Automated Detection of Diabetic Retinopathy using Fluorescein Angiography Photographs

Marco Alban
Stanford University
marcoal@stanford.edu

Tanner Gilligan
Stanford University
tanner12@stanford.edu

Abstract

State-of-the-art convolutional neural networks (CNNs) and denoising techniques were used to diagnose the presence and severity of Diabetic Retinopathy from Fluorescein Angiography photographs. Data was provided by EyePacs consisting of fudus photographs with varying ranges of DR severity labeled by clinicians. A convolutional neural network classifier engineered from GoogLeNet for 5-class severity classification performed best with an AUC of 0.79% and an accuracy of 0.45%. This paper improves on past work in the scale and heterogeneity of the dataset used, to the best of our knowledge, no other published work on DR screening deals with a dataset of our magnitude.

1. Introduction

There exist multiple techniques for Diabetic Retinopathy (DR) diagnosis, an ocular manifestation of diabetes that affects more than 75% of patients with longstanding diabetes and is the leading cause of blindness for the age group 20-64 [6]. In this paper we focus on diagnosis through Fluorescein angiography (fudus) photographs, which involves careful examination of photographs taken with expensive equipment by highly trained clinicians. This detection technique is very resource intensive and requires very specialized clinician knowledge [1]. We aim to develop a computer vision model that closely matches human performance with the hope of one day being useful for the clinical community.

For more information on the epidemiology of DR and an analysis of how early detection of the disease can help slow or even avert its spread, consult [21]. For a comparative research paper on studies of risk factors of DR consult Yau et al. [13]. We stress that Fluorescein angiography is not the only a technique for diagnosis of DR; a comprehensive analysis on other detection techniques to diagnose DR can be found in [8].

The general format of our model is as follows. We take as input images that have been down-sampled to a tractable size (256 x 256). These images are preprocessed (normalied and denoised) and then used to train a convolutional neural network, either from scratch or via transfer learning. Once trained, test images are passed forward through the network and the model attempts to predict the severity of diabetic retinopathy.

One of the most interesting applications of the work done in this paper is the use of our model as a standardization technique. Currently, fudus photographs are labeled qualitatively by physicians, which is a rather subjective process. The goal is for our results to be useful in comparing labels across different clinicians, with the assumption that qualitative human measurements contain some degree of error.

2. Related Work in DR Screening Models

Previous work has been done in using machine learning and statistical models for automated DR screening. The methodologies can be categorized as using architectures that explicitly try to model features of interest, or methodologies that use automated feature extraction.

2.1. Previous Work in Explicit Feature Extraction Methods

Much work has been done in developing algorithms and morphological image processing techniques that explicitly extract features prevalent in patients with DR. For an overview of such algorithms consult [19]. Faust et al. [16] provide a very comprehensive analysis of models that use explicit feature extraction to DR screening. Shortcomings of these studies are in the magnitude of their scope (all the studies present results derived from less than 400 total data points), the homogeneity of the dataset and, the narrowness of the explicit features extracted from the images. For instance, Vujosevic et al. [20] build a binary classifier on a dataset of 55 patients by explicitly forming single lesion features. The authors in [3] use morphological image processing techniques to extract blood vessel, microaneurysm, exudate, and hemorrhage features and then train an SVM on a data set of 331 images achieving sensitivity 82% and

specificity 86%. The authors in [15] report accuracy of 90% and sensitivity of 90% (on binary classification task with a dataset of 140 images) using image processing techniques to extract area of blood vessels, area of exudates, and texture features which are then fed into a small Neural Network. Recent work by Rahim et al. uses fuzzy image processing techniques (fuzzy histogram equalisation and fuzzy edge detection) for a DR detection system.

2.2. Previous Work in Methods with Neural Network Based Feature Extraction

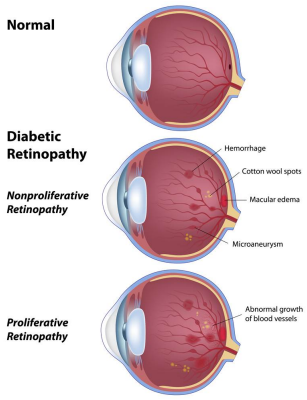
In a very recent work, Wang et al. [17] use a CNN (LeNet-5 architecture) as a feature extractor for addressing blood vessel segmentation. The model has three heads at different layers of the convnet which then feed into three random forests. The final classifier uses and ensemble of the random forests for a final prediction achieving an accuracy and AUC on 0.97/0.94 on the DRIVE [11] dataset (a standard dataset for comparing models addressing vessel segmentation). Similarly, the authors in [22] propose a model that uses convolutional neural networks for the general task of thin image segmentation and show benchmark results on the DRIVE dataset achieving an AUC of 0.89. Perhaps, the most similar published work to the one proposed by this paper is that of Lim et al. [9] where the authors propose building a convolutional neural network for lesion-level classification and then use the learned feature representations for image-level classification, however the scope of the study is limited in that the dataset used contains 200 images for a homogenous source.

2.3. Background Literature Survey

To inform our model development and result analysis, we conducted a literature survey of medical journal articles describing DR features and past work done in DR detection. Beyond useful for feature extraction and feature engineering as well as model benchmarking understanding the medical basis for the problem at hand helped focus our model examination later in the development process (for instance by examining if neurons in a neural network were in fact activated by features that were deemed important by our literature survey). In addition, understanding the medical basis for our problem informed our analysis of with image normalization and denoising techniques to use (to try to preserve the most relevant features).

The National Eye Institute provides a standardized description of the severity class of DR patients (which are the classes that our classifier predicts). There are four severity scales, the first three describe non-proliferative DR (NPDR) and the last proliferative DR (PDR). The severity scales are characterized through a progression of four stages:

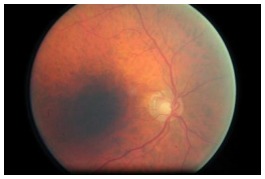
- Mild NPDR - Lesions of micro-aneurysms, small areas of balloon-like swelling in the retinas blood vessels.
- Moderate NPDR - Swelling and distortion of blood vessels
- Severe NPDR - Many blood vessels are blocked, which causes abnormal growth factor secretion [7]
- PDR - Growth factors induce proliferation of new blood vessels inside surface of retina, the new vessels are fragile and may leak or bleed, scar tissue from these can cause retinal detachment. [2]



3. Dataset

3.1. Overview

Data was drawn from a dataset maintained by EyePacs, and provided via Kaggle. The dataset is composed of multiple, smaller datasets of fudus photographs drawn from various sources. Each image is assigned a class based on the presence and severity of DR (see scale explanation in 2.3), where each image was labeled by a trained clinician. Below are examples of dataset images:



(a) Fudus p without DR

(b) Eye with DR

3.2. Heterogeneity and Noise

The challenges presented by this task and dataset are numerous. The dataset used is highly heterogeneous; the photographs are from different sources, cameras, resolutions, and have vastly different degrees of noise and lighting (converse to other published works, see section 2). Resolutions

ranged from 2592x1944 to 4752x3168. We believe that being able to generalize to this noisy dataset adds to the value of the work done here, since the results would likely be more robust and general. Figure 1 shows some examples of the poor quality of images in our dataset.

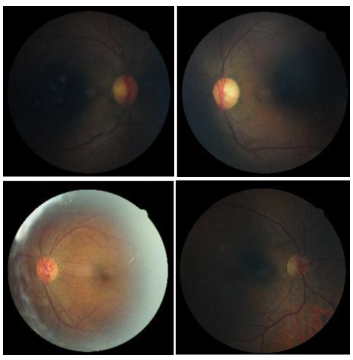


Figure 1: Top images show poor quality images where lighting is targeted to a specific neighborhood. Bottom-left images shows over-exposed fudus photograph in the dataset.

3.3. Preprocessing

Due to the noise in our data, as well as the limited number of examples of some classes, there were numerous preprocessing steps we took. From our error analysis, we found that many of our images had excess black-space on either side of the eye, part of the preprocessing was removing this background. The images came in varying sizes and aspect ratios, so we standardized this by downsizing all images to 256-by-256 images. About 80% of the pictures had an aspect ratio of 3:2, so our downsampled/cropped images retained this ratio, but images with other aspect ratios (most commonly 4:3) became stretched. To approach this issue we devised an algorithm that altered cropping and downsampling based on the aspect ratio of the image. Once the images had been downsized, we implement various denoising schemes (see Section 4.2).

Due to the limited number of training examples for some classes, we also created augmented images to increase class sizes. For this, we took each of the images created in the previous step, and produced a mirrored image of it. Both the original and mirrored image were then duplicated at 90, 180, and 270 degree rotations, effectively increases our class sizes by 8x.

3.4. Challenges Intrinsic to DR Screening

In addition to the images themselves being heterogeneous, the presence of Diabetic Retinopathy is also heterogeneity. One of the main indicator of DR is the existence of lesions and exudates on the eye. These features, however, can have vastly different sizes, shapes, and frequency (see Figure 2). In order to construct a viable model, our classifier must contain some degree of robustness to these different features.

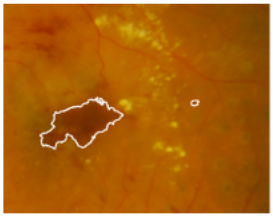


Figure 2: Lesions of Vastly different Sizes in DR [9]

3.5. Summary Statistics and Class Imbalance

Our dataset consists of highly imbalanced class-labels. The table below shows the class proportion statistics:

Class	Number	Percentage
Negative	25810	73.5%
Mild NPDR	2443	6.90%
Moderate NPDR	5292	15.10%
Severe NPDR	873	2.50%
PDR	708	2.00%

This adds to the difficulty of the task at hand, for an analysis of how we addressed class imbalance see section 4.3

4. Methodology

4.1. Overview

We begin by trying to solve the 5-class classification task on our noisy dataset. We denoise, normalize, and augment the data as described in the preprocessing section, and address the class imbalance problem by either over-sampling the minority classes or using cost-sensitive learning. Next, we build three different models: a custom architecture built as a baseline where all layers are trained, a classifier built using a pretrained AlexNet [14] where only the last layer is retrained, and a GoogLeNet [18] constructed similarly to AlexNet. All weights that weren't loaded via transfer learning were initialized using the Xavier [10] initialization scheme. For all three of our models, the final prediction is made using a softmax layer. Therefore, our loss function is defined as:

$$L_i = -\log\left(\frac{e^{w_{y_i}}}{\sum_j e^{s_j}}\right)$$

where s_{y_i} is the score for example i's label, and s_j is the score for a particular label j. The softmax contained in the log ensures that the prediction probabilities are a proper probability distribution.

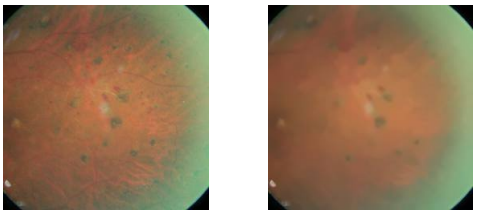
Upon conducting analysis of our errors, we propose two problem relaxations, namely a 2-class and 3-class classifier. We then build classifiers for these two relaxations using a similar pipeline as the one described above. The problem relaxations are motivated by the idea that the severity classes are somewhat subjective, i.e. the differences between an image labeled as mild and an image labeled as moderate as nuanced and clinicians may disagree in their labeling.s

4.2. Pipeline and Normalization Schemes

To address the issue of heterogeneity in the dataset, we conduct a brief overview of image denoising techniques. For a comprehensive overview of image denoising techniques, consult [5]. Our task necessitates very nuanced denoising techniques as certain artifacts of fudus photographs (such as lexions and exudates) may very well appear to be noise yet are precisely the features that we would like to keep for our classifier prediction. In particular, we use Non-Local Means Denoising (NLMD) as proposed by Buades et al. [4] as a preprocessing step. The denoising of an image $x = (x_1, x_2, x_3)$ on channel i at pixel j is implemented as:

$$\begin{aligned} \hat{x_i}(j) &= \frac{1}{C(j)} \sum_{k \in B(j,r)} x_i(j) w(j,k), \\ C(j) &= \sum_{k \in B(j,r)} w(j,k) \end{aligned} \tag{1}$$

In the above, $B(j, r)$ denotes a neighborhood of radius r around pixel j, and the weight $w(j, k)$ depends on the squared of the Frobenius norm distance (or another induced norm distance) between color patches centered at j and k that decays under a Guassian kernel. This denoising technique was chosen among the ones surveyed because of its flexibility. In particular, by varying the width of the kernel we adjust our denoising scheme to better suit the needs of DR screening. The figure below shows a side-by-side comparison of a denoised training example with class label 4 (Proliferative DR).



(a) Original (b) Denoised

Figure 3: Comparison of Non-Local Means Denoising

As demonstrated by the images above, we lose some information with NLM Denoising, but we are able to preserve some of the exudates and scar tissue produced by the leaking blood vessels. Additionally, as a baseline normalization scheme, we also subtract the training image mean from the datasets.

4.3. Addressing Class Imbalance

We address class imbalance in two approaches: using cost-sensitive learning, and using class balanced training sets. As a cost-sensitive learning approach, we modify our loss to be a generalization of the multinomial logistic loss. Specifically we use the InfoGain Loss as described in [12]. The loss is computed as:

$$L = \frac{-1}{N} \sum_{n}^{N} H_{ln} \log \hat{p}_n \tag{2}$$

where H_{ln} denotes the ln row of H, an info-gain matrix. For simplicity we use a diagonal matrix defined as:

$$H_{ij} = \begin{cases} 0 & \text{if } i \equiv j \\ 1 - f_i & \text{otherwise} \end{cases}$$

where f_i is the frequency of class i in the batch.

As another approach, we address class imbalance by training on a class-balanced subset. We preprocess the dataset by producing augmentations of underrepresented classes to increase the class size, and subsample from the overrepresented classes. This allows us to have a uniformly balanced training set.

4.4. Baseline

As a baseline, we built a convolutional neural network from scratch that acts as our control. The model is trained using randomized hyperparameter search. The architecture for our baseline is:

[Conv - ReLU - Pool]x2 - Affine - Softmax

The model was initialized using the Xavier initialization scheme, and updated using Adam. The model served the purpose of guiding our research and the results motivated some of the decisions in our improved transfer learning models.

4.5. Convolutional Network Architectures

4.5.1 AlexNet

The first pretrained model we use was AlexNet [14]. AlexNet, developed in part by Alex Krizhevsky in 2012, is one of the best CNNs today, having won the imagenet challenge. We utilized this model by loading the pretrained weights, and only retrain the final fully connected layer to

predict 5 classes rather than 1000. This usage of transfer learning is viable because many of the early layers of the network learn similar features, such as edges and lines. By loading these pretrained weights, our model effectively already knows how to detect lines and edges, and need only learn how to use them to make predictions for our problem. Below is an image that show the basic architecture of Alexnet from the original paper.

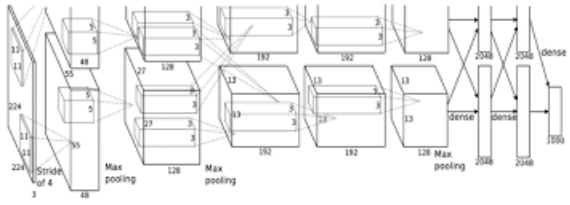


Figure 4: Architecture for AlexNet [14]

4.5.2 GoogLeNet

The second pretrained model we used was GoogLeNet [18]. GoogLeNet, which was developed at Google, won the imagenet challenge in 2014, setting the record for the best contemporaneous results. Motivations for using this model was a deeper architecture, the addition of Inceptions Layers and the possibility of using an ensemble classifier from the three different heads of the net output in future work. Similar to Alexnet, we loaded the pretrained weights into our network, and retrained the final layer to predict 5 classes rather than 1000. Below is an image which demonstrates the GoogLeNet's general architecture. Similarly we also retrain the last two layers of the net as a different scheme.

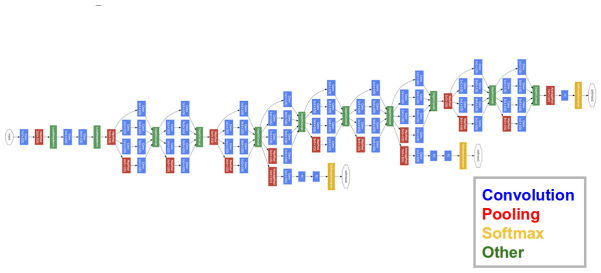


Figure 5: Architecture for GoogLeNet [18]

4.6. Problem Relaxations

As part of our error analysis, we concluded that the problem could be simplified in a way that would still yield actionable results to the clinician community. We propose two other characterizations of the problem:
· 2 Class - Binary classification of DR presence
· 3 Class - Merge classes 1 with 2, and 3 with 4

The motivation for proposing the binary classification problem is self-evident; automatically diagnosing the presence of DR (even without a severity measure) is useful in and of itself. The motivation for proposing the 3-class severity classification problem is twofold. Firstly, it tests our hypothesis that we could simplify the problem given the highly noisy data, and secondly it acts as a coarser measure of severity that may in fact be more robust to subjectivity. As our survey of medical journals informed us, the severity classifications are qualitative and can have a certain degree of subjectivity. That is, a clinician may label a fudus photographs as "Mild DR", whereas another may label the same image as "Moderate DR".

5. Results

Our models were run AWS machines with 8 Intel Xeon E5-2670 (Sandy Bridge) processors and a gpu with 1,536 CUDA cores. Both the custom and transfer-learned models were developed leveraging Caffe, a framework developed by the Berkeley Vision and Learning Center [12].

5.1. Evaluation Metrics

For our models, we had 4 main metrics for evaluating their performance on the data. The first was accuracy, which is simply the proportion of examples that were classified correctly, which could be calculated for all predictions. The other three metrics, recall (proportion of positives correctly predicted), precision (proportion of positive predictions that were correct), and AUC (area under ROC curve) had to be calculated on a per-class basis since this is a multi-class problem.

5.2. Hyperparameters

For our baseline model, hyperparameters were selected using a random search of the parameter space. Due to the limited amount of time and computational resources available to us, we were not able to run as thorough a search as we would have liked, but we settled on a learning rate of 0.00001 with momentum 0.9 and decay of .005. We do concede that, given the rather poor performance of our baseline, there are likely a better set of hyperparameters, we elected to focus our time and resources on the transfer-learning models. For our batch size, we used 25 images. We chose this value because it was one of the largest batches we could use as constrained by the memory of the machines.

For both AlexNet and GoogLeNet, baseline hyperparameters were provided in the model's description. In both cases, we took the existing parameters and perturbed them by an order of magnitude in both directions to see how this

altered our prediction (in addition we also tried increasing the learning rate multiplier for the layers that were trained from scratch). In most cases, we found that this perturbation lead to a decrease in the model's overall performance, but in a few cases, these perturbations lead to improved results. One such example of the latter case is with AlexNet. When first training AlexNet, we found that our loss would become very large very quickly, and then never change. To attempt to correct this, we performed randomized hyperparameter search on the learning rate bounded a few orders of magnitude above and below the initial parameters. By examining the loss and training accuracy we determined a final learning rate an order of magnitude lower than the given. Further refining the learning rate beyond this did not lead to noticeable improvements, but the initial alteration was highly beneficial. Just as in the baseline, we used a batch size of 25 since larger batch sizes could not be handled by our machines.

We also attempted to use the Adam update rule for both GoogLeNet and AlexNet. In the case of GoogLeNet, the introduction of Adam lead to a much faster convergence time, which allowed it to out-perform AlexNet. When introduced into AlexNet, however, the results of the model were equivalent to a random prediction. We attempted to further tune the hyperparameters so that Adam would be viable, but after much time spent, we were unable to produce better results. We advocated for a more thorough search under a larger financial budget and longer time frame. To account for the slower convergence expected, AlexNet is trained for more iterations.

5.3. Model Performance

Note that all results for the following models are based on a uniformly distributed class set in the testing data.

5.3.1 Baseline

Our baseline performed rather poorly. In the 2-class, 3-class, and 5-class cases, it performed slightly better than randomly guessing on the validation and test sets. When tested on the training set, however, it was able to perform noticeably better than randomly guessing; we can conclude that the network was at least able to learn some decision boundaries. Below summarizes the results for our baseline on the test set, where the recall and precision have been averaged across all classes:

	Accuracy	Recall	Precision
2-Class	0.541	0.502	0.489
3-Class	0.353	0.387	0.301
5-Class	0.227	0.201	0.235

5.3.2 AlexNet

After developing our baseline, the first model we tried was AlexNet. Loading the pretrained model and retraining the final layer greatly improved on the results produced by our baseline, and generated our first legitimate results. We were able to achieve a training accuracy of 72.9% on the 5-class problem, so we were clearly able to overfit our data. Interestingly, even as we continued to overfit more and more (loss ¡0.1), our validation accuracy remained relatively constant. We attempted to introduce higher regularization via layers' weight_decay parameter to counter this traning-set overfitting, but the effect was an overall decrease in our validation performance, so we elected to omit it. The accuracy results for AlexNet on varying numbers of classes is summarized below under the best learning rate and hyperparameters that were searched.

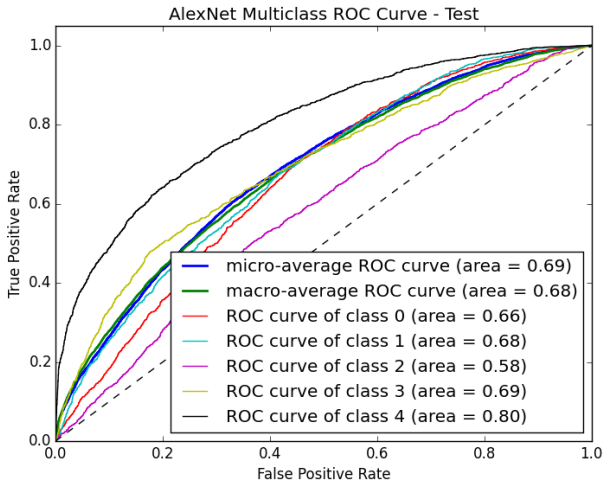
Classes	Accuracy
2	0.6695
3	0.5705
5	0.4073

From the above table, we can derive a few insights. First, although the binary classification task of detecting Diabetic Retinopathy may be easier than the original task, we were not able to show this with our classifier. The 2-class model only performed 17% better than random, while the 3-class and 5-class models performed 24% and 21% better respectively. Second, we find the interesting result that the we were able to achieve a higher evaluation metric on the 3-class problem. In addition to accuracy, we also break down the recall and precision of the 5-class model by class, allowing us to analyze which classes our model performs poorly at. Below summarizes these results:

Class	Recall	Precision
0	0.398	0.306
1	0.386	0.343
2	0.294	0.254
3	0.282	0.458
4	0.456	0.571

From the above, we see that classes 3 and 4 have relatively high precision, while classes 0, 1, and 2 have relatively low precision. This indicates that our model is less likely to predict 3, but when it does, it tends to be correct. In contrast, the model seems to more liberally predict classes 0, 1, and 2, which as a result causes its precision to go down. One interesting result that can be drawn from the above is that the model is relatively good at identifying images of class 4 compared to the other classes. This is likely a result of many class 4 images displaying extreme cases of Diabetic Retinopathy in which there are numerous, large

lesions on the eye, allowing for easy identification. In addition to the results above, we also provide the ROC curve for the 5-class version of AlexNet below, where the AUC for each class is given by the legend:



5.3.3 GoogLeNet

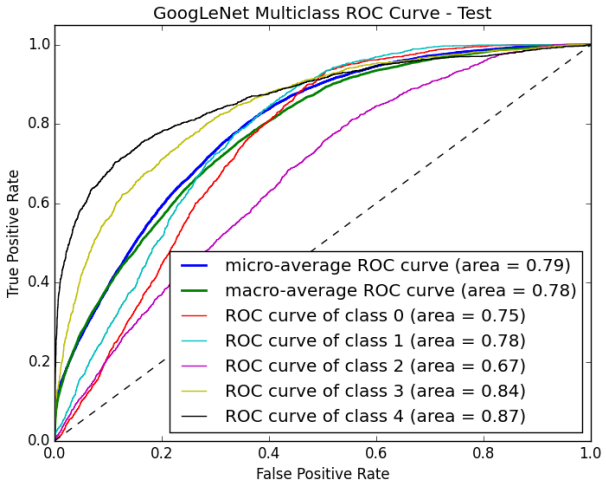
The second model on which we attempted to use transfer learning was GoogLeNet. In general, GoogLeNet seemed to perform marginally $(1-2\%)$ better than AlexNet in virtually all situations where we tried both. Similar to AlexNet, we were able to achieve a significantly higher training accuracy than validation accuracy (74.2% vs. 41.7%), indicating that we were overfitting our training data. We again tried to alter the weight_decay parameter via randomized hyperparameter search. Below summarizes our accuracy results across varying class numbers:

Classes	Accuracy
2	0.7105
3	0.5821
5	0.4168

When compared to the results obtained via AlexNet, we can see that GoogLeNet preformed better in every category. Most noticeably, we can see a nearly 5% improvement in the binary classification task, while the 3-class and 5-class problem experienced marginal, but non-trivial, improvements. In contrast to what the results from AlexNet conveyed, it seems that the binary classification problem is on par with the 5-class problem in terms of performance. Even with this substantial improvement in 2-class performance, however, the 3-class version of the problem still produces the best results. To better understand what kind of predictions GoogLeNet is making, we again break down the recall and precision by class:

Class	Recall	Precision
0	0.288	0.343
1	0.536	0.379
2	0.334	0.327
3	0.567	0.575
4	0.567	0.762

The above table tells us a few things. First, similar to AlexNet, GoogLeNet has a high precision for class 4 compared to the other classes. This again indicates that the model is good at prediction class 4 (notice also the comparitively high recall 0.567). In addition, GoogLeNet seems to be much better than AlexNet at predicting class 4, as both the recall and precision for GoogLeNet are higher. One interesting result for GoogLeNet is that classes 0 and 2 seem to be fundamentally harder to predict than classes 1, 3, and 4. This is evident from the recall for classes 1, 3, and 4 being much higher (¿0.2) than those for classes 0 and 2, and the precisions being higher as well. This is in contrast to AlexNet, where class 4 seemed to be the only class that was noticeably easier to predict. Below is the 5-class ROC curve for GoogLeNet, where the AUC is provided in the legend for each class.



5.4. Error Analysis

Analyzing the images that our models incorrectly classified was challenging in a and of itself since this task is hard even for trained clinicians and doctors. There were, however, some common things that seemed to frequently co-occur with misclassified images when compared to correctly classified images.

5.4.1 Black Space

As demonstrated by the example images in section 3.1, most images have some black space on either side of the eye, however some do not. When examining predictions for images, it seemed that images with more black space tended to be misclassified more frequently. In order to counteract this, we attempted to crop a fixed number of pixels from either side of the image, so as to leave only the eye. This was problematic, however, because not all images had black space, and we were effectively removing part of the eye. Even though we were damaging some of the images, the overall effect on our performance was positive, so we elected to keep it. Ideally, we would automatically detect a radius ball of there the eye is and only remove the outside, but this was infeasible due to the time constraints of this project.

5.4.2 Image Color

The fact that the images of our dataset come from multiple smaller datasets leads to a high amount of heterogeneity in the images. One manifestation of this is the fact that the color of the eye in the image can vary greatly. In some cases, the eye would appear as a dark green or blue, while other times it could be a red or yellow. This is problematic because the colors aren't necessarily distributed evenly among classes, so the model may learn that certain colors correspond to a certain class, even though these are independent. Future work on this project should involve standardize the images eye color. Projecting the images to gray-scale may be hurtful as some of the important features (hemorrages) can only be distinguished via colored images.

5.4.3 Image Brightness

One of the most common types of images that was misclassified were those that were extremely dark. As demonstrated by figure 2 in section 3.2, many images in the dataset are so dark that most of their features are indistinguishable. This effectively made the models have very little information on which to make a prediction, and as a result, made near-random predictions on them. Our existing model has no method for rectifying this problem, but ideally we would have some process that would detect when an image's average brightness is too low, and would somehow increase the brightness of the region containing the eye.

5.4.4 Bad Images

With images that are too dark, there is at least something that can be done to improve their quality by enhancing the information that is present. In some cases, however, there are images where the information in the affected region is irrecoverable. An example of this is the bottom-right image in figure 2 of section 3.2. In this image, the outer ring has some sort of glare or whitening that causes the information there to be lost. This makes it so the features from which the model can make a prediction are substantially reduce, making it difficult to predict correctly unless the lesions happen to occur only in this region.

6. Future Work

In the future, there are a few things we would like to have done. First and foremost we would like to have a human-oracle evaluate the difficulty of the learning task. As stated above, the data is corrupt and highly noisy, having a trained human physician label a subset of the images given could help us understand how our models compare to the true difficulty of the task. If interested please contact the authors of the paper.

Further work may also include using both of a patient's eyes passed through the classifier where the final prediction is an ensemble of the predictions of each individual eye. Additionally, we would like to train ensemble learning with the different head outputs of GoogleNet (to try to combat overfiting). Future work may also include more preprocessing, in particular we would like to implement other methods for denoising the images. Due to time constraints we were not able to run a model with some of the other denoising schemes in our literature survey. We recommend trying median filters for denoising as these have been shown to perserve image features. Other work may also include more sophisticated methods for cleaning up the blackspace in the images. For our existing implementation, we simple removed a predefined number of pixels from each edge, but ideally we would remove blackspace based on the amount present in the image. We would also want to have increase the brightness of images detected to be extremely dark, so as make the features we're looking for show an increased contrast with its surroundings. Standardizing the colors of the eyes either via a grayscale (as a starting point) and intensity transformation would likely also prove beneficial.

7. Conclusion

We presented an analysis of a model for multi-class identification of the severity of DR from fluorescein angiography photographs. The model performs well in comparison to human evaluation metrics. Further work can be done in exploring more nuanced data normalization and denoising techniques. For instance, apriori knowledge of the sources of error for equipment used to capture fudus photographs could help in building more robust normalization schemes. Other work may be in combining weak learners in ensembles or using an ensemble of classifiers trained on raw image pixels and trained on explicit feature extractors (as much work has been done in these, section 2.1).

References

[1] Grading diabetic retinopathy from stereoscopic color fundus photographsan extension of the modified airlie house classification: Report number 10.

[2] Facts about diabetic eye disease. *NIH National Eye Institute*, 2014.

[3] Ng EY Chee C Tamura T. Acharya U, Lim CM. Computer-based detection of diabetes retinopathy stages using digital fundus images. *Proceedings of the Institute of Mechanical Engineers, 545-553*, 2009.

[4] J.M. Morel Buades, B. Coll. A non local algorithm for image denoising. `http://dx.doi.org/10.1109/CVPR.2005.38`.

[5] J.M. Morel Buades, B. Coll. A review of image denoising methods, with a new one, 2006.

[6] Linda Geiss et al. Engelgau, Michael. The evolving diabetes burden in the united states. *Annals of Internal Medicine*, 2004.

[7] Vinod Patel Eva M Kohner and Salwan M B Rassam. Role of blood flow and impaired autoregulation in the pathogenesis of diabetic retinopathy. *American Diabetes Association*.

[8] Hykin PG Fraser-Bell S, Kaines A. Update on treatments for diabetic macular edema. *Current Opinion in Opthalmology*, 2008.

[9] Wynne Hsu T. Wong Gilbert Lim, Mong Li Lee. Transformed representations for convolutional neural networks in diabetic retinopathy screening. *Modern Artificial Intelligence for Health Analytics*, 2014.

[10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*, 2010.

[11] M. Niemeijer M.A. Viergever B. van Ginneken J. Staal, M.D. Abrmoff. Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 2004.

[12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[13] Ryo Kawasaki et al Joanne Yau, Sophie Rogers. Global prevalence and major risk factors of diabetic retinopathy. *Diabetes Care* `http://care.diabetesjournals.org/content/35/3/556.full.pdf`, 2012.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[15] Bhat P. S. Acharya U. R. Lim C. M. Nayak, J. and M Kagathi. Automated identification of different stages of diabetic retinopathy using digital fundus images.

[16] K. Ng J. Suri O. Faust, R. Acharya. Algorithms for the automated detection of diabetic retinopathy using digital fundus images: A review. *Springer Science, Journal of Medical Systems*.

[17] G. Cao B. Wei Y. Zheng G. Yang S. Yang, Y. Yin. Hierarchical retinal blood vessel segmentation based on feature and ensemble learning. *Journal on Neurocomputing Vol 149, p. 708-717*, 2015.

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[19] P. Massin A. Erginay T. Walter, J. Klein. A contribution of image processing to the diagnosis of diabetic retinopathydetection of exudates in color fundus images of the human retina. *IEEE Transactions on Medical Imaging*.

[20] Benetti E. Massignan F. Pilotto E. Varano M. Cavarzeran F. Avogaro A. Vujosevic, S. and E. Midena. Screening for diabetic retinopathy: 1 and 3 nonmydriatic 45-degree digital.

[21] Baxter H Forrester J et al Williams R, Airey M. Epidemiology of diabetic retinopathy and macular oedema: a systematic review. `http://dx.doi.org/10.1038%2Fsj.eye.6701476`, 2004.

[22] V. Lempitsky Y. Ganin. N4 fields: Neural network nearest neighbor feilds for image transforms.