Diabetic Retinopathy Detection

Adarsh Jois
Courant Institute of Mathematical Sciences
New York, New York
avj225@nyu.edu

Jasmine Hsu
Courant Institute of Mathematical Sciences
New York, New York
jch550@nyu.edu

Kshitiz Sethia
Courant Institute of Mathematical Sciences
New York, New York
ks3839@nyu.edu

***Abstract*—**

Diabetic retinopathy is a degenerative disease that causes blindness in Diabetics. Its detection methods are only through manual detection by trained clinicians, which induces a high cost as well as delayed response time. We attempt to facilitate this detection by trying to produce a model that is able to predict the severity of the disease by its corresponding medical level. Our methodology uses convolutional neural networks; the final output of the algorithm predicts a severity level to images of retinas that contain this disease. We got a baseline result of 30.35% accuracy using Random Forests. The final output with ConvNets trained on wavelet edges gave an accuracy of 78.35%.

Keywords—diabetic retinopathy, convolutional neural networks, multiclass classification, computer vision, class balancing, …

I. Introduction

Diabetic retinopathy is a debilitating and degenerative disease that causes vision degradation. It is an ocular manifestation from diabetes and is the leading cause of blindness for people aged 20-64 years. Diagnosis of this disease is detected during eye examination, and includes a variety of visual tests that require complex imaging techniques and systematic programs. These examinations cost both financially and in time, which has led to vigorous industry research in improvements of detection techniques. Our goal for this project is to apply novel machine learning algorithms and methods to improve the automated method of DR screening.

Figure 1. Example of normal vision.

Figure 2. The same view with diabetic retinopathy.

II. Dataset

The dataset that we are working with, supplied by Kaggle, is provided by EyePACS, a retinopathy screening platform. The format of the dataset is in the form of raw jpegs, and includes an image of both the left and right eye (fundus). The training set has roughly 35,000 images and the test set has roughly 53,000 images. The total size of the dataset is around 70GB. The images have a variation of resolutions and thus preprocessing steps must be taken before we input it into the model, which will be discussed in a proceeding section.

The fundus images are taken under a variety of imaging conditions and labeled with a subject id. For each subject's left and right eye, a clinician has rated the presence of symptoms that attribute to diabetic retinopathy from a scale of 0 to 4. The rating runs from no DR (0), mild (1), moderate (2), severe (3), and proliferative DR (4). The noise generated by the variation of camera type and model causes noise and image artifacts such as underexposure, overexposure, or blur.

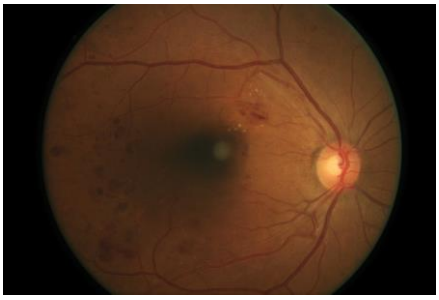


Figure 3. Left eye sample from our dataset.

Figure 4. Right eye sample from our dataset

III. Feature Analysis

In terms of feature engineering, we have studied intensively on previous research from industry on this particular disease, as well as consulted with professional eye doctors. We summarize some of our findings below, which will be used in terms of creating a preprocessing pipeline that best emphasizes the important features for detecting diabetic retinopathy.

The eye care professional will look at the retina for early signs of the disease, such as:

- leaking blood vessels
- retinal swelling (such as macular edema)
- pale, fatty deposits on the retina (exudates) – signs of leaking blood vessels,
- damaged nerve tissue (neuropathy)
- any changes in the blood vessels.

IV. Preprocessing

The following methods are what we selected as a set of preprocessing methods for the dataset. The selection of preprocessing methods for this framework is a challenging task.

Most images obtained by medical imaging systems are noisy or do not possess sufficient contrast. These factors make it difficult to obtain regions of interest in the image. The methods below are meant to eliminate some of these problems and obtain a relatively well contrasted image that enhances the visibility of the ROI's that are indicative of the symptoms of DR.

Based on the machine learning method used subsets of these preprocessing methods were used. A sample image is also presented as an example of the preprocessing (Figures 6-8). The dataset on which the preprocessing method was tested was the MESSIDOR database.

Enhancement of the Retinal Image

The images had a large amount of padding that needed to be cropped closer to the image of the retina. This serves the purpose of reducing the image size removing most of the padding and thereby the dimensionality of the input. The image was then contrast adjusted using uniform histogram equalization in the manner presented by the imagemagick library. [7]

This proved very useful in greatly enhancing the visual contrast of the eye itself. Vascular characteristics of the eye were enhanced since they were darker in the original image. Final RGB image was subjected to a battery of other preprocessing ideas drawn from image processing and medical literature.

Figure 5. Raw Image

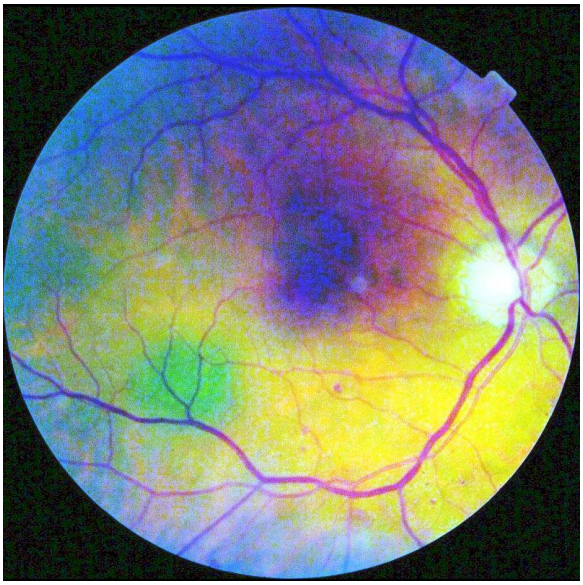


Figure 6. Enhanced image (equalization and contrast adjustment).

The image was then split into its constituent channels. Several papers proposed that the green channel was the most useful in identifying the vasculature of the eye and of regions that possessed microaneurysms due to their contrast level. We decided to follow this approach as we visually verified that the red and blue channels were very noisy and the green channel contained almost all of the smaller features of the image. This was also done to reduce the overall size of the dataset to be fed into the ConvNet.

CONTRAST ADAPTIVE HISTOGRAM EQUALIZATION

Every image was subjected to contrast adaptive histogram equalization over each channel. This is a popular technique used in biomedical image processing. The image is split into disjoint regions and in each regions a local histogram equalisation is applied. The boundaries between the regions are then eliminated using bilinear interpolation. It enhances the contrast between regions much like the human eye would perceive it. This was applied to each channel in order to enhance the contrast of channel specific features. [8]

MORPHOLOGICAL FEATURES

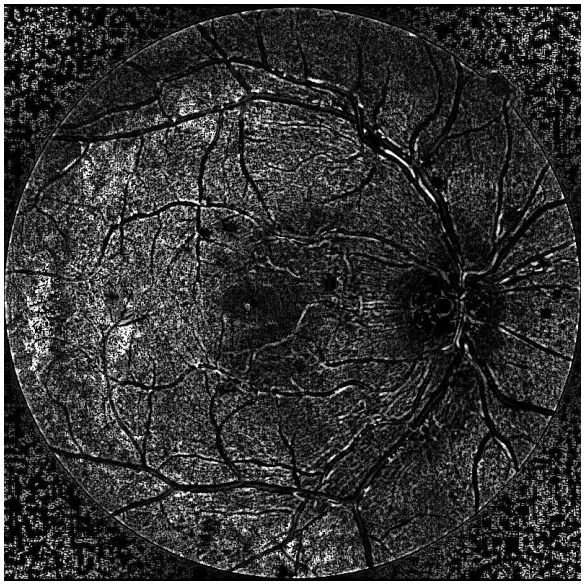


Figure 7. Morphological transformation.

The image was then subjected to a series of morphological transformations, image contrast enhancements and subtractions to extract vasculature and microaneurysms. These transformations do a series of enhancements and smoothing operations to the image that help distinguishing these features from the noisy background of the retinal image. The morphological operations are implemented as in the paper Li Yun, et al.

Shortly the steps are as below:

1) Morphological opening with a structuring element which is a disk of radius 10.
2) Morphological opening with a structuring element which is a disk of radius 18.
3) Morphological opening with a structuring element which is a diamond of size 3.
4) Subtract the opened image from the original image after each opening.
5) Adjust the contrast of the image after each morphological opening such that only 1% of data is saturated at low and high intensities.

The reasons for each of these steps are as follows, the first two disc openings and subtractions help remove the noisy characteristics of the retinal background. The diamond opening and subtraction step, adds increased sharpness to edges and boundaries in the image. [17]

BINARIZATION

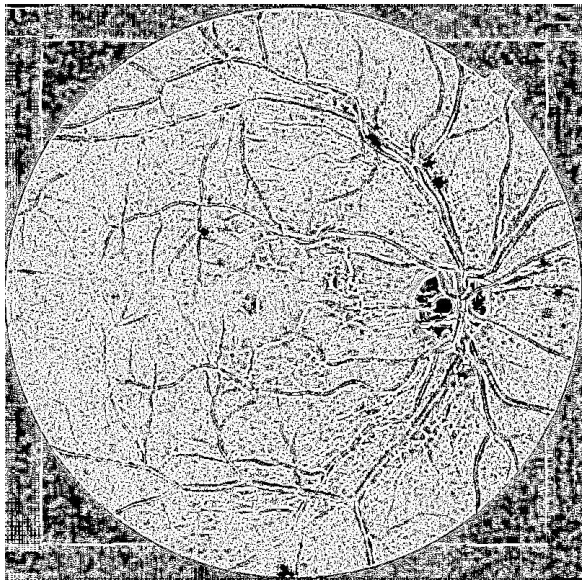


Figure 8. Binarization.

The morphological image was thresholded to obtain a simple binarized representation of the image. A simple thresholding process as described inLi Yun, Et. al. A threshold was applied on the 25th percentage of the gray intensities of the image. 25% of the lower gray intensities are discarded. This was set empirically. [17]

WAVELET FEATURES

Wavelets are useful to perform multiresolution analysis, which can help get rid of some of the noise that get do not get fully suppressed by the morphological transformations.

The wavelet of choice was the biorthogonal wavelet because they have properties that are good for image compression as they provide a good compression ratio, which means that only a few coefficients of the wavelet transformed image are useful and the rest can be discarded, as they contain noisy or lower resolution characteristics which might not be of interest to the problem. [18]

WAVELET APPROXIMATION COEFFICIENT

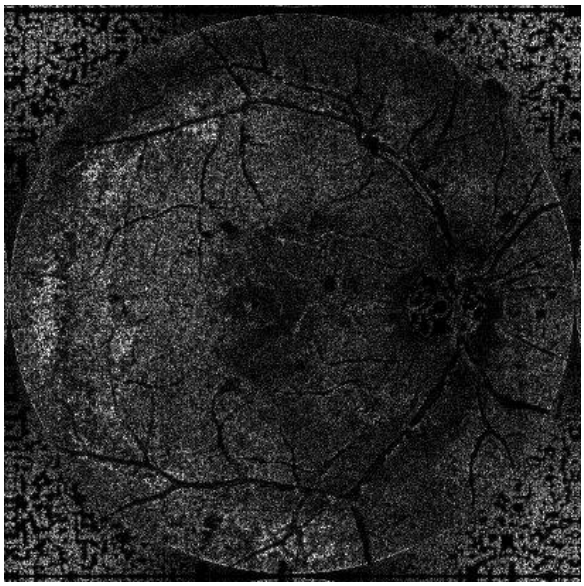


Figure 9. Wavelet approximation. (contrast-adjusted)

The wavelet coefficients of the morphologically transformed data set was extracted. We extracted the approximation coefficients of the discrete wavelet transform over the image. That gives us the set of objects in the image that are of the highest order of resolution, getting rid of granular noisy features. We refer to the model built using these wavelet coefficients as the wavelet-largest model. [18]

WAVELET EDGE DETECTION

The sum of the horizontal and vertical coefficients was also used as this gave us the borders around some of the entities in the eye, such as any vasculature or aneurysms. This model is referred to as the wavelet-edge model.

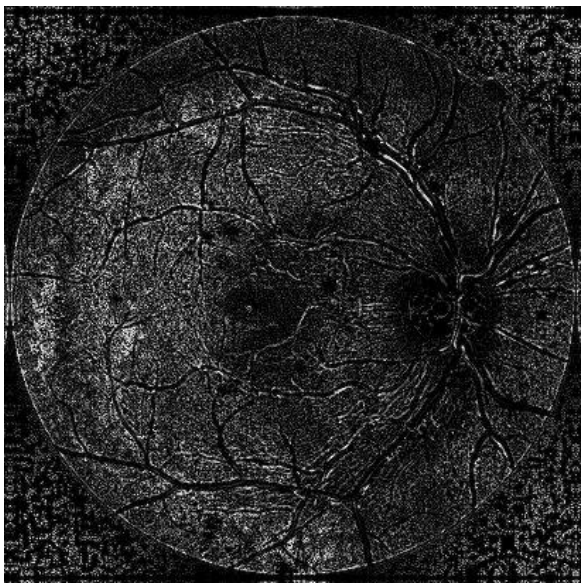


Figure 10. Wavelet edge. (contrast-adjusted)

V. METHODOLOGY

The dataset we received was extremely unbalanced. As most learning algorithms do not work well with unbalanced data, we had to take care of balancing the dataset. The initial class distribution of our dataset was as follows.

Number	Class	Percent
25810	0 – No DR	73%
2443	1 – Mild DR	7%
5292	2 – Moderate DR	15%
873	3 – Severe DR	2%
708	4 – Proliferative DR	2%

Figure 11. The class distribution of the dataset.

Our main methodology after balancing out the class distribution in the dataset was to run our dataset through a convolutional neural network. Due to computational limits, we were unable to run with entire RGB channels and/or all four preprocessed images types into one neural network. This led us to develop a framework where we ran four separate neural networks on one channel of each preprocessed image types. We chose the green channel based on literature studies that the green channel alone provided enough information for detection. [14] Of course, should we have enough computatianal power and time in the future, the supplemental red and blue channels can only improve accuracy. We will address these issues more in our future work.

Each of the models were able to produce their own predictions. However, the model learned that by predicting one class more likely than another, it achieved a higher global accuracy at the cost of class accuracy. Thus, we ended up with models that chose to only predict the middle class, which was level three. In order to alleviate this, we chose to follow up the model predictions by inputing the model weights into model averaging or model boosting. Instead of the our neural network simply outputing the argmax of the five negative log likelihood values, we let a multiclass classifer learn an alternate argmax function. We will now discuss the details of how we achieved class balancing, the specifics of convolutional network and the boosting strategy.

CLASS BALANCING

Although it is stated that level of retinopathy can be different in both the eyes of a patient, the causing factor indicates that there could be a correlation. We include below the matrix of left and right eye retinopathy.

```
[[12155   407   295     3    11]
 [  435   600   171     2     4]
 [  336   222  1998    96    50]
 [    3     1    87   307    27]
 [   10     1    39    40   263]]
```

Figure 12. Matrix with disease levels for left and right eye

Elements on the diagonal have same level in both eyes, and difference increases as we move away from the diagonal. We see that most of the elements are on the diagonal, therefore in a model with independent sampling, both the left and right eye should be in the same dataset (train/test/validation).

Our approach was to oversample in order to create balanced datasets:

a) Bin the eyes according to level of infection in left and right eye. (Similar to the matrix in Figure 6.) Each row+column entry is one class.
b) Combine bins with size <=30, so that random sampling from them gives less repetition.
c) Decide the training, validation and test data sizes.
d) Sample with replacement in Round Robin fashion from all classes. Keep track of all samples which have been assigned to training dataset
e) Put remaining samples into validation and test dataset. These datasets have not been kept unbalanced.

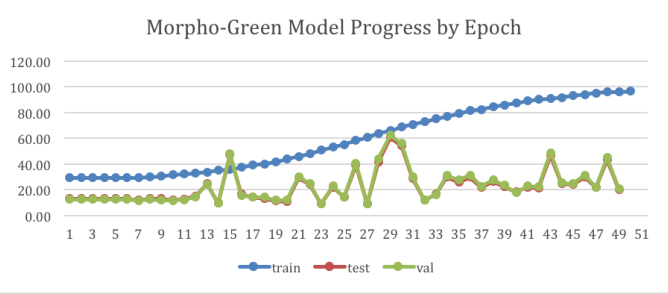
After class balancing, we ended up with a training set of 10,029 images, a validation set size of 23,742 images, and testing set size of 6102.

CONVNET ARCHITECTURE

Based on the four preprocessing methods that we used, we developed four models per processing method. Thus we ended up with four separate models that produced the weight vectors later used in multiclass classification.

Our architecture was a simple 2-layer neural network. We used the negative log likelihood loss as our criterion and gradient descent as our optimization method. Through our validation set, we tuned our model to have certain hyperparameters: a mini-batch size = 16, momentum = 0.8, dropout = 0.5, and learning rate of 1e-3. As an example, we show a graph of the validation and training accuracy per epoch.

We can see the point where the model starts overfitting as training continues to increase while the validation set starts to decrease. This was consistent across all 4 ConvNet models, most of them reached highest test accuracy around epoch 27-29. With adjustments to momentum and learning rate, this peaking of test accuracy could be made more gradual.



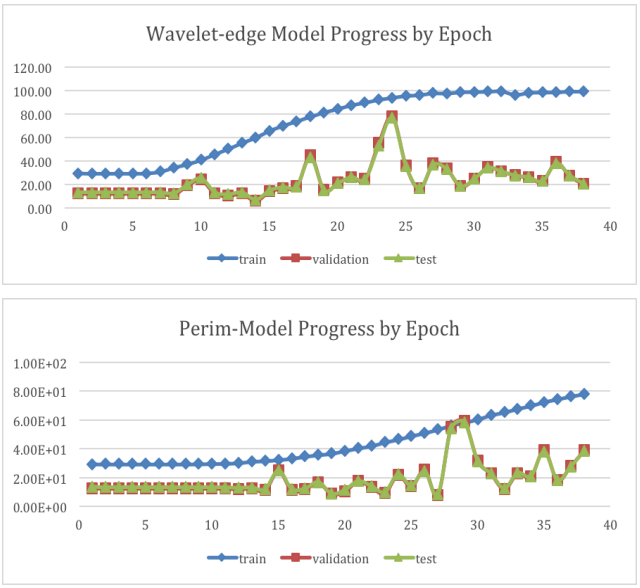


Figure 13. Hard-classification accuracy (per model type) as a function of epoch.

The basic model we have has typical layer of convolution, ReLu, and pooling. We briefly discuss the intuition behind these steps below.

1) Convolution: This layer creates the filter bank; with each layer, a filter size and stride is chosen and the function of the convolution operator is to extract features of the input with the filter-size specified window. The first few convolutional layers will obtain the lower-level features (i.e. edges, lines, or corners). These low level features will then be combined into higher level as we get deeper into the network. [14]

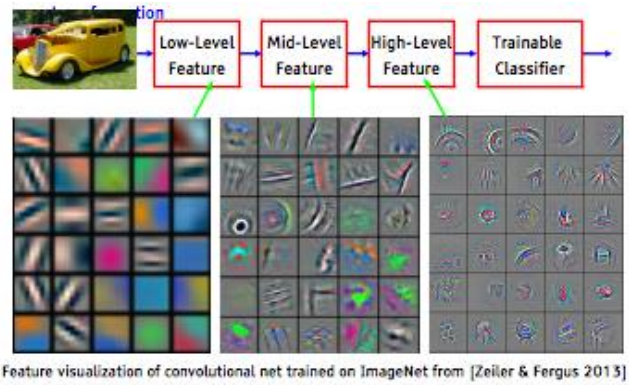


Figure 14. Example of ConvNet's hierarchy of learning.

2) ReLu: The ReLU layer is a rectified linear unit that uses a non-saturating activiation function. The purpose of this layer is to increase the nonlinearity of the decision function. This is chosen over the hyperbolic tangent function due to increase in speed. Often, logistic-type "squashing" functions are used to allow the network to fit more complex input-output surfaces. [14]

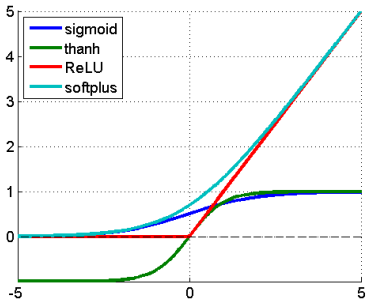


Figure 15. Example of rectifier nonlinearities. [15]

3) Pooling: The purpose of the pooling layer is to reduce variance. The pooling layer is either the max or average value of a particular feature over a region of an image. This ensures the ability to have the same classifiction result even if image features have small transaltions. Our model uses max pooling. [14]

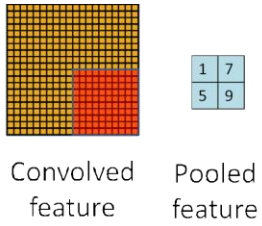


Figure 16. Example of pooling layer.

Our network starts with image sizes of 408x408. The inner box in red shows our chosen filter size. After a full layer of convolution -> reLu -> pooling, the image becomes 199x199. This is repeated until with have a 95x95 image at the final layer. The final linear layer maps it into five classification categories. The number that represents the width is the number of filter maps that exists at the layer (i.e. 1 layer in the beginning, since we only have one channel, then increasing the number of filter maps to 16 .. 32 .. etc.)

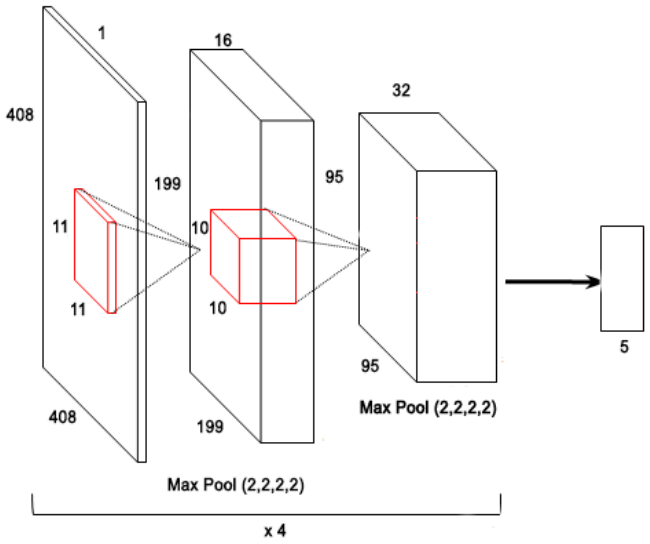


Figure 17. Architecture of our convnet.

BOOSTING ARCHITECTURE

As we discussed in our methodology section, the prediction output from the convnet was influenced into predicting only one class in order to achieve the maximum global accuracy. We combined the outputs of our convnet net and ran a multiclass classifier over it.

We realized that our metric, the weighted kappa measure was very different from the metric being optimized by ConvNet (mean squared loss). An algorithm could have done well by just predicting the middle as the correct one to minimize global loss.

We tried training a multi-class classifier on the log-likelihood values for the level of infection, output by the ConvNet. This let us learn a replacement for the argmax function which a ConvNet uses to predict the level of retinopathy.

The setup was organized as follows:

1) Train a ConvNet on training images. Get Negative Log Likelihood (NLL) values for the test and validation data sets. We chose the best epoch from the best ConvNet model we had.

2) Train a multiclass classifier on the log-likelihood values in the validation dataset. We did not train our classifier on training data as it could be over-fitting.

3) Test the multi-class classifier on the hold out test data.

This step gave us an improvement of 0.004 in the kappa metric. The small increase was due to bad performance on level 0, but there was a significant improvement in level 1-4. (Figure 18) This indicates that this approach helps, and could do better once we have better ConvNets.

The classifier which gave a gain in kappa was an Ada Boost classifier with 6000 rounds and depth 2 Decision Tree Classifiers as the weak classifiers. We tried other models which did not work: Random Forests (Grid search with 1000, 2000, 3000, 6000, 7000, 10000, 11000 estimators), Decision Tree Classifier (grid search with tree depth of 2 – 20).

VI. Results

Our convolutional neural networks produced very high training accuracy but inconsistent accuracy during validation and testing. In the graphs provided in Figure 7, we can see that there are some peaks of accuracy for validation and accuracy roughly around 30 epochs of the model training. However, this accuracy is a hard classification and not measuring a kappa score. We hypothesize the model may try to learn by predicting outside the median class but possibly incorrect classify close levels (such as predicting class 2 if it is class 1), thus a kappa score as a function of epoch would possibly provide some insight. We have kappa scores for the test and validation set, however, not for the training set, thus we have provided the confusion matrix instead.

The final multiclassication method, as stated in the previous section, was able to improve .004 on the kappa score.

Before:

```
[[ 0.95  0.04  0.    0.    0.01]
 [ 0.93  0.06  0.    0.    0.01]
 [ 0.95  0.03  0.    0.    0.01]
 [ 0.94  0.04  0.    0.01  0.01]
 [ 0.94  0.03  0.    0.    0.03]]
```

After:

```
[[ 0.51  0.19  0.26  0.03  0.02]
 [ 0.49  0.2   0.26  0.04  0.01]
 [ 0.53  0.18  0.25  0.03  0.02]
 [ 0.53  0.15  0.24  0.05  0.03]
 [ 0.49  0.13  0.31  0.02  0.05]]
```

Figure 18. Confusion matrix before and after passing it into the multiclass classifier. (Rows: true label, Columns: predicted label)

Algorithm	Dataset	Features	Channel	Best Test Accuracy (%)	Note
Random Forest	Unbalanced	Morphological	Green	73.38	predicting all as level 0, and few level 0 are wrong
Random Forest	Balanced	Morphological	Green	30.35	Baseline
ConvNet	Unbalanced	Morphological	Green	73.58	predicting all zeros
ConvNet	Balanced	Morphological	Green	61.06	
ConvNet	Balanced	Wavelet - Largest Areas	Green	37.55	
ConvNet	Balanced	Wavelet - Horizontal and Vertical Edges	Green	78.35	Best Model
ConvNet	Balanced	Morphological Perimeter	Green	59.24	
ConvNet -> AdaBoost	Balanced	Wavelet - Horizontal and Vertical Edges	Green	44.88	performing well on levels 1-4, performance decreases on level 0

Figure 19. Final Results

However, we can see that by using the multiclass classifier to learn the final argmax function, the model shys away from predicting only one class, and attempts to predict off the median class. Improvements to the foundational convnet models will likely give a larger increase gap in accuracy from this final step.

VI. Future Work

We believe that we have developed a well-founded framework in which we can improve results with further adjustments to our pipeline. While our ideas are large in number, we will condense it to just a few of the most significant tricks that should produce the greatest increase in accuracy.

Since we had computational limitations, we were unable to feed our convolutional net a higher number of samples and the full RGB image. We propose to reduce the image size from 408x408 down to 256x256 in order to run the model through all three channels. Since the size is reduced, we will also be able to add in data augmentation: in order to make the model robost to scale and location, we will produce a variety of image transformations per input, and feed it into the model. We believe the data augmentation alone should improve our foundational baseline models, which will improve the final multiclass classifier.

We also hope to attempt a deeper architecture that includes several more layers in order to capture the higher level features. Once the model is robust to the lower-level features, a deeper network will be able to benefit from the lower layer inputs.

While we used AdaBoost as our final classifier, we have considered using other strategies such forms of mixture of expert models, where we train models to learn a binary classification per level (or groupings of levels); an example would be a model learning that whether level 4 vs level 1-3. This reduces the noise and complexity of the model in trying to predict five separate classiciations.

Acknowledgment

We would like to acknowledge our advisor Brian d'Alessandro as well as our professor David Rosenberg in providing guidance to our methodologies and final framework. Further acknowledgement is to Kaggle and EyePACS for the our data sources.

References

[1] Diabetic Retinopathy Detection. Kaggle. 2014. www.kaggle.com/c/diabetic-retinopathy-detection/

[2] Diabetic Retinopathy. Wikpedia. http://en.wikipedia.org/wiki/Diabetic retinopathy

[3] Macular Edema. Wikpedia. http://en.wikipedia.org/wiki/Macular edema

[4] Macula of retina. Wikpedia. http://en.wikipedia.org/wiki/Macula of retina

[5] Angiogenesis inhibitor. Wikpedia. http://en.wikipedia.org/wiki/Angiogenesis inhibitor

[6] Vitreous Hemorrhage. Wikpedia. http://en.wikipedia.org/wiki/Vitreous hemorrhage

[7] Image Magick. http://www.imagemagick.org/Usage/color_mods/#equalize

[8] Decenciere et al. Feedback on a publicly distributed database: the Messidor database. Image Analysis & Stereology, v.33, n.3, p.231-234. Aug. 2014. ISSN 1854-5165.

[9] K. Zuiderveld. Contrast limited adaptive histogram equalization. Graphics gems, vol. IV, pp. 474–485, 1994

[10] R. Fisher. Bilateral Filtering for Gray and Color Images. CV Online. 2014. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL COPIES/MANDUCHI1/Bilateral Filtering.html

[11] P. M. Panchal, S. R. Panchal, S. K. Shah. A Comparison of SIFT and SURF. International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2013.

[12] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. A Comparison of SIFT and SURF. Interna- tional Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2013. [Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Speeded Up Robust Features", ETH Zurich, Katholieke Universiteit Leuven]

[13] Xiaoyu Wang, Han, T.X., Shuicheng Yan An HOG-LBP human detector with partial occlusion handling

[14] Yun, Wong Li, et al. "Identification of different stages of diabetic retinopathy using retinal optical images." Information Sciences 178.1 (2008): 106-121.

[15] Convolutional Neural Networks. Wikipedia. http://en.wikipedia.org/wiki/Convolutional_neural_network

[16] Wei. "Rectifier Nonlinearities." https://imiloainf.wordpress.com/2013/11/06/rectifier-nonlinearities/

[17] Y. V., Chee, C., Lim, C.M., and Ng, E. Y. K. (2008) Identification of different stages of diabetic retinopathy using retinal optical images

[18] Villasenor, J.D. ; Dept. of Electr. Eng., California Univ., Los Angeles, CA, USA ; Belzer, B. ; Liao, J. Wavelet filter evaluation for image compression