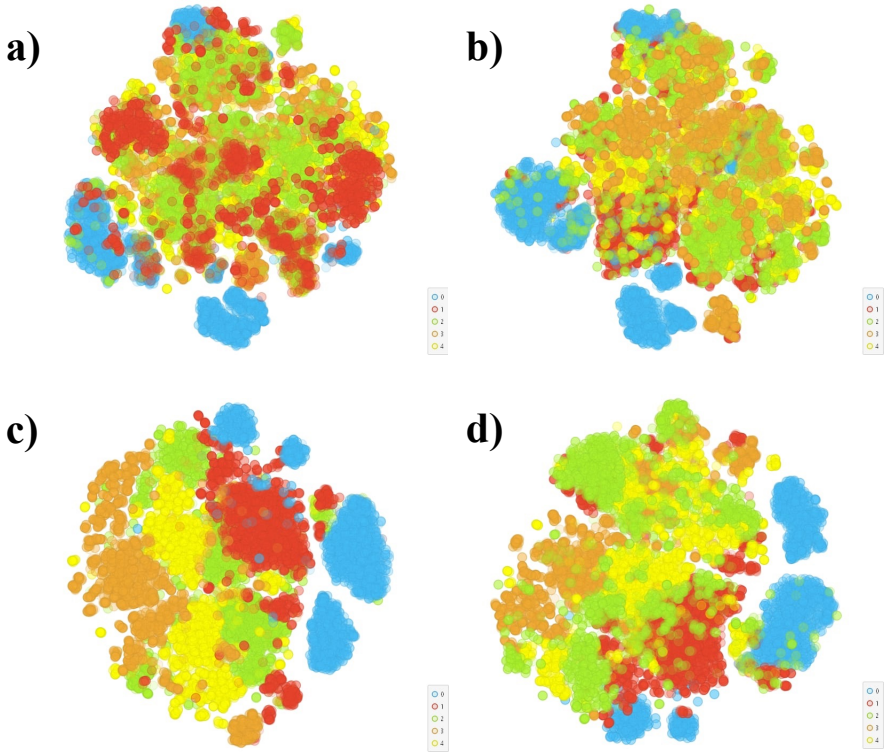

Figure 4.17 The t-SNE visualization of the DenseNet 161 model's feature space. The colors represent different classes; cyan is class 0; red is class 1; green is class 2; orange is class 3; and yellow is class 4, respectively. a) and b) illustrate the feature space of model without tuning while c) and d) illustrate the feature space of model with tuning. Moreover, a) and c) are the feature space of RGB images while b) and d) are the feature space of grayscale images.

Table 4.22 The classification performance of DenseNet161 and Swin s models before and after fine-tuning. Pretrained models are trained on ImageNet-1K dataset. Fine-tuned models are pretrained model and then, tuned on APTOS 2019 dataset. Notably, we use only **macro** scores for comparison.

Model name	Pretrained			Fine-tuned		
	Precision	Recall	F1	Precision	Recall	F1
DenseNet161	0.659	0.605	0.601	0.733	0.658	0.657
Swin s	0.723	0.679	0.680	0.745	0.684	0.681

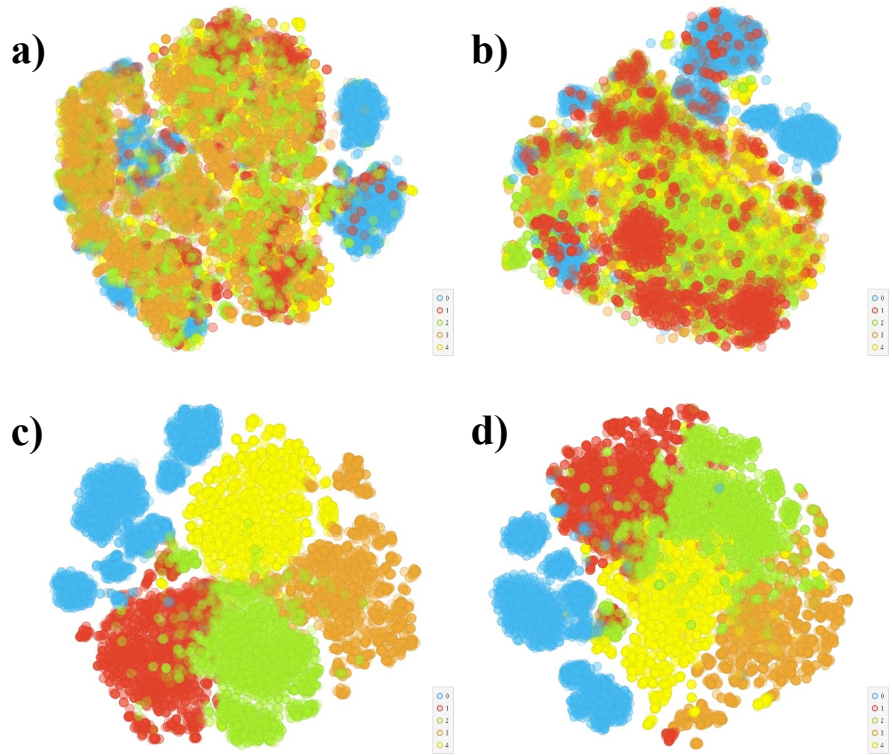

Figure 4.18 The t-SNE visualization of the Swin s model's feature space. The colors represent different classes; cyan is class 0; red is class 1; green is class 2; orange is class 3; and yellow is class 4, respectively. a) and b) illustrate the feature space of model without tuning while c) and d) illustrate the feature space of model with tuning. Moreover, a) and c) are the feature space of RGB images while b) and d) are the feature space of grayscale images.

Table 4.23 The classification performance of KNN model with different image types and pretrained models. Notably, we use only **macro** score for comparison

Model name	Image type	Pretrained		Fine-tuned	
		F1 (Train)	F1 (Val)	F1 (Train)	F1 (Val)
DenseNet161	Gray	0.968	0.635	0.978	0.622
	RGB	0.950	0.562	0.983	0.692
Swin s	Gray	0.935	0.576	0.977	0.684
	RGB	0.895	0.564	0.984	0.701