DRSS Severity Classification on OCT Images

Harish Krishnamoorthy
M.S in Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, USA
hkrishna7@gatech.edu

Disha Gulur
M.S in Electrical and Computer Engineeing
Georgia Institute of Technology
Atlanta, USA
dgulur3@gatech.edu

***Abstract*—Diabetic retinopathy is a condition which affects the eye, blood vessels and retina of individuals affected by diabetes. This project aims to build a machine learning-based classifier to classify the condition of diabetic retinopathy using severity scores. Various supervised learning approaches are used and thorough performance analysis is done to determine the severity of the disease. Incorporating non-human based AI tools for medical detection and treatment is always a challenging task and this project aims to evaluate 4 different types of classifiers and evaluate their efficacy.**

I. Introduction

In this project we are tasked to classify different biomedical images of the eye into corresponding Diabetic Retinopathy Severity Scale (DRSS) classes. We have OCT (Optical Coherence Tomography) scans and Fundus images both of which are gray scale images. Along with that we have corresponding metadata for each of the images which contains details about different biomarkers like frame number, diabetes year, BMI, CST, Week number of the treatment etc. In this project four different classifiers are used which are namely Naïve Bayes, Linear Regression, Support Vector Machines (SVM) and Convolutional Neural Networks(CNNs). We have experimented with the hyperparameters and various pre-trained architectures involved in the above classifiers to deduce the highly performing ones and also attempted to fuse these classifiers for better performance and feature extraction.

Before delving into the design and usage of model we need to analyze the dataset. There are 8 different DRSS scores available [35,43,47,53,61,65,71,85] and these 8 scores are further being classified into 3 classes namely [0,1,2]. The number of data available in each of the target classes is imbalanced [Class 0: 7788, Class 1: 11760, Class 2: 4704]. With imbalanced data we need to make sure that the classifier model is not biased towards any specific class. This observation is kept in mind in each of our classifier method to modify input dataset, training methods, loss functions and performance evaluators. We have two types of inputs provided for the classification tasks. The first one is the gray scale images through OCT and fundus scans of eye giving us pixel values in a single channel. The second one is the 16 different biomarkers provided in a text based format. Techniques have been incorporated in this paper to efficiently fuse these biomarkers along with the training data.

II. Basic Classifiers

Our very first experiment was to pass the input data through the basic classifiers including Naive Bayes, Logistic Regression and SVM classifiers. The Normal Accuracy, Balanced Accuracy, Precision, Recall and F1 Score was recorded for all three methods.

A. Image Data

The raw image data of size 224x224 is used. The images are flattened to produce vectors of size [1x50176]. These vectors are then concatenated and passed through the various classifiers. The SVM classifier failed to converge over such a huge dataset. Due to this PCA - a data reduction technique was needed to be used. The dimensions of the matrix was reduced to result in a feature size of 100. The results are depicted in table 1.

Classifier	Accuracy	Balanced Acc.	Precision	Recall	F1-Score
Naive Bayes	0.49	0.46	0.47	0.47	0.469
Logistic Regression	0.42	0.36	0.35	0.36	0.36
SVM	0.45	0.34	0.35	0.3	0.32

TABLE I
Performance with Image Data.

B. Image Data + Biomarkers

In this experiment we combine the pixels of the image with the biomarkers that are available. These biomarkers are added only during training and not during testing. During testing the image data is used as is. The biomarkers provide valuable information from which the ML algorithm can draw patterns and derive useful information. The biomarkers are represented as 15x1 vectors, but to use them for image analysis, they are resized to match the dimensions of the image, which is 224x224. The biomarker vector is used as a template for the center of the image, and the rest of the image is padded with zeros to match the desired dimensions. By doing so, we can successfully add the image and the biomarker information. Figure 1 is a depiction of how this addition is done. The performance evaluation metrics are shown in Table 2. While running SVM, the dimensions was again reduced to 100.

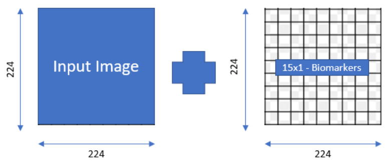


Fig. 1. Image+Meta Data construction

Classifier	Accuracy	Balanced Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.49	0.47	0.47	0.47	0.47
Logistic Regression	0.47	0.39	0.40	0.39	0.39
SVM	0.46	0.399	0.4	0.399	0.385

TABLE II
PERFORMANCE WITH IMAGE DATA WITH BIOMARKERS.

C. PCA with Image Data and Biomarkers

The image that consists of both the image pixels and the biomarkers is reduced to lower dimensions - 100 features using PCA.This was done in an attempt to extract the essential features only and remove any redundancy/noise form the data. Patterns are also easily identifiable when the data set is reduced.

Classifier	Accuracy	Balanced Acc.	Precision	Recall	F1-Score
Naive Bayes	0.458	0.381	0.368	0.381	0.358
Logistic Regression	0.437	0.408	0.408	0.408	0.397

TABLE III
PERFORMANCE WITH PCA.

Observations and Analysis:

- Naive Bayes Outperforms both Logistic Regression and SVM as the accuracy and balanced accuracy is higher in all cases.
  - This could be because Naive Bayes works better with imbalanced data.
  - Naive Bayes also assumes feature independance which could work in our favour as pixels of an image are independant of each other for the most part.
- Almost all performance metrics slightly improve when biomarkers are included.
  - Biomarkers carry essential information such as age of the patient, week of visit, Diabetes Type etc.
  - The improvement is not massive as the size of biomarkers is extremely low compared to the size of the image vector.
  - We see an improvement in the F1 score as well indicating lower number of false positives and false negatives.
- Reducing the dimensionality does not help with accuracy.
  - The important features possibly got masked due to a dimensionality reduction.

III. Convolution Neural Networks

A. Loss Functions

Every neural network has a loss function that is used to train the neural network and optimize the weights. Using the correct loss function is important especially when the data is imbalanced. Through the course of this section we have experimented with different types of losses.

- Cross Entropy Loss : This loss function leverages the probability distrubtion of the predicted outputs and the true labels. It converges easily and is also easy to optimize. The Cross entropy loss is given by the formula $\mathcal{L}_{CE} = -\sum_{i=1}^{n} y_i \log(p_i)$ where y_i is the class label, and p_i is the predicted probability of the sample belonging to the positive class.

- Weighted Cross Entropy Loss: This is derived from the cross entropy loss but it takes into account imbalance in the training data. It provides a higher weight to samples that are under-represented in the dataset. The formula that was used to calculate weights for this project is given by : $weight\_class = \frac{total\_samples}{samples\_in\_class \times (len\_class)}$ The overall equation of weighted loss is given by: $\mathcal{L}_{\text{weighted}} = -\frac{1}{n}\sum_{i=1}^{n} w_i y_i \log(p_i)$ The loss vector that was used is as follows - [1.0380071905495634, 0.6874149659863945, 1.7185374149659864]

- Focal Loss: This is also used primarily for imbalanced data. As the name suggests it adds a modulating term to cross entropy loss in order to focus on the misclassified samples and ultimately improve the training an optimization process. There are two hyper parameters - weights and gamma. Gamma is the focusing parameter - set to 2 and alpha is the weight assigned to each class. We calculated alpha the same as described in the previous equation. The Equation for focal loss is: $\mathcal{L}_{\text{FL}} = -\frac{1}{n}\sum_{i=1}^{n} w_i (1-p_i)^{\gamma} y_i \log(p_i)$

B. Pretrained models and Classifiers

Using traditional classifiers like Naïve Bayes, SVM and Logistic Regression would lead to exponentially increased computations especially when we are simply vectorizing the images. To approach this problem we use a pre-trained Resnet model which will be used to extract features. The output of this model is fed back to one of these classifiers. Our motivation behind this approach was to get better results by extracting essential features through the neural network (instead of using PCA like before) and also reduce the computation time. Alexnet and Resnet were used to perform this experiment with a transfer learning approach. SVM was not used for this expirement due to the exceptionally long time it took to converge to a solution even after PCA. Image data and meta data were combined as shown in Figure 1 and the model that was used is explained in Figure 2. The results are depicted in

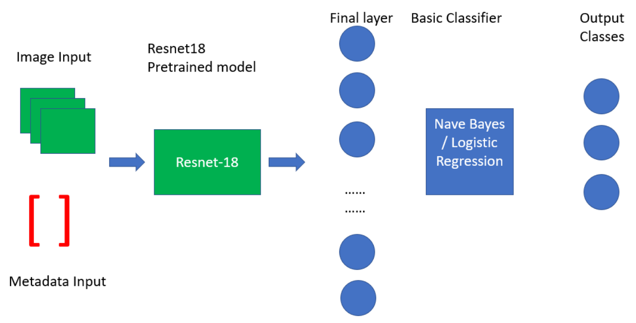


Fig. 2. Using a pretrained network with a classifer

	Accuracy	Balanced Acc.	Precision	Recall	F1-Score
Naïve Bayes + Alexnet	0.389	0.364	0.371	0.355	0.362
Naïve Bayes + Resnet	0.412	0.408	0.397	0.382	0.389
Logistic Regression + Alexnet	0.376	0.355	0.325	0.346	0.335
Logistic Regression + Resnet	0.40	0.394	0.402	0.394	0.397

TABLE IV
PERFORMANCE METRICS FOR A COMBINATION OF BOTH CNN AND CLASSIFIER.

Observations and Analysis:

- Reducing the feature dimensions using a neural network reduces the accuracy from the previous case where the entire image was passed through a classifier.
  - Using all pixel values is probably a more efficient way of classifying images
- Resnet18 outperforms Alexnet in both cases
  - Resnet is able to learn features more efficiently due to its deeper architecture and complex structure.
  - Resnet has also been trained on larger, more diverse datasets.
- There is potential to improve accuracy by training these architectures from scratch.
  - A pretrained model uses cross entropy loss and this is not in our control. Given that our dataset is imbalanced using weighted loss might be more advantageous

C. *Training existing CNN Architectures*

Our next approach was to train existing neural networks from scratch. Here we use two different loss functions - Cross entropy loss and weighted cross entropy loss. We take the existing image, integrate it with the metadata as shown in Figure 1 and train it with the neural network. Adam optimization technique is used and the learning rate is set to 1e-3. The network is trained over 50 epochs.

Classifier	Case	Accuracy	Balanced Acc.	Precision	Recall	F1-Score
Alexnet	Cross Entropy Loss	0.391	0.382	0.376	0.362	0.368
Alexnet	Weighted Loss	0.412	0.405	0.392	0.387	0.389
Resnet	Cross Entropy Loss	0.407	0.399	0.398	0.36	0.378
Resnet	Weighted Loss	0.412	0.405	0.38	0.401	0.390

TABLE V
PERFORMANCE METRICS FOR A FULLY TRAINED NETWORK

Observations and Analysis:

- The performance metrics with weighted cross entropy were found to be better than when normal cross entropy loss function was used.
  - The imbalance in the data requires a cost function that accounts for the different weights of each class
- Resnet outperforms Alexnet just like the previous implementation.
- Training the network from scratch helps with the performance of the neural network as the most optimum weights are found.
- Given the better performance of Resnet it was used in the last experiment we conducted to further improve performance.

D. *Siemese Networks*

To address the challenge of multimodal data we used a Siamese network which is essentially a combination of 2 neural networks. It is demonstrated in Figure 3. The two kinds of inputs, namely the gray scaled images and the metadata are passed in 2 separate neural networks whose final layer is fused together. For images we used the pretrained models of Resnet18 where tweaks were made to output layer to support just three classes. The metadata vector is passed through a fully connected neural network with 64 and 128 hidden neurons in the corresponding layers with RELU activation function. The outputs from both neural networks are combined and trained through a linear layer to produce class output using three neurons.

In order to account for imbalance in data we attempt to modify the input dataset. To resolve this, we can either increase or decrease the dataset by using semi-supervised learning techniques. This is explained in the following section.

1) Input Data Modification: From the dataset we used two approaches first to decrease the samples of the dominating class by using random dropouts and entropy-based reduction to have a reduced size of 1000 and 4000 entries per class. In case of entropy-based reduction we tried to extract entropy of each entry and preserve the ones with higher values to provide the most features from the reduced dataset. In case of increasing dataset size of a specific class we again used entropy as a parameter to duplicate those data entries. Here

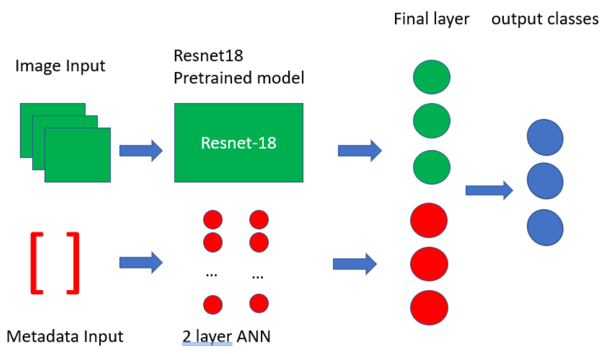


Fig. 3. Siemese Network

the entropy is being calculated using the metadata present in the training dataset. This architecture is depcited in Figure 4.

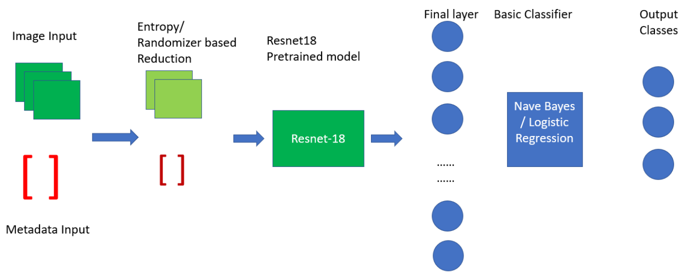


Fig. 4. Semi Supervised Learning

	Accuracy	Balanced Acc.	Precision	Recall	F1-Score
Baseline	0.4259	0.3758	0.3512	0.3629	0.356
WL	0.4212	0.3842	0.3764	0.362	0.369
FL	0.4357	0.4098	0.3925	0.3719	0.381
RSR (4000)	0.4333	0.3971	0.3982	0.3729	0.385
RSR (1000)	0.3673	0.3229	0.3648	0.3517	0.358
EBDR	0.4049	0.3954	0.4017	0.3888	0.395
EBO	0.474	0.4076	0.4098	0.4005	0.405
PCA	0.4445	0.3829	0.3682	0.3792	0.373

Abbreviation	Full Form
WL	Weighted Loss
FL	Focal Loss
RSR (size per class)	Random Sampling Reduction
EBDC	Entropy Based Data Reduction

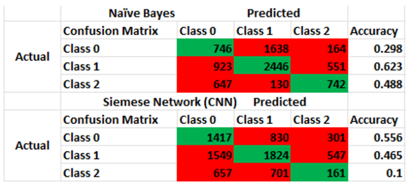
Observations and Analysis:

- With 4000 samples we observe similar accuracy but increased precision and F1 score for entropy-based sampling compared to random sampling.
  - This is because we are using the most informative samples in the former.
  - The difference in accuracy is minimal due to large sampling size.
- Between weighted cross entropy and focal loss, we observe a minor improvement in focal loss-based model
  - A modulation factor is used for weight computation in the latter.
- Overall, With Siemese Networks observe an improvement in balanced accuracy among the various CNN based models.

This is because we are using the most informative samples in the former. The difference is minimal due to large sampling size. Among weighted cross entropy and focal loss we could see a minor improvement in focal loss based model due to the fact that a modulation factor is used for weight computation in the latter. Hence the semi-supervised learning and focal loss based model were giving performance close to Naïve Bayes in the above scenario. Overall, With Siemese Networks we are able to see an improvement in balanced accuracy among the varous CNN based models.

IV. Confusion Matrix

The confusion matrices were calculated for all cases but only the basic Naïve Bayes implementation and Focal Loss based Siemese Network implementation has been depicted. Figure 5 depicts a confusion matrix of Naïve Bayes which denotes least accuracy for Class 0 where most samples are mispredicted as Class 1. Even though the accuracy was similar for weighted Siemese Network we could see that the accuracy of Class 0 has significantly improved but the accuracy of Class 2 has gone down rapidly.



	Naïve Bayes	Predicted			
	Confusion Matrix	Class 0	Class 1	Class 2	Accuracy
Actual	Class 0	746	1638	164	0.298
	Class 1	923	2446	551	0.623
	Class 2	647	130	742	0.488
	Siemese Network (CNN)	Predicted			
	Confusion Matrix	Class 0	Class 1	Class 2	Accuracy
Actual	Class 0	1417	830	301	0.556
	Class 1	1549	1824	547	0.465
	Class 2	657	701	161	0.1

Fig. 5. Confusion Matrices

V. Summary and Conclusion

In this project we tried multiple classifiers and their combination to evaluate their efficacy. Even though we were able to see differences among the cases, overall, it was very challenging for us to achieve accuracy even in higher 40s. The basic Naïve Bayes implementation and Focal Loss based Siemese Network implementation techniques provide similar accuracy values with the best overall better performing results.

To expand this project we intend to explore efficient active learning techniques along than semi-supervised learnings. We also want to use efficient feature extractors like HOG (Histogram of Oriented Gradients), SIFT (Scale-Invariant Feature Transform), SURF (Speeded Up Robust Features) etc. rather than only using already existing CNNs like Alexnet or Resnet and fuse these techniques to improve the efficiency of the model.

References

[1] M. Prabhushankar, K. Kokilepersaud*, Y. Logan*, S. Trejo Corona*, G. AlRegib, C. Wykoff, ”OLIVES Dataset: Ophthalmic Labels for Investigating Visual Eye Semantics,” in Advances in Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks, New Orleans, LA,, Nov. 29 - Dec. 1 2022.

[2] Saad M Khan, Xiaoxuan Liu, Siddharth Nath, Edward Korot, Livia Faes, Siegfried K Wagner, Pearse A Keane, Neil J Sebire, Matthew J Burton, and Alastair K Denniston, “A global review of publicly available datasets for ophthalmological imaging: barriers to access, usability, and generalisability,” The Lancet Digital Health, vol. 3, no. 1, pp. e51–e66, 2021

[3] Y. Logan, R. Benkert, A. Mustafa, G. Kwon, G. AlRegib, ”Patient Aware Active Learning for Fine-Grained OCT Classification,” in IEEE International Conference on Image Processing (ICIP), Bordeaux, France, Oct. 16-19 2022

[4] Y. Logan, K. Kokilepersaud, G. Kwon and G. AlRegib, C. Wykoff, H. Yu, ”Multi-Modal Learning Using Physicians Diagnostics for Optical Coherence Tomography Classification,” in IEEE International Symposium on Biomedical Imaging (ISBI), Kalkota, India, Jan. 7 2022

VI. Appendix

Click on the below links to find the Github Repository, Presentation and Presentation Recording.

- Github Repo Link
- Presentation Link
- Presentation Recording Link