Deep Learning Application in Primary Open-angle Glaucoma Detection and Analysis

Eric Zhewen Li*
zl733@cornell.edu
Jacobs Technion-Cornell Institute,
Cornell Tech, Cornell University
New York, NY, USA

Flora Yu Shen*
ys2223@cornell.edu
Jacobs Technion-Cornell Institute,
Cornell Tech, Cornell University
New York, NY, USA

Yifan Peng†
yip4002@med.cornell.edu
Population Health Sciences,
Weill Cornell Medical College
New York, NY, USA

ABSTRACT

Primary open-angle glaucoma (POAG) is one of the leading causes of blindness for people over the age of 60. It is expensive and time-consuming to manually detect glaucoma from fundus photographs. Recently, applied deep learning is widely used in the biomedical field. With the rapid development of convolutional neural networks, it is easier to perform object detection, extraction, and image analysis instead of handcrafting. In this work, we propose a deep learning network to detect POAG from fundus photographs. Using transfer learning techniques with three different pretrained models, we are able to achieve a model with accuracy of 90.94%, ROC AUC of 0.7621, and F-score of 0.4497.

1 INTRODUCTION

Primary open-angle glaucoma (POAG) is one of the leading causes of blindness for people over the age of 60. Many forms of glaucoma have no warning signs. Hence, it is important to have regular eye exams so a diagnosis can be made in its early stages and treated appropriately, helping more patients to get health back. However, it is expensive and time-consuming to manually detect glaucoma from images. Deep learning (DL) techniques, which have been widely and successfully applied to many tasks that involve image analysis, could be used as supportive tools for medical professionals caring for patients with suspected or known glaucoma. Some additional advantages of an automated model include its error-prone property, and the possibility to be deployed to under-development regions where there is a lack of supply in ophthalmologists. In this project, we have implemented deep learning models using transfer learning techniques to detect POAG from clinical trial eye-scan fundus photographs, with three different pretrained models, including *VGG*, *ResNet*, and *DenseNet*.

2 RELATED WORK

There is a similar work where a research team uses deep learning to detect glaucoma prior to disease onset. [5] Their approach is to train deep learning models on 85% of the OHTS data [1] (training set) and validate on the remaining 15% held-out set. This research shows that deep learning models can predict glaucoma development prior to disease onset with reasonable accuracy. When predicting glaucoma development 4-7 years before onsite, this research achieved an AUC of 0.77 (95% confidence interval 0.75, 0.79).

Our research approach is quite similar to the aforementioned work. However, we are predicting whether glaucoma is present directly from the fundus photos, instead of prior to onsite. Our research work serves as the foundation to the deep learning-based survival model to predict glaucoma before onset.

3 METHOD

Image classification models have millions of parameters, and training them from scratch is very computationally expensive. Transfer learning (TL) is a deep machine learning technique that focuses on "storing knowledge gained while solving one problem and applying it to a different but related problem." [6] For this project, we are working on the image recognition problem; hence, we are able to use some pre-trained image recognition models as base models before we fine-tune them to fit on our dataset. We have experimented with different pre-trained models, transfer learning techniques, and hyperparameters, to benchmark their performances on predicting POAG from fundus photos.

3.1 Pretained Models

For our experiment, we utilize three different pretrained models instead of training from scratch. Pretrained models are base architectures developed by other research groups / companies. They have been trained on a large-scale dataset with a large number of layers. These models have been trained for a long time, and they can achieve high accuracy on image classification tasks. For our project, we have chosen three different pretrained models - *VGG*, *ResNet*, and *DenseNet*. All three models are pretrained on *ImageNet*, a dataset including 14 million images belonging to 1000 classes.

3.1.1 VGG. *VGG* was initially proposed by Karen Simonyan and Andrew Zisserman in 2014. [4] The model achieved 92.7% top-5 test accuracy when predicting on the *ImageNet* test set. There are several variations of this network, with different configurations from 11 to 19 weight layers. For our experiment, we chose *VGG-16* because it has enough layers to provide meaningful training while being not too computationally expensive to fine-tune.

3.1.2 ResNet. *ResNet* implemented an "identity shortcut connection" that skips one or more layers [2], which helps to feed the input from previous layer to the next layer without any modifications. This model also has a bottleneck design which to reduce its complexity. Compared to *VGG* network, *ResNet* uses fewer kernels but has more of them stacked alternating between convolutional operation and non-linear activation functions. There are five different configurations of this network. We chose *VGG-152* since our

[*] Both authors contributed equally to this research.
[†] Advisor

model is trained on a relatively smaller dataset. It doesn't need a 1001 layers deep model to hold parameters. With 152 layers, it can capture the right amount of information without overfitting.

3.1.3 DenseNet. *DenseNet* [3] adds residual connections between every forward convolutional layer to subsequential convolutional layers. This model is featured with strong gradient flow, so that the error signal can be easily propagated to earlier layers. It also has advantages on parameter and computational efficiency, which means with a deeper layer, it has a much smaller size than *ResNet*. Therefore, the complexity of features is very low too. *DenseNet* has the smallest trainable parameter size among all three pretrained models, so we chose *DenseNet-201* that has the most number of layers.

3.2 Transfer Learning

Transfer learning focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [6]. In this project, we experiment with two transfer learning strategies.

The first strategy is to use pretrained model as a fixed feature extractor. We only train on the last linear layer while freezing all the other layers (i.e., to keep their weights fixed) in the network. This approach utilizes the knowledge previously learned by the model as the starting point for our re-training.

The second strategy is to fine-tune on the entire network, including both the last linear layer and the weights of the pretrained network by continuing the backpropagation.

3.3 Image Augmentation and Transformation

All three base models in this project require input images of size $(224, 224, 3)$, representing their width (in pixels), height (in pixels) and RGB color channels. In order to match our dataset with the pretrained model requirements, we have also implemented the following image transformations for our development and test sets before we feed them to the network.

- Resize to 256×256
- Center Crop to 224×224
- Normalization

To prevent our models from overfitting during training, we performed the following image augmentations and transformations on the training dataset before feeding them into the network and start training.

- Random Resized Crop to 224×224
- Random Horizontal Flip
- Random Vertical Flip
- Random Rotation from -30 to 30 degrees
- Normalization

In the normalization step, all of our datasets (train, development, and test) are normalized to have $mean = [0.485, 0.456, 0.406]$, and $standard\ deviation = [0.229, 0.224, 0.225]$, each value corresponding to the RGB channel, as expected by our pretrained models.

3.4 Loss Function and Optimizer

For this project, we use the cross-entropy loss as our objective function because it measures the performance of a classification model with a probability output between 0 to 1. The function is defined as $loss = -(y \log(p) + (1 - y) \log(1 - p))$, where y is each class label, and p is the probability of predicting that label. Our dataset is highly imbalanced, where 90% of the images are of the negative class and only 10% of the positive class. In order to prevent our model from simply predicting everything to be negative, we incorporated weighted cross-entropy loss with $scale = [0.9, 0.1]$. The weighted cross-entropy losses will increase or decrease the relative penalty of a probabilistic false negative for individual classes.

We use Adam optimizer during training, which combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems. Adam optimizer is also advantageous with high computational efficiency and low memory requirements.

4 EXPERIMENTS

4.1 Dataset

The fundus photographs of this study (i.e., the dataset) come from the Ocular Hypertension Treatment Study (OHTS) [1]. This is a fifteen-year-long clinical trial, consisting of 37,334 visits from 1,636 patients. There are three types of images in the dataset, $-L$, $-R$ and $-S$, meaning left eye image, right-eye image, and two eyes on the same image correspondingly.

Among the 1,636 total patients, there are 178 patients who developed POAG on their left eyes, and 184 patients developed POAG on their right eyes, with 83 patients who developed POAG on both eyes (Figure 1). There is an average of 11.41 visits record on left eyes among all patients, with a minimum of 1 visit and a maximum of 21 visits, while there are 11.41 visits on right eyes, with a minimum of 1 and a maximum of 23 (Figure 2). In general, the median days among all patients to develop glaucoma (left or right) is 2919 days from the beginning of the trial, with an interquartile range of $1777.25 - 3773.25$ days. (Figure 3)

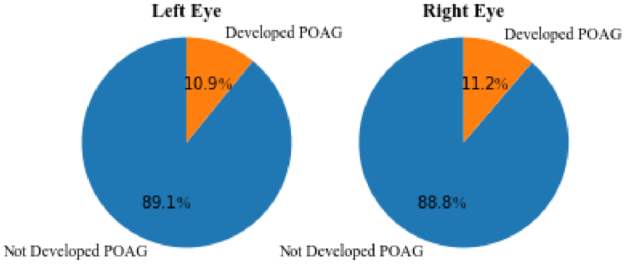

Figure 1: Distribution of positive/negative labels of dataset on patient level

4.2 Experimental Settings

4.2.1 Preprocessing. Our dataset has two types of images, one is L/R single eye image, and the other is both eyes on one image. In order to clean up the dataset, first, we develop a method to split up $-S$ images where the left and right pairs are in the same image via a single simultaneous shot of the eye. We cut the images by the vertical middle line.

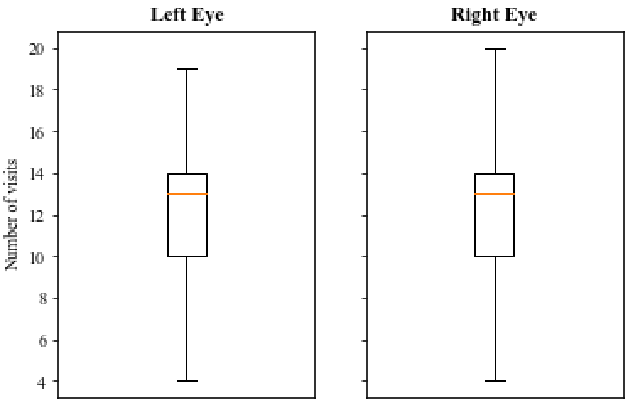

Figure 2: Median and interquartile range for number of visits among all the patients in this cohort

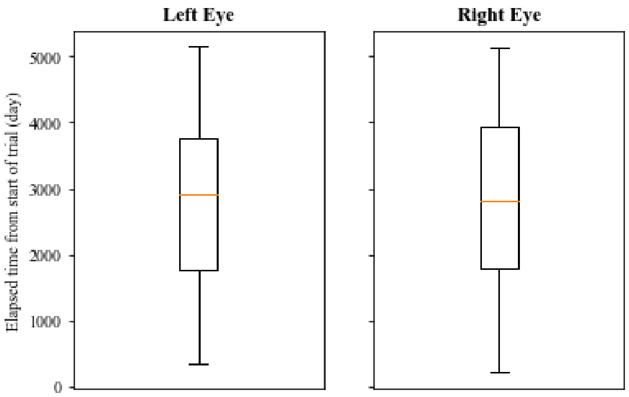

Figure 3: Median and interquartile range for time when patients were reported positive among all that have developed POAG in this clinical trial, in days since the beginning of the trial

Since we wanted to keep the central retina part of the image, we cut out the peripheral black border. For $-L/R$ images, the original image size is 1502×1000 pixels, and $-S$ images have an original image size of 751×1000 pixels. In order to keep the area in the image corresponding to roughly where the central retina is, we preprocessed the images by cutting from its geometry center with a square of size length 350 (for $-L/R$ images) and 250 pixels (for $-S$ images), resulting in final image sizes of 700×700 and 500×500 pixels, respectively.

4.2.2 Data Splitting and Trackback Days. We randomly split our dataset into the training, development, and test sets. The training set is used to train the model. The development set serves as the held-out set to check whether the model is overfitting on the training data. The test set is used to make predictions on unseen data. The split is conducted on patient level, i.e., all the images belonging to one patient are split into one of the three subsets. Splitting on patient-level ensures that the network can see the full progress on patients' gradual development of glaucoma across time during training time. (Figure 4)

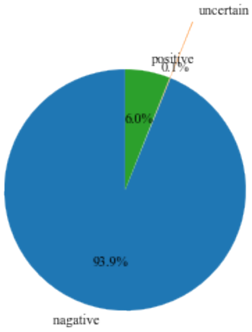

Figure 4: Distribution of positive/negative/uncertain labels of dataset on image level

For each patient, we labeled their visits by Positive, Negative, and Uncertain. An uncertain label implies an early sign of glaucoma. We categorized these images as uncertain as the last negative visit might be very close to the positive visit, and it might be a sign of early development of glaucoma. The default traceback days in our experiment are 180 days. (Figure 5)

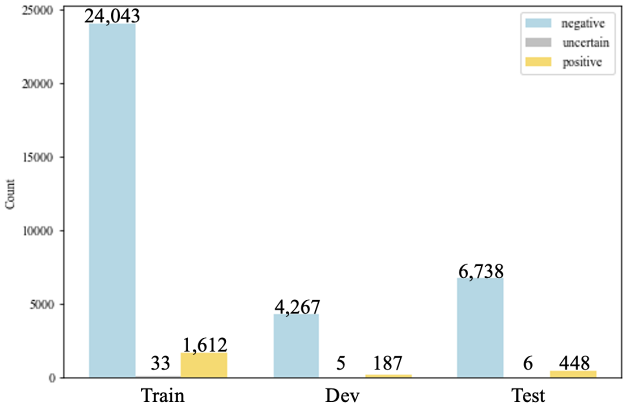

Figure 5: Number of images in train/dev/test set, with respect to their labels

4.2.3 Metrics. To benchmark the performance of these models, we are using them to predict on the test set and report F-score and the area under the Receiver Operating Characteristic curve (ROC AUC). We chose these two metrics as our dataset is unbalanced. The ROC AUC measure is a good measurement for model performance on imbalanced data, especially when there is a minority class(positive label in our case). We also use F-score since we care equally about recall and precision.

4.2.4 Implementation details. We implemented our experiment using Python and PyTorch library. All the training is conducted on the Scientific Computing Unit (SCU) provided by the Weill Cornell Medical College.

4.3 Results and Discussion

In this section we compared three different models, learning rate, split trace back days and image type and transfer learning techniques. We also discussed our prediction results on uncertain images.

4.3.1 Pretrained Models. We first compared three models, *VGG-16*, *ResNet-152*, and *DenseNet-201*. Table 1 shows that if we look at accuracy, *ResNet-152* and *DenseNet-201* give better results than *VGG-16*; if we look at ROC AUC, *VGG-16* and *ResNet-152* performs better; for F-score, *ResNet-152* and *DenseNet-201* works better. Since the dataset is highly imbalanced, ROC AUC and F-score would be more important metrics for the model. Overall, we concluded that *ResNet-152* has the best performance under this setting.

Table 1: Pretrained models benchmark, with 180 traceback days and 1×10^{-5} learning rate

Model	Accuracy	ROC AUC	F-score
VGG-16	0.7780	0.7681	0.2983
ResNet-152	0.9094	0.7621	0.4497
DenseNet-201	0.9210	0.7349	0.4518

4.3.2 Transfer learning techniques comparison. We compared two transfer learning strategies. Table 2 shows that fine-tuning the whole network achieved better AIC and F-score than using CNN as a fixed feature extractor.

Table 2: Transfer learning techniques comparison, with *VGG-16* model

Technique	Accuracy	ROC AUC	F-score
Fixed feature extractor	0.8526	0.5891	0.1958
Fine-tuning	0.7780	0.7681	0.2983

4.3.3 Comparison of learning rate. In order to find the best set of hyperparameters, we benchmarked transfer learning on the same pretrained model using three different learning rates - 1×10^{-3}, 1×10^{-4}, and 1×10^{-5}. As shown in Table 3, we find that 1×10^{-5} is the optimal learning rate we should use as it has the best performances in both ROC AUC and F-score.

Table 3: Traceback days benchmark, with *ResNet-152* and 1×10^{-5} learning rate

Learning rate	Accuracy	ROC AUC	F-score
1×10^{-5}	0.9094	0.7621	0.4497
1×10^{-4}	0.8760	0.7536	0.3816
1×10^{-3}	0.9377	0.5000	-

4.3.4 Traceback Days. We also experimented with different traceback days. We split the data with 180, 90, and 270 days. We used the best model of all three *ResNet-152* to experiment on different traceback days. Refer to table 4, if we compare accuracy, 270 days outperforms 180 and 90 days; if compared ROC AUC, 180 days will be the best threshold; if compare F-score, 270 days outperforms the other two. This may because we had more positive cases when 270 traceback days were used.

Table 4: Hyperparameter comparisons, with *ResNet-152* and 180 traceback days

Traceback Days	Accuracy	ROC AUC	F-score
180 days(default)	0.9094	0.7621	0.4497
270 days	0.9363	0.7212	0.4824
90 days	0.8869	0.7212	0.4156

4.3.5 Predictions on -L/R and -S Images. We also combined our testing results on $-L/R$ images and $-S$ images. Since these two types of images have different compressing sizes, we think it might affect the predicting results. Refer to table 5, the test results are from the same model, *ResNet-152* and 1×10^{-5} learning rate with 180 days split. We observed that $-L/R$ images have higher predicting accuracy, ROC AUC, and F-score. We believe the reason is, after we cut the $-S$ images, the sizes of the split images are smaller than $-L/R$ type images, which will lower the predicting performance.

Table 5: Prediction result comparisons between $-L/R$ and $-S$ images, with *ResNet-152* and 1×10^{-5} learning rate

Image Type	Accuracy	ROC AUC	F-score
$-L/R$ Images	0.9140	0.7819	0.4627
$-S$ Images	0.8929	0.7068	0.4084

4.3.6 Predictions on Uncertain Images. For the uncertain labeled data, we experimented on setting them to be positive and negative. We found out that our model predicts more accurately when they are set to be negative. This suggests that most of these uncertain images are negative.

5 CONCLUSION AND FUTURE WORK

In this project, we applied deep learning models to dectect primary open-angle glaucoma from fundus photographs. Extensive experiments show that overall the best model is *ResNet-152* with a learning rate 1×10^{-5} and threshold 270 days. In Fall 2021, we plan to propose a deep learning-based survival model to predict glaucoma before onset. We hope that this project will have a positive impact on saving time and money and improving clinical decision-making.

REFERENCES

[1] 2007. Validated Prediction Model for the Development of Primary Open-Angle Glaucoma in Individuals with Ocular Hypertension. *Ophthalmology* 114, 1 (Jan. 2007), 10–19.e2. https://doi.org/10.1016/j.ophtha.2006.08.031

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

[3] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. 2016. Densely Connected Convolutional Networks. *CoRR* abs/1608.06993 (2016). arXiv:1608.06993 http://arxiv.org/abs/1608.06993

[4] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs.CV]

[5] Anshul Thakur, Michael Goldbaum, and Siamak Yousefi. 2019. Predicting glaucoma prior to its onset using deep learning. *bioRxiv* (2019). https://doi.org/10.1101/828681 arXiv:https://www.biorxiv.org/content/early/2019/11/02/828681.full.pdf

[6] Jeremy West, Dan Ventura, and Sean Warnick. 2007. *Spring Research Presentation: A Theoretical Foundation for Inductive Transfer.* Retrieved August 05, 2007 from http://cpms.byu.edu/springresearch/abstract-entry?id=861