DRSS Severity Classification on OCT images

Harish Krishnamoorthy
gtID:903664559
MS ECE

Disha Rakesh Gulur
gtID:903694666
MS ECE

Georgia Tech

Problem Statement

- Diabetic retinopathy is a condition which affects the eye, blood vessels and retina of individuals affected by diabetes.

- Build a machine learning model to determine the severity of this disease

Georgia Tech

Objectives

- Process the training and testing data for image classification.
  - normalize the data
  - convert grayscale images to RGB format.
- Use basic ML classifiers – Naïve Bayes, Logistic Regression and SVM to perform classification on the processed data.
- Use pre-trained and non-pre-trained neural networks to perform classification.
- Resolve data imbalance and multimodality challenges.
- Complete the performance evaluation and modify the architecture based on the metrics.

Method 1 - Basic ML Classifier – Naïve Bayes, Logistic Regression, SVM

- The ML classifier was used to predict labels on
  - raw image data
    - Size of the image – 224x224
  - Combination of image data and biomarkers
    - The 15x1 biomarker vectors are resized to 224x224 to match the image dimensions.
    - The vector serves as a template for the center of the image, with the rest padded with zeros to match the desired dimensions.
    - The biomarkers are added only during training and not during testing.
  - PCA with Image data + biomarkers
    - The 224x224 image data was reduced to a 10x10 image and then passed through the classifier

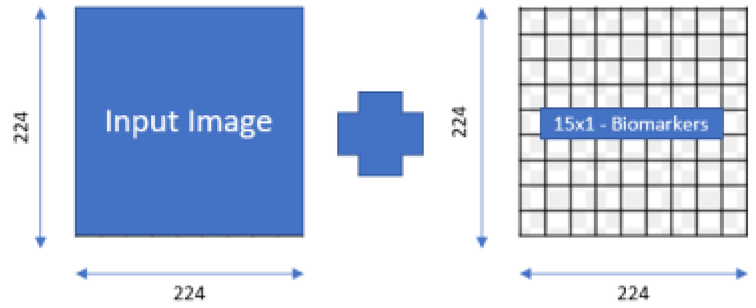

Fig 1: Input + Metadata implementation

Observations and Analysis

- Naive Bayes Outperforms both Logistic Regression and SVM
  - Naive Bayes works better with imbalanced data
- Performance metrics slightly improve when biomarkers are included.
  - Biomarkers carry essential information
  - Improvement is limited due to the dominance of pixel values over size of biomarkers
- Reducing the dimensionality leads to poorer performance.
  - PCA led to important getting masked

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.

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.

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.

Georgia Tech

Method 2 - Pretrained Models + Basic Classifiers

- Combine the input image data and biomarkers as described before.
- Use a pre-trained Alexnet, Resnet18 model to extract essential features from this image.
- Pass these features through Naïve Bayes and Logistic Regression Classifiers.

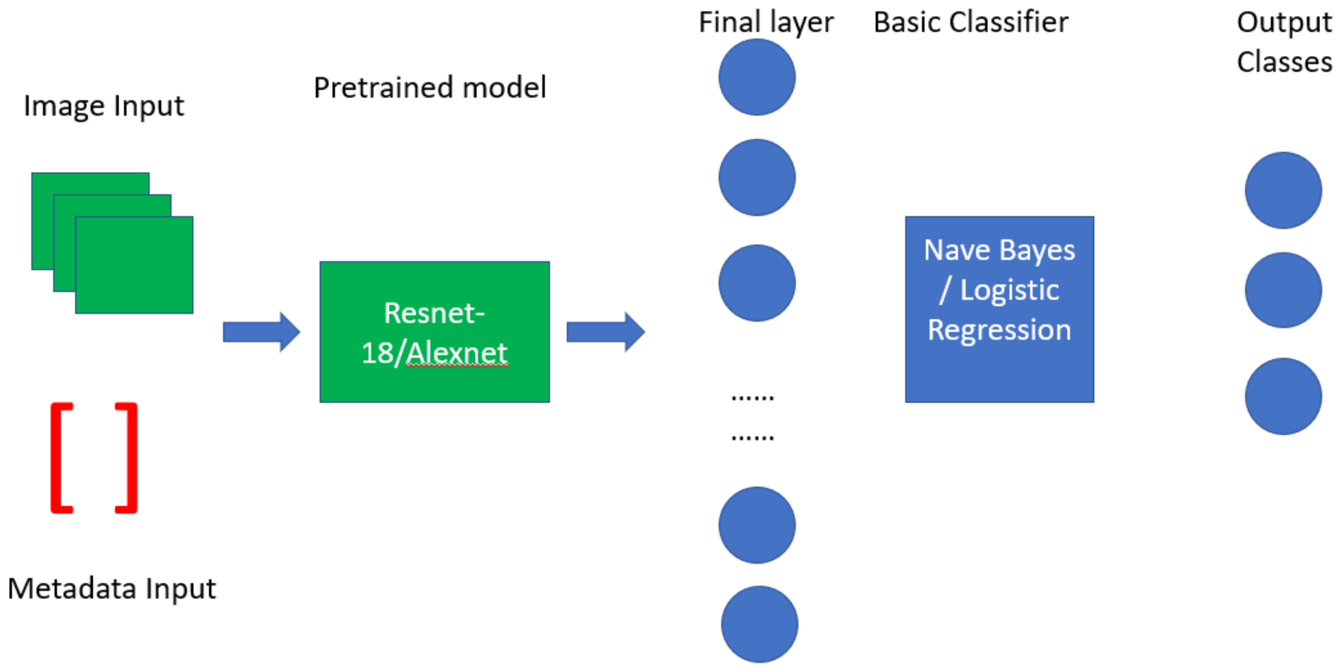

Fig 2: Pre-trained CNNs for feature extraction

Observations and Analysis

- The accuracy has reduced 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.
  - Use different types of Loss functions to account for imbalance in data

	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.

Method 3 - Training the Model

- Train existing predefined neural networks from scratch.
- Train with two different loss functions
  - Cross entropy loss and weighted cross entropy loss.
  - Weighted cross entropy loss provides a higher weight to samples that are under-represented in the dataset.

$$weight\_class = \frac{total\_samples}{samples\_in\_class \times (len\_class)}$$

- Input Image is integrated with biomarkers.
- Adam optimization technique is used. Learning rate is set to 1e-3.
- The networks are trained over 50 epochs.

Georgia Tech

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.

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

Georgia Tech

Method 4 – Using Siemese Networks

- This network is a combination of 2 neural networks for 2 kinds of data.
- Images are passed through a pretrained Resnet18 network
  - Output layer is tweaked to have just 3 classes.
- The metadata is passed through a fully connected neural network:
  - Two hidden layers consisting of 64 and 128 hidden neurons with RELU activation function.
- The outputs from both neural networks are combined and trained through a linear layer to produce a three-class prediction

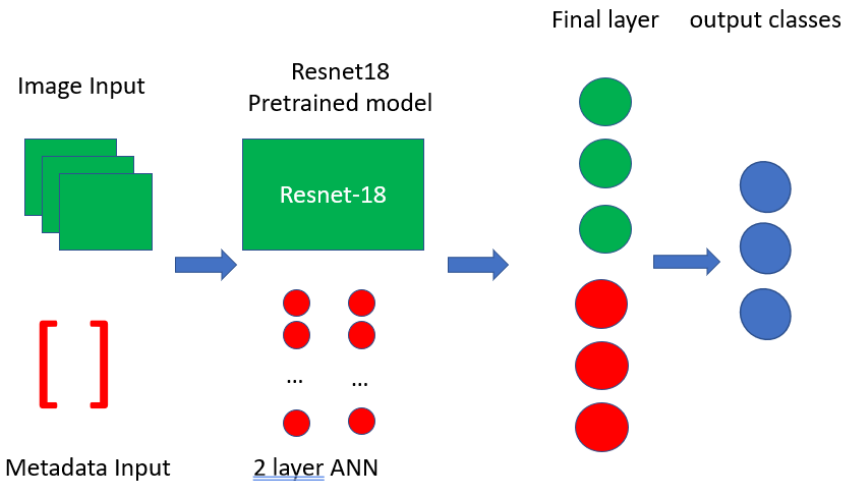

Fig 3: Siemese Network Design for inputting metadata and images

Semi-Supervised Learning

- To account for imbalance in data we attempt to modify the input dataset using 2 approaches.
- Entropy-based reduction to have a reduced size of 1000 and 4000 entries per class.
  - Extract entropy of each entry and preserve the ones with higher values
- Decrease the samples of the dominating class by using random dropouts.
  - Use entropy as a parameter to duplicate those data entries.
- Here the entropy is being calculated using the metadata present in the training dataset.

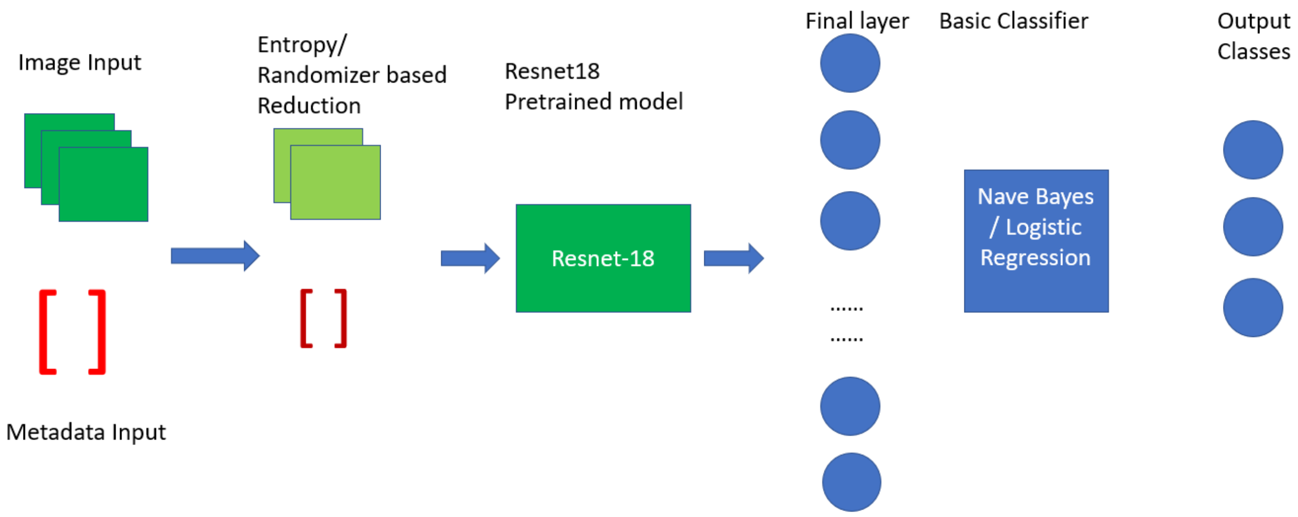

Fig 4: Siamese network is fused with semi-supervised approach

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.

	Accuracy	Balanced Accuracy	Precision	Recall	F1-Score
Baseline Network	0.4259	0.3758	0.365	0.376	0.363
Weighted Loss	0.3944	0.3302	0.325	0.33	0.327
Focal Loss	0.4357	0.3496	0.343	0.35	0.342
Random Sampling Reduction (4000 per class)	0.4333	0.3482	0.337	0.348	0.339
Random Sampling Reduction (1000 per class)	0.3673	0.3229	0.287	0.323	0.293
Entropy Based Data Reduction	0.4049	0.357	0.348	0.357	0.347
Entropy Based Oversampling	0.4123	0.3355	0.325	0.335	0.328
PCA	0.4445	0.3669	0.343	0.367	0.351

Table VI . Results for Siamese Networks

Georgia Tech

Confusion Matrix

- The confusion matrices are calculated for the basic Naïve Bayes implementation and Focal Loss based Siemese Network implementation.
- The figure depicts a confusion matrix of Naïve Bayes which denotes least accuracy for Class 0 where most samples are mis-predicted 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 6. Confusion Matrix for Naïve Bayes and Siemese Networks

Conclusion and Future Scope

- The basic Naïve Bayes implementation and Focal Loss based Siemese Network implementation techniques provide similar accuracy values with the best overall better performing results.
- In the future we intend to explore efficient active learning techniques
- We can also 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

Thank You

Georgia Tech