DRSS Severity Classification on OCT images

Santhana Bharathi Narasimmachari
School of Electrical & Computer Engineering
Georgia Institute of Technology
Atlanta, USA
sbn6@gatech.edu

Venkata Hanuma Sandilya Balemarthy
School of Electrical & Computer Engineering
Georgia Institute of Technology
Atlanta, USA
vbalemarthy3@gatech.edu

***Abstract*—Diabetes brings with it microvascular complications which are a subset of disorders that affect blood vessels in various organs. One such complication is Diabetic Retinopathy (DR), which occurs due to prolonged high blood sugar levels that cause blockage of the light-sensitive tissue at the back of the eye. In this paper, we discuss four unique machine learning-based classifier systems for Optical Coherence Tomography (OCT) images to classify DR using the OLIVES dataset.**

***Index Terms*—Diabetic Retinopathy, Biomedical classifier, Naive Bayes, KNN, SVM, U-Net CNN.**

I. Introduction

The project aims to develop a Diabetic Retinopathy Severity Scale (DRSS) classifier using OCT images from the PRIME component of the OLIVES dataset. Four distinct classifiers are implemented and evaluated based on balanced accuracy, F1 score, and other metrics. The dataset comprises 32,337 samples, divided into 658 patient volumes, with each volume containing 49 right (OD) and left (OS) eye images representing a single patient's clinical visit.

II. Data Analysis & Dataset Visualization

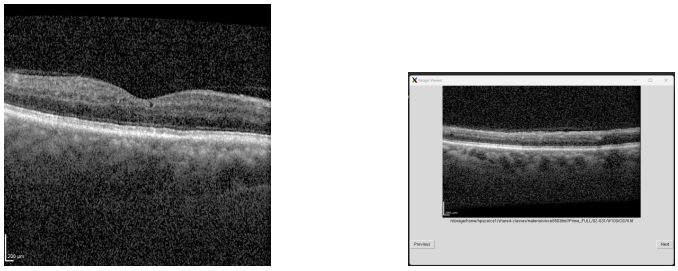
A. Data Analysis

Initially, our aim was to gather information about the methods involved in assessing the DRSS value in a practical scenario. Our preliminary research indicated that OCT images were utilized to infer central macular thickness based on retinal features such as reflectivity, curvature, and thickness. To obtain these features, the classifier must extract the critical retinal layer sections of the image shown in Fig. 1a. This analysis guided us in our search for techniques capable of extracting relevant features from OCT images.

While examining the class distribution, we found an imbalance in the dataset(class 0: 30%, class 1: 50%, and class 2: 20%). This imbalance forced us to look for compensation techniques like weighted random sampling, cost-sensitive learning, and randomizing inputs as discussed in Section III of the paper.

B. Dataset Visualization

We created a simple image viewer GUI to efficiently sift through images for obvious patterns as shown in Fig. 1b. During this analysis, we found that 49 adjacent OCT images captured as a series of closely spaced scans are to be combined to create a more comprehensive view of the tissue. This can help with the visualization and analysis of larger areas or complex structures within the biological tissue.



(a) Retinal layers (b) Case viewer GUI

Fig. 1: OCT Image Visualization

III. Pre-Processing & Feature Extraction

A. Compensating dataset imbalance

Given the imbalanced nature of the OLIVES dataset, we explored various techniques to address this issue as follows.

1) Weighted over sampler: The WeightedRandomSampler from PyTorch was used for training the data loader and assigning higher weights to underrepresented classes. An increase in test accuracy after implementing this sampler was observed.

2) Random Flip: Random flip transform is a data augmentation technique that creates a mirror image of the original input along the vertical axis. It helps the model become more robust to data variations.

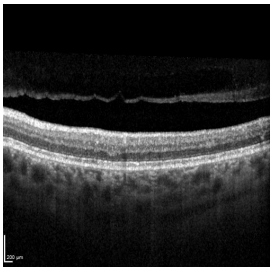
3) Random Rotation: Random rotation transform is a data augmentation technique that creates rotated images at random input intervals.

4) Cost-sensitive learning: Implementing a customized loss for penalizing miss predictions on underrepresented classes helped counter the impacts of dataset imbalances.

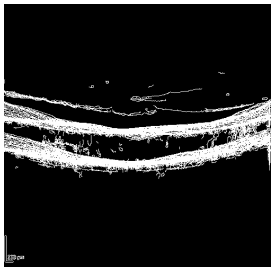
B. OCT Image Pre-processing

Feature extraction techniques improved training efficiency by capturing relevant retinal layers. The dataset, consisting of 49 images per unique patient ID, week ID, and label combination, was consolidated into 658 derived images, retaining essential features. Several pre-processing methods listed below were used to achieve this.

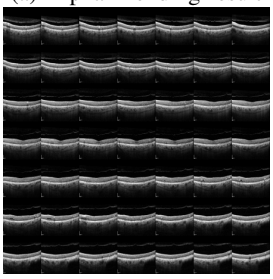
a) Laplacian pyramid fusion: This technique merges the most prominent characteristics of several input images to produce a unified composite image. It takes into account the local quality of each individual image as defined by their corresponding weight maps.



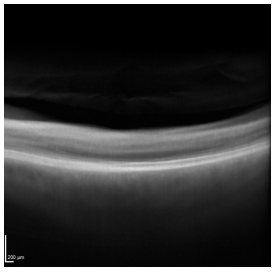
(a) Alpha Blending result



(b) Canny Edge results



(c) 7x7 Grid of original images



(d) Laplacian Pyramid fusion

Fig. 2: Pre-processing schemes

b) Alpha blending: This is a fusion method that merges multiple input images into a single composite image by executing a weighted linear interpolation of their pixel values. The assigned alpha values dictate the weights used in the blending process.

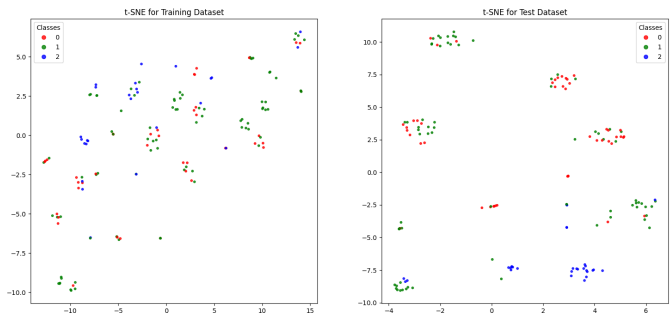
c) Canny edge detector: This is used in identifying edges within an image that represent boundaries or significant changes in intensity between adjacent pixels. In our case, this detection helps us in capturing the structural information such as retinal layers in the image. Based on our experiments, we noticed the highest accuracy when this pre-processing technique is combined with others.

d) Grid image format:original images: This idea stems from trying to mimic the doctor's approach in viewing scan images in a grid format. When the 49 images in a volume are placed in a grid, each scan's features are well preserved and the overall trend of the retinal layers are also captured allowing the classifier to exploit spatial hierarchies.

e) Grid image format:canny edge images: This variant was generated by combining the canny edge applied images into a 7x7 grid for the same reasons as the above.

C. Run-time Optimization

To efficiently train classifiers on the large OLIVES dataset, we implemented various optimization techniques like resizing images to 224x224 pixels, reducing feature size, using Google Colab premium compute units instead of local PACE ICE clusters, implementing batch-wise data loaders, and leveraging GPU optimizations. Additionally, we addressed challenges related to multiple processes by experimenting with 2-4 workers for data loaders.



(a) t-SNE Plot for Train Set (b) t-SNE Plot for Test Set

Fig. 3: t-SNE plots

IV. Classification Methods

Four different classification techniques used are listed below in increasing order of accuracy.

A. Support Vector Machine (SVM): ***(Accuracy range: 39-41%)***

SVM aims to identify the optimal hyperplane for separating input data points into respective classes. To assess the dataset's separability, we generated t-Distributed Stochastic Neighbor Embedding (t-SNE) plots for both training and testing sets, visualizing high-dimensional data in 2D space. In the training set plot (Fig.3a), pairs class-2/class-1 and class-2/class-0 have better separability. Conversely, the class-0/class-1 pair appears indistinguishable due to their proximity. In the testing set plot (Fig.3a), while the class-2/class-1 and class-2/class-0 pairs remain separable, the class-0/class-1 pair is intertwined, complicating classification.

In our study, we employed a One-vs-Rest (OvR) classifier using the SGDClassifier from scikit-learn to handle the three output classes. An individual binary classifier is trained for each class, where the class is treated as positive and the others are combined as negative. To address the requirement of a non-linear kernel for class separation, we incorporated RBF sampler, which efficiently approximates the kernel in a lower-dimensional space, maintaining computational efficiency.

Parameter tuning of the classifier is conducted, with the variations in balanced accuracy documented in Fig. 4. The key parameters are:

Gamma: Dictates the shape and spread of the RBF kernel. **Number of Components:** Represents the dimensionality reduction input for the RBF sampler. A balance is sought to reduce dimensionality while preserving critical features.

Grid Image				
Accuracy for different Gammas & Num_components				
Num_components	**0.01**	**0.1**	**1**	**10**
1000	28.88	36.79	32.77	33.15
5000	34.29	**39.01**	29	36.28
10000	31.34	31.57	35.76	28.93

Fig. 4: Accuracy variation w.r.t Num of Components & Gamma

We assessed the SVM classifier on pre-processed images discussed in Section III, with the results depicted in Fig. 5. Grid images demonstrate the highest performance when used with the SVM classifier, attributable to their retention of information and spatial context from all 49 individual images in the analyzed volume. The dimensionality reduction for other pre-processed images have further obscured essential retinal features, impacting classification accuracy.

Class-wise separability metrics confirm the inferences made from t-SNE plots, with Class 2 exhibiting the highest separability score (0.56), compared to Classes 0 and 1 (0.44 and 0.46, respectively). The balanced accuracy improvement to 39-41% is potentially due to the successful classification of Class 2, as the SVM classifier effectively exploits its separability.

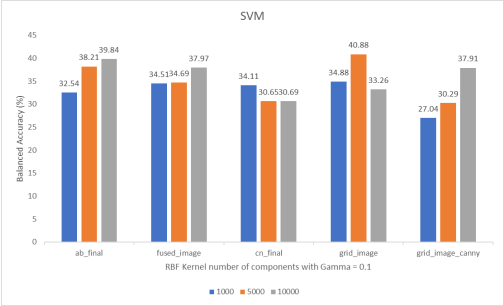


Fig. 5: Accuracy for pre-processed images-SVM

B. K-Nearest Neighbors (KNN): ***(Accuracy range: 44-45%)***

In this study, we employ the K-Nearest Neighbors (KNN) algorithm for classification, utilizing Principal Component Analysis (PCA) for dimensionality reduction. PCA effectively extracts the most significant features from high-dimensional datasets, reducing dimensions to 50 components. The reduced datasets are then passed to the KNN Classifier from the sklearn package, with the number of neighbors set to 3.

We assess our classifier on four distinct neighbor combinations, utilizing a pre-processed set of OCT images. Evaluation results, depicted in Fig. 6, reveal that the Alpha Blended images yield the highest accuracy, followed by Canny Edge images. Notably, a neighbor value of '5' demonstrates the peak accuracy.

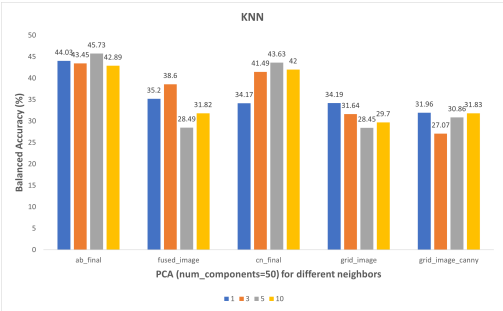


Fig. 6: Accuracy for pre-processed images-KNN

The superior performance of Alpha Blending and Canny Edge Detection in conjunction with K-NN classifiers can be attributed to the reduction of noise and suppression of irrelevant features in the transformed images. K-NN classifiers are particularly sensitive to noisy features; thus, the enhanced accuracy observed with Alpha Blended and Canny Edge fused images is a result of mitigating this sensitivity by providing cleaner, more relevant data for classification.

C. Naive Bayes: ***(Best accuracy hit: 49%)***

A Naive Bayes classifier is a probabilistic machine learning model that is based on the Bayes theorem. Naive Bayes showed quick learning capability and made decent predictions. The final Naive Bayes implementation reached a peak accuracy of 49.55% as shown in Fig. 7

When dealing with high-dimensional data, Naive Bayes can outperform other classifiers as it avoids the curse of dimensionality. Naive Bayes scales well to handle multi-class classification problems, as it requires only one model to be trained, while SVM and KNN may require multiple models or additional techniques (e.g., One-vs-Rest, One-vs-One).

Upon evaluating the Naive Bayes model using various pre-processed images as described in Section III, we found Laplacian pyramid fusion images yielding the highest accuracy of 49.55%. This superior performance can be attributed to the following factors:

Feature Independence: The fusion process introduces some degree of independence between features, benefiting the Naive Bayes classifier's assumption of conditional independence.

Noise Reduction: Fusion process suppresses noise by blending information from multiple input images, resulting in a cleaner fused image, which helps in identifying relevant patterns more effectively.

Naïve Bayes	
Input image	Accuracy
ab_final	48.33
fused_image	49.55
cn_final	34.11
grid_image	44.6
grid_image_canny	41.42

Fig. 7: Accuracy for pre-processed images-Naive Bayes

D. U-Net: ***(Best accuracy hit: 56-60%)***

Image segmentation is vital for identifying multiple objects within an image. It includes semantic segmentation, which assigns class labels to pixels for pixel-level interpretation, and instance segmentation, which distinguishes separate instances of a class. Semantic segmentation is especially useful in biomedical imaging, as it outlines organs and tissues, enabling extraction of essential diagnostic features.

U-Net Architecture: Fig. 8 shows the U-Net architecture with a contracting encoder and an expanding decoder, forming a U-shape. The encoder extracts high-level features, while the decoder produces a segmentation map. Skip connections(links between contracting and expanding path) enable capturing

high- and low-level features by transferring high-resolution feature maps between encoder and decoder.

Reasons for choosing U-Net semantic segmentation: The retinal layers in OCT images exhibit hierarchical organization. U-Net's ability to capture these relationships is attributed to the different scales of the contracting and expanding paths. U-Net also uses pixel-wise softmax cross-entropy loss function that helps counter dataset imbalance.

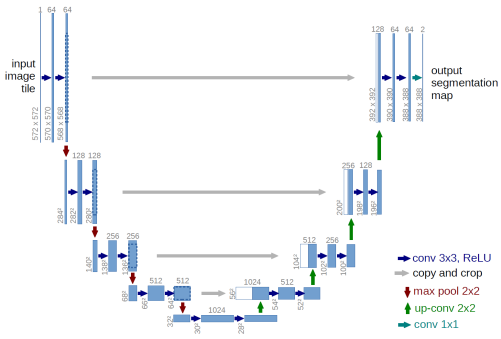


Fig. 8: U-Net Architecture

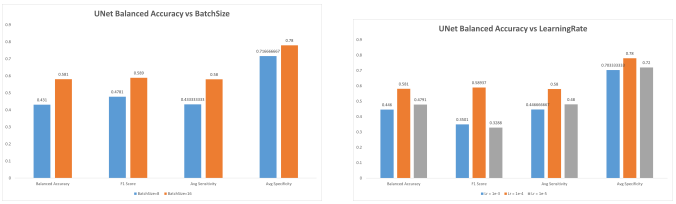
We implemented a U-Net model with PyTorch segmentation models package, featuring three input channels and output classes. Exploring various encoder architectures baed on resNet34 and EfficientNet-B0-7, we selected EfficientNet-B7 for its efficient feature extraction using MobileNetV2-like modules. Leveraging transfer learning with pretrained ImageNet weights allows the model to adapt general image features to our specific use-case.

With the help of Grid search, we found that the U-Net model achieved the peak performance (balanced accuracy: 59%, F1 score: 0.589, Sensitivity: 0.58, Specificity: 0.78) at batch size: 16, learning rate: 1e-4, and 50 epochs when trained with alpha blended canny-edge feature image as shown in Fig. 9. We stored the weights for further analysis using Grad-CAM heat maps on sample inputs. As can be seen in Fig. 10, the heat map shows the importance of relevant retinal layers in correct DRSS classification.

When we performed alpha blending on top of the canny edge detected individual OCT images, critical information like the shape and the curvature of blood vessels, vitreous, and fovea regions were enhanced. This helped the U-Net model understand and learn about the DRSS characteristics effectively.

E. Alternative methods

3D-CNN: We experimented with 3D-CNN by integrating a 2D U-Net subclass with Convolutional LSTM to process a 49-image volume. However, this approach was computationally intensive, reaching GPU memory limits quickly. We managed single-digit epochs on 32x32 images, yielding lower accuracy than 2D U-Net CNN. A 3D-CNN Jupyter notebook is available in the GitHub repo for future work.



(a) Grid Search: Batch size (b) Grid Search: Learning Rate

Fig. 9: U-Net hyperparameters Grid search

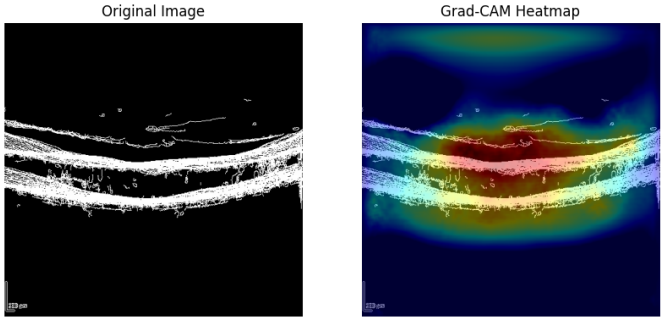


Fig. 10: U-Net extracting retinal layers

V. Results

This study examined various machine learning algorithms and feature extraction techniques to identify the optimal Diabetic Retinopathy Severity Score (DRSS) classifier for Optical Coherence Tomography (OCT) images. Performance depended on feature compatibility and algorithm learning capabilities, as summarized in Fig. 11.

The U-Net-based classifier showed the best performance, with a 59% peak accuracy. Its high sensitivity of around 0.6 indicated a strong ability to detect true positives, essential for biomedical systems. Furthermore, the U-Net classifier demonstrated notable true negative detection, especially for class 2 (the most severe level), displaying high specificity.

The Naive Bayes model ranked second with a 49% peak accuracy. Its fast learning and decent predictions on smaller datasets allowed it to outperform Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), which usually require larger datasets for optimal performance.

KNN showed a 45.73% peak accuracy with alpha-blended images. SVM reached a 41% peak accuracy using grid images, possibly due to feature preservation after dimensionality reduction via the Radial Basis Function (RBF) sampler and kernel trick.

OCT DRSS Classification Accuracy				
Feature Extraction Techniques	KNN	SVM	Naïve Bayes	U-Net
Alpha Blended image	**45.73**	38.21	48.33	36.76
Laplacian Pyramid Fused image	28.49	34.69	**49.55**	36.05
Canny Edge Result image	43.63	30.65	34.11	**59**
Grid on orig image	28.45	**40.88**	44.6	30.8
Grid on Canny Edge Result image	30.86	30.29	41.42	32.6

Fig. 11: Summary of machine learning techniques used

In conclusion, the U-Net-based classifier emerged as the most effective DRSS classifier for OCT image datasets, followed by Naive Bayes, KNN, and SVM. This study provides valuable insights into the performance of various machine learning algorithms in conjunction with different feature extraction techniques for DRSS classification, offering a foundation for future research and development in this domain.

VI. Contributions

GitHub repository: https://github.com/sandilyabvh/DRSSSeverityClassifier
Poster presentation: https://drive.google.com/file/d/1MPEJVVdHZ8d3Cn1gICUw8_39dCMwMtUU/view?usp=share_link Both authors contributed equally to this work, collaborating on all critical tasks, from pre-processing techniques to the development of the models. The distribution of tasks, based on the authors' recollection, is as follows:

1) KNN:

a) Santhana: Developing KNN model and test setup.

b) Venkata: Implementing PCA mechanism and obtaining results.

2) SVM:

a) Santhana: Developing SVClassifier test setup, Creating t-SNE plots.

b) Venkata: Implementing RBF Sampler and performing grid search.

3) Naive Bayes:

a) Santhana: Experimenting with data augmentation and obtaining results.

b) Venkata: Developing GaussianNB test setup and exploring batch size tuning.

4) U-Net:

a) Santhana: Exploring techniques to handle imbalanced data, testing different encoders and performing grid search, and obtaining perf metrics.

b) Venkata: Developing UNet model and train setup. Experimenting with data augmentations techniques and obtaining Grad-CAM results

References

[1] Harpal Singh Sandhu, Ahmed Eltanboly, Ahmed Shalaby, Robert S. Keynton, Schlomit Schaal, Ayman El-Baz; Automated Diagnosis and Grading of Diabetic Retinopathy Using Optical Coherence Tomography. Invest. Ophthalmol. Vis. Sci. 2018;59(7):3155-3160.

[2] Gahyung Ryu, Kyungmin Lee, Donggeun Park, Inhye Kim, Sang Hyun Park, Min Sagong; A Deep Learning Algorithm for Classifying Diabetic Retinopathy Using Optical Coherence Tomography Angiography. Trans. Vis. Sci. Tech. 2022;11(2):39.

[3] Elman MJ, Aiello LP, Beck RW, et al. Randomized trial evaluating ranibizumab plus prompt or deferred laser or triamcinolone plus prompt laser for diabetic macular edema. Ophthalmology. 2010; 117(6): 1064–1077.e35.

[4] Massin P, Bandello F, Garweg JG, et al. Safety and efficacy of ranibizumab in diabetic macular edema (RESOLVE study): a 12-month, randomized, controlled, double-masked, multicenter phase II study. Diabetes Care. 2010; 33(11): 2399–2405.

[5] Ruamviboonsuk P, Wongcumchang N, Surawongsin P, Panyawatananukul E, Tiensuwan M. Screening for diabetic retinopathy in rural area using single-field, digital fundus images. J Med Assoc Thai. 2005; 88(2): 176–180.