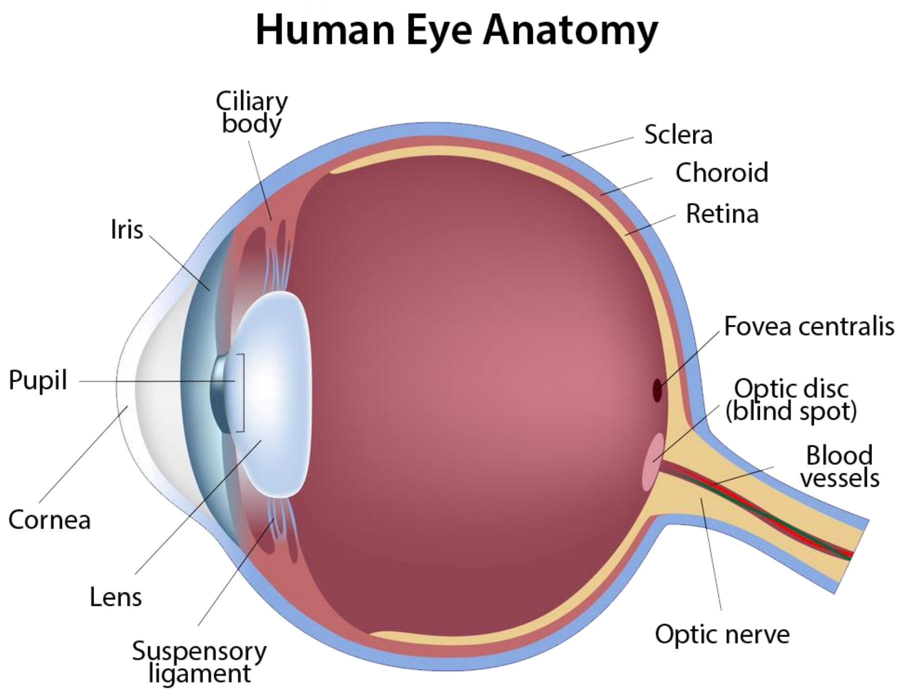
Capstone Project Report

Glaucoma Prediction



Summary

This report analyzes survey data to understand the cost and burden of the Glaucoma chronic condition and seeks to propose a possible diagnostic solution using image classification in order to reduce time and costs as well as increased accuracy of eye exam diagnosis of Glaucoma among adult optometry patients. *Source code can be found here.*

Background

Glaucoma is a general term for a number of eye conditions that progressively damage the optic nerve, consequently causing vision loss. Diagnosis of glaucoma is complex, but is often associated with elevated intraocular pressure, optic nerve damage, and reduction in visual acuity and visual field. Vision loss from glaucoma is permanent, but progression may be slowed or halted through early diagnosis and treatment. Glaucoma is not a common problem; but it is a costly and crippling disease particularly for individuals and the family of those diagnosed with it. According to an assessment of the economic burdens associated with Glaucoma, an estimated 3% of the global population over 40 years of age currently has Glaucoma, the majority of who are undiagnosed. The direct medical costs for those diagnosed with Glaucoma include ocular hypertensive medications, physician and hospital visits, and glaucoma-related procedures. The Indirect costs include lost productivity, days missed from work, and the cost borne by caregivers such as family and friends. Direct cost estimates for approximately 2 million US citizens are 2.9 billion dollars each year for those whom have been diagnosed, to say the least for those whom are undiagnosed. For those receiving treatment for glaucoma the estimated average annual incremental cost is 137 dollars per patient per year; a cost that only increases with severity of disease. From my brief investigation of the data I found that the overall prevalence to be anywhere between 4.8 and 6.6 percent in the United States among adults 40 years or older. Assuming there are approximately 154,911,104 adults over 40 in the United States an estimated 7,435,732.992 to 10,224,132.864 adults per year suffer from Glaucoma.

Project Overview

Problem

Using a limited dataset of retinal images create a classification system capable of distinguishing between incidents of diagnosed Glaucoma and from normal undiagnosed retinal images.

Client

Cambia Health is looking to hire an outside consultant use Cambia's collection of retinal images to help develop a machine learning algorithm to ultimately reduce medical practitioner time diagnosing patients with Glaucoma.

Approach

For basic population estimates for the report I downloaded data from CDC and analyzed the Behavioral Risk Factor Surveillance System in order to report the scope and burden of the chronic condition. For model development and training I used data downloaded from Harvard Dataverse which was seemingly already intended to be used for machine learning applications; however, at the time I was unaware of any publications using the dataset for model development. In order to complete my task I knew I wanted use some kind of convolutional model for classification, because object recognition in images is what CNN's were designed for[1], but unfortunately after downloading the data I found that the dataset was small in size so I knew that I was going to have to build a model that was deep in order to get the most out of each image.

Deliverables

Source code with presentation slides (Jupyter Rise) will be displayed in Github as well as a series of Medium blog posts showing my procedures from development to deployment of the model alongside a video presentation of the report and my methods.

---

[1] *https://en.wikipedia.org/wiki/Convolutional_neural_network*

Data Wrangling

Neither the image dataset nor the telephone survey required much data cleaning. For the telephone survey I had to create a calculated variable in order to flag the responses I was interested in investigating for Glaucoma; this included properly investigating and coding NaN values and distinguishing them from "Refuse" to respond and "Don't Know" options. Furthermore, workings with weighted analyses are also particularly tricky to do in Python since available statistical APIs are poorly documented on how to work with complex weighting schemes so there was some trial and error involved in order to return the correct or near correct values. There were additional difficulties working with multi-year surveys since typically not all BRFSS surveys always have consistent question design; requiring specialized attention to detail to make sure the responses match.

For the image dataset there wasn't any data wrangling involved aside from correctly specifying locations to store the images. The image set was provided in the form of a zip file that had the 'normal' control images, 'early stage', and 'advanced stage' images. It is not typically a good practice to have such a small dataset to be used for image classification; since I was dealing with a small set rather than trying to classify into three distinct categories I chose to simply classify to 'Glaucoma' or 'Not Glaucoma'. My reasoning is statistically speaking the greater number of units we try to classify will increase the error for each additional level of classification; therefore I chose to keep the classification binary to avoid such problems since the dataset was small anyways. This required moving images from the 'Advanced' and 'Early Stage' Glaucoma into a separate folder marked as 'Cases' and the 'Normal Controls' to simply 'Controls'. After separating the image files into two distinct folders in the appropriate folder locations I wrote a class object and a number of functions that randomized the images, performed image augmentation, and split them into appropriate train-test splits (70%, 10%, 20%) into training, validation, testing, and augmentation folders[2]. Lastly, I created a Dockerfile so I could work on training my algorithm in a closed development environment that can be potentially utilized by Amazon AWS Sagemaker.

[2] *Image augmentation is the act of creating new images by adding rotation, flipping images, adding distortions, and introducing noise in order to increase the dataset size and introduce statistical noise and thereby increasing generalizability of the trained model.*

Modeling

The ResNet Model

For classification of retinal images I have chosen to use a ResNet model as described by Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun in ***Deep Residual Learning for Image Recognition (2015)*** and Adrian Rosebrock in ***Deep Learning for Computer Vision with Python (2017)***. The ResNet is a famous and commonly used model for image classification that won the ILSCRC2015 & COCO 2015 image classification challenges in 2015. In fact the model is so useful and used so often, despite building the model myself as part of the project; the model actually comes pre-built as part of the Keras pre-trained model package.

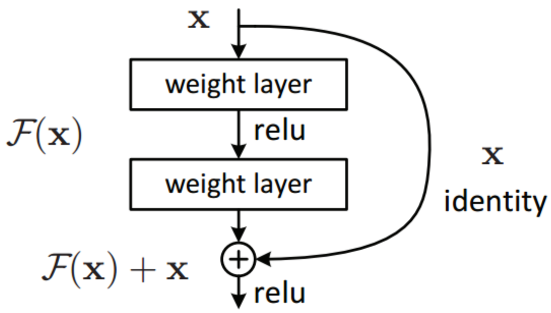
For an excellent detailed explanation of the ResNet model, checkout Gracelyn Shi's Medium blog post: https://towardsdatascience.com/implementing-a-resnet-model-from-scratch-971be7193718

Why use a Residual Network (ResNet) model?

In traditional Deep Neural Networks during training you typically run into problems during backpropagation[3] while using gradient descent optimization; where the data is fed from the final layers back into the first initial activation layers. As the loss function (error/residual) with respect to the weights becomes updated certain activation functions such as the sigmoid function, constrain the input space between 0 and 1. Since the gradient in a Neural Network is a product of the gradients preceding it, multiplying two smaller numbers returns an even smaller number. The small change in the gradient causes training times to be painfully slow, prone to error, and adding a large number of layers to become too costly and prohibitive.

Residual Networks provide a convenient solution to the **'vanishing gradient'** problem. By skipping (shortcut connections) the training of one or more layers and creating a **residual block** then we can add the residuals together in what is called a **bottleneck** prior to moving to the next residual block.

[3] ***Backpropagation****: Moving from later stages and fed back through the Neural Network to the initial layers*



Since residuals are being added together $H(x)=F(x)+x$ *approximates the residual function* $(H(x)=F(x)-x)$.

Where: $F(x)=W(2)*\mathrm{relu}*(W(1)*x+b(1))+b(2)$

Model Architecture

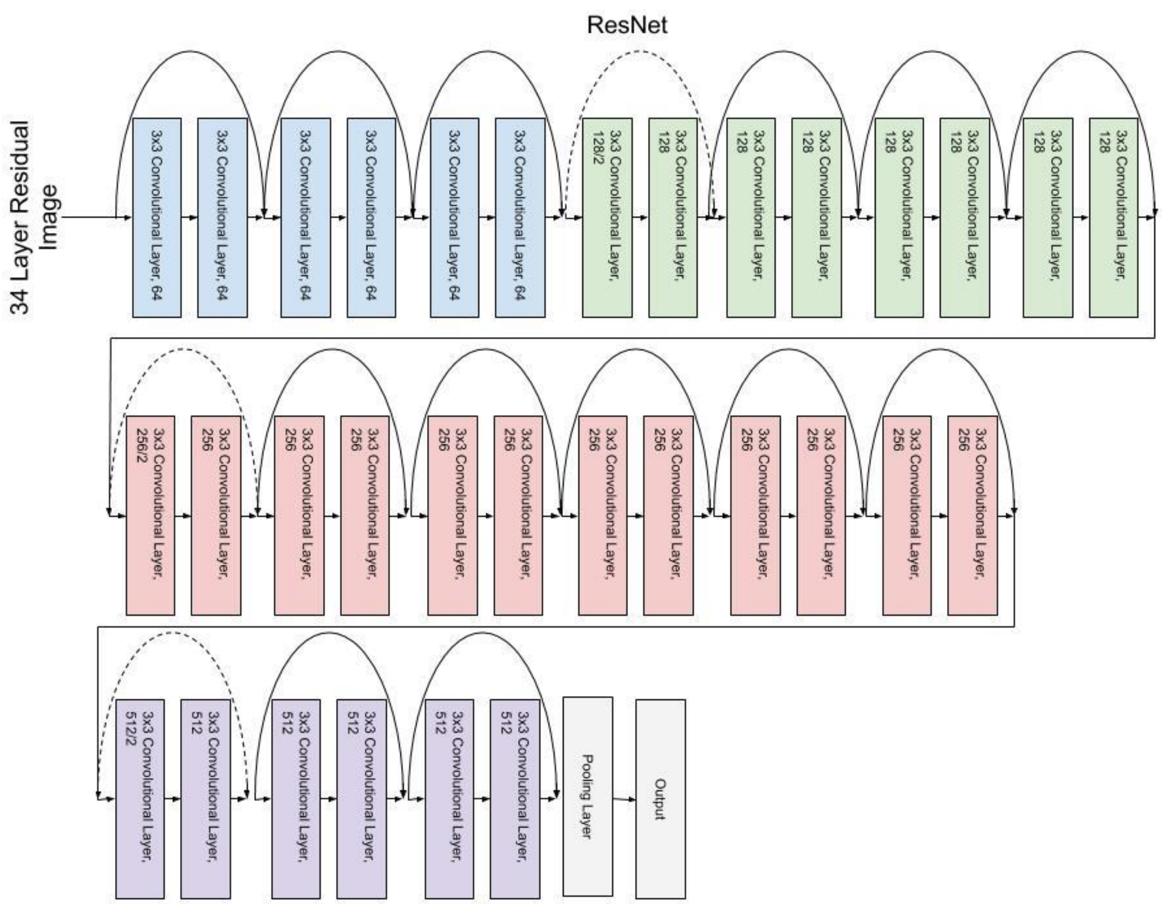
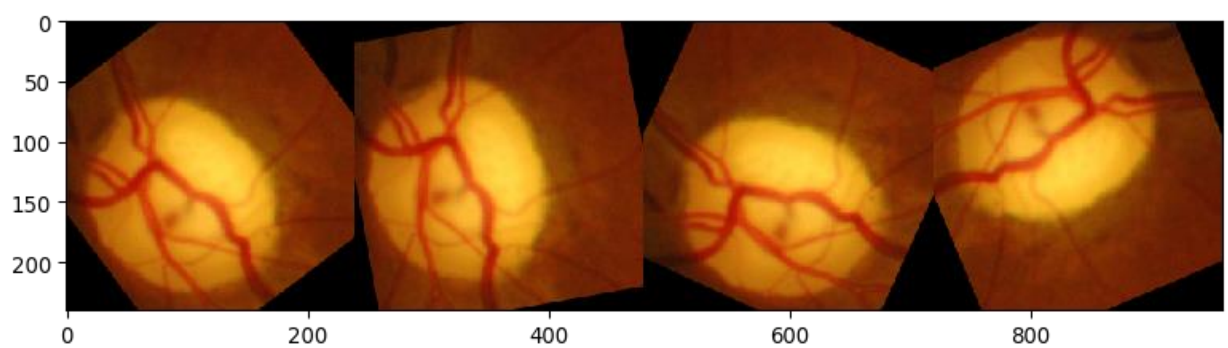


Image Augmentation

Because of the relatively small dataset size I had decided it was appropriate to try applying image augmentation to the data. I had played around with image augmentation in the past by randomly rotating images and then super-imposing them on random backgrounds, so it didn't seem to me to be a leap in applying the process to the retinal images. Sadly, only after I added the function to do this operation in my data preprocessing class object I found out that Keras already made image augmentation as a regular feature making my bit of code a moot point, but I kept piece of code in there because I put some time into it and maybe someday I will find some use for it. Below we can see an example of the output of some of the code for image rotations from the code I wrote, it also gives an illustration of what is happening with the images during augmentation. By rotating the images and adding noise we can increase the size of the dataset meanwhile maintaining generalizability for the model.

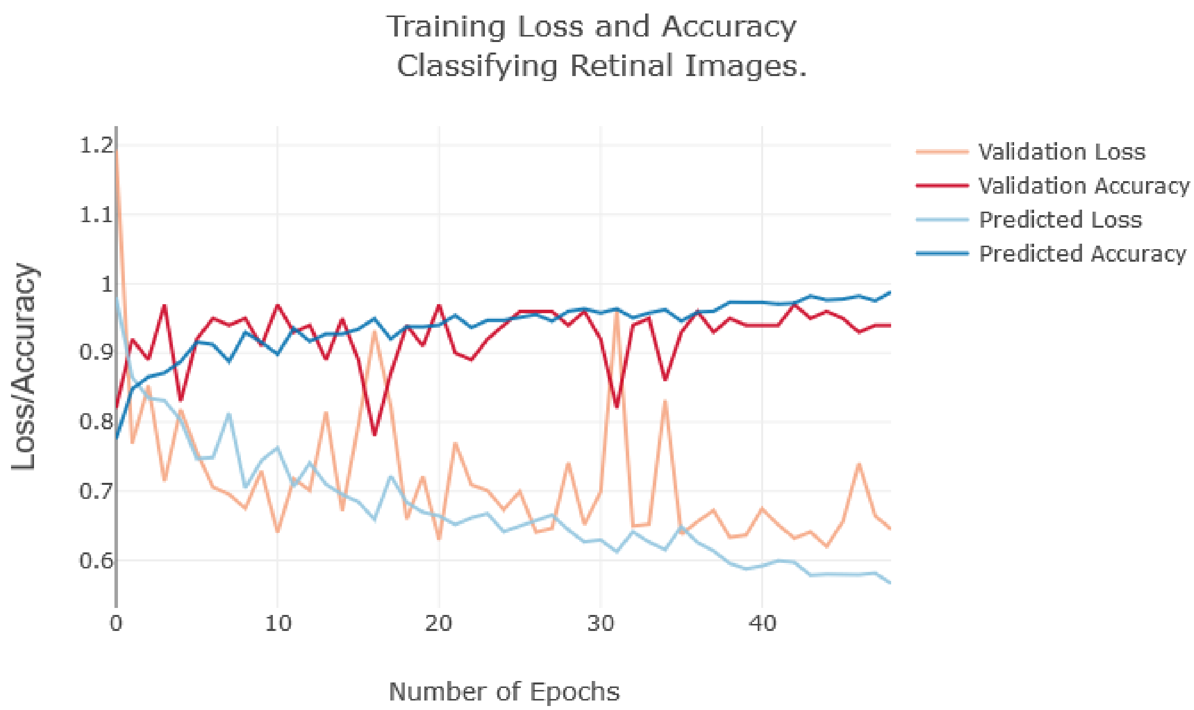


For my training I deliberately chose to only augment twenty percent of the images in order make sure I wasn't biasing the data and causing overfitting; which I already knew was going to be a problem for me since I was working with a small dataset and using a model type prone to overfitting.

Results

The training set had 80% of the data to work with and the rest was equally split between validation and testing sets with 20% of the data being augmented. After a fair amount of experimentation the hyperparameters I had decided to use were 49 epochs, with batch sizes of 50, and a learning rate of 0.02 and regularization parameter (lambda) of 0.0001.

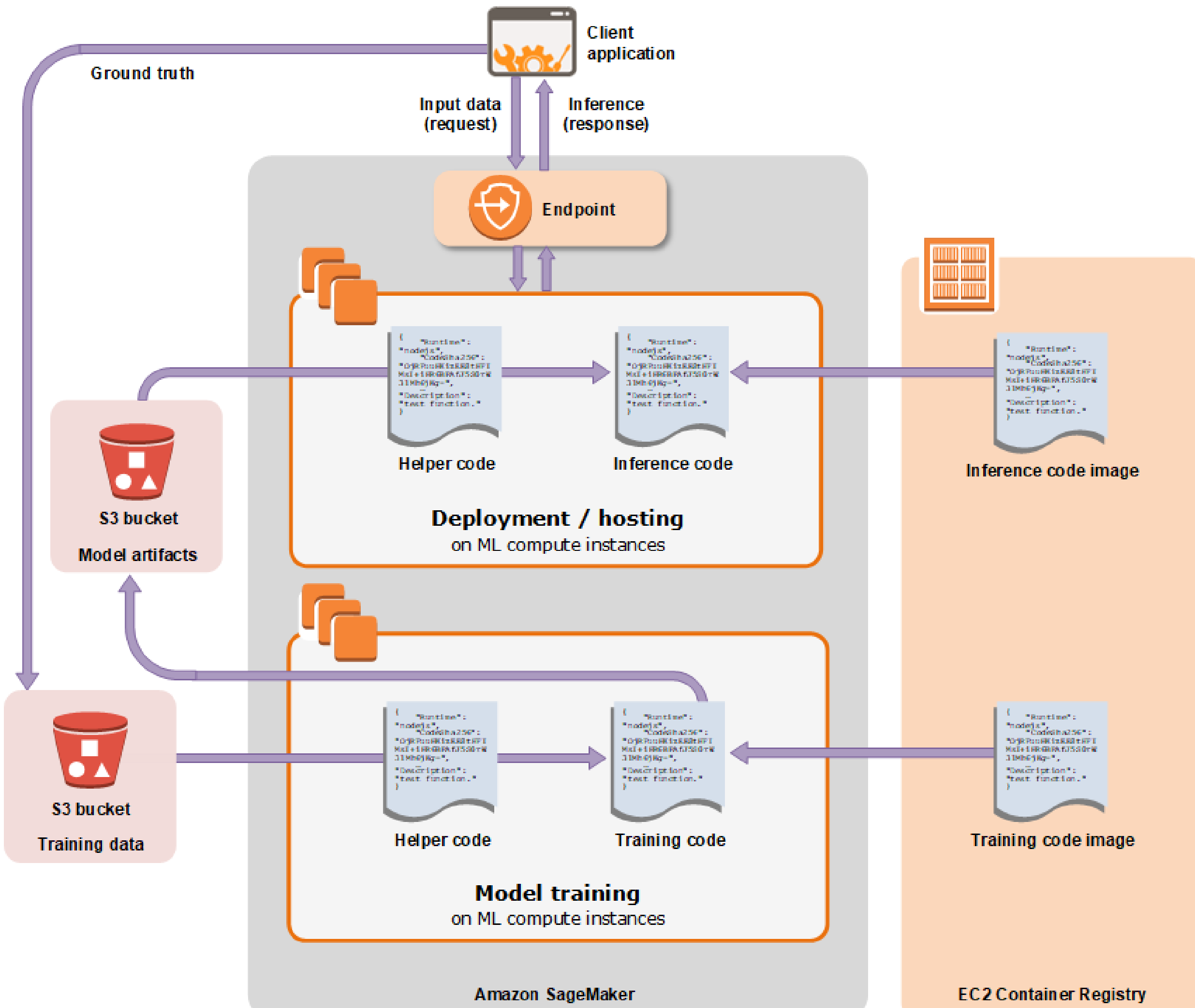
The results seem to look pretty good. I suspect there may be some overfitting happening with the data, but when I checked the model to predict an image after training the results seem to work fine. Unfortunately I don't have access to any novel retinal images to test the model against to really get a true sense of the generalizability of the model but the printouts seem to look fairly convincing.



	precision	recall	f1-score	support
cases	0.92	0.95	0.94	8
controls	0.97	0.96	0.97	163
avg / total	0.96	0.96	0.96	251

(Additional) Model Deployment

After I achieved my desired results I saved the model architecture and weights in order to move on to the next stage; model deployment. For model deployment I decided to use AWS Amazon Sagemaker to and get it to an AWS 'Endpoint' stage. An [AWS endpoint] allows you to provision resources and deploy models; essentially once an endpoint is properly setup you should be able to connect it to an API call for your client application to run inferences on a dataset.



Above shows the structure of the AWS hosting services.

I don't really want to get too much into any details here because I feel doing so would be reaching beyond the actual scope of the project.

After I got results that I felt I was satisfied with my model, I [saved the model weights and architecture] on my local machine and logged into AWS and fired up a **ml.m4.xlarge**; which has 4 virtual cpus and 16 gigs of ram which is roughly similar to my local machine's compute capability. Once the Sagemaker's instance was created I started a Jupyter Notebook with the **conda_tensorflow_p36** option and uploaded my model weights and architecture along with some example image files and managed to connect my model to an AWS endpoint. If you would like to see a detailed description about how I managed to do this; I have a much more detailed Jupyter Notebook on Github that walks you through the entire procedure.

Conclusion

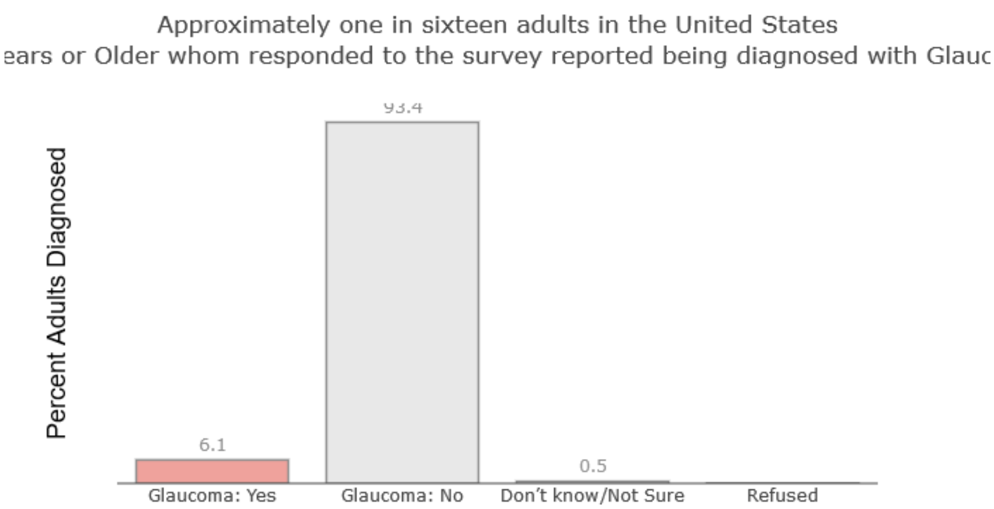
At this point I have walked through the entire lifecycle of the model from data acquisition stage to model deployment. Although what I have demonstrated was fairly simplistic I feel at least it demonstrates the challenges associated with accomplishing the task. There were definitely a number of pain-points that I ran into while I was working on this project; chiefly among them was working with Sagemaker and AWS. I feel they have an idea of an easy to use product but there were a lot of bugs that I felt I had to learn from trial and error in order to make things work; eventually I almost universally chose to use the easiest options I could but still turned out to be a daunting task. If I had my preferences I would like to have more data available to train the model on. I'd also like to play around with the architecture of the model as well and see if I can modify the performance of the model somehow; maybe I could change the way the bottlenecks worked or develop my own way of dealing with the vanishing gradient problem. I'd also like to create some kind of public facing web-interface to my model hosted on AWS as well so other people could play around with it; but I felt time didn't really permit that distraction. Lastly, I investigated hosting the model on AWS marketplace, which is another way people could have access to the model I developed, but time didn't permit me figuring out this last step of the process. Perhaps in the future I could add that step and show how I could connect 'Batch Transform' to the model hosted on AWS.

Basic Survey Analyses[4]

Single Year Prevalence

BRFSS 2010: Self-reported diagnosis with Glaucoma.

	Percent	Frequency	Weight_freq
Glaucoma: Yes	6.061600	1478.0	4.815537e+05
Glaucoma: No	93.401140	22774.0	1.077470e+07
Don't know/Not Sure	0.516753	126.0	4.687352e+04
Refused	0.020506	5.0	3.439010e+03
BLANK	1849.956937	451075.0	0.000000e+00
Total	1949.956937	475458.0	1.130657e+07



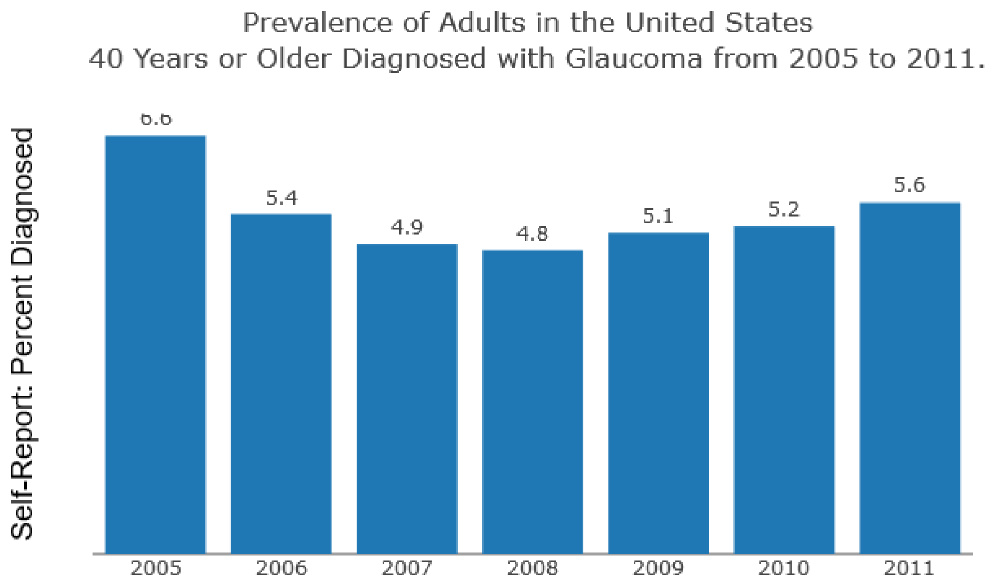
Explanation: In the year 2010 among those whom were 40 years of age that received the vision module and responded to the question on the survey, six percent or one-in-sixteen adults responded as having been diagnosed with Glaucoma.

[4] *For best viewing of my results and interactive graphs follow this link:* https://nbviewer.jupyter.org/github/pmleffers/Glaucoma/blob/3d07c76b765dde9c627960bddd442970ca532130/Glaucoma%20Analysis.ipynb

Multi- Year Prevalence[5]

Trend Graph

	Year	Data_Value
0	2005	6.635965
1	2006	5.389437
2	2007	4.913592
3	2008	4.818158
4	2009	5.093863
5	2010	5.199083
6	2011	5.572816

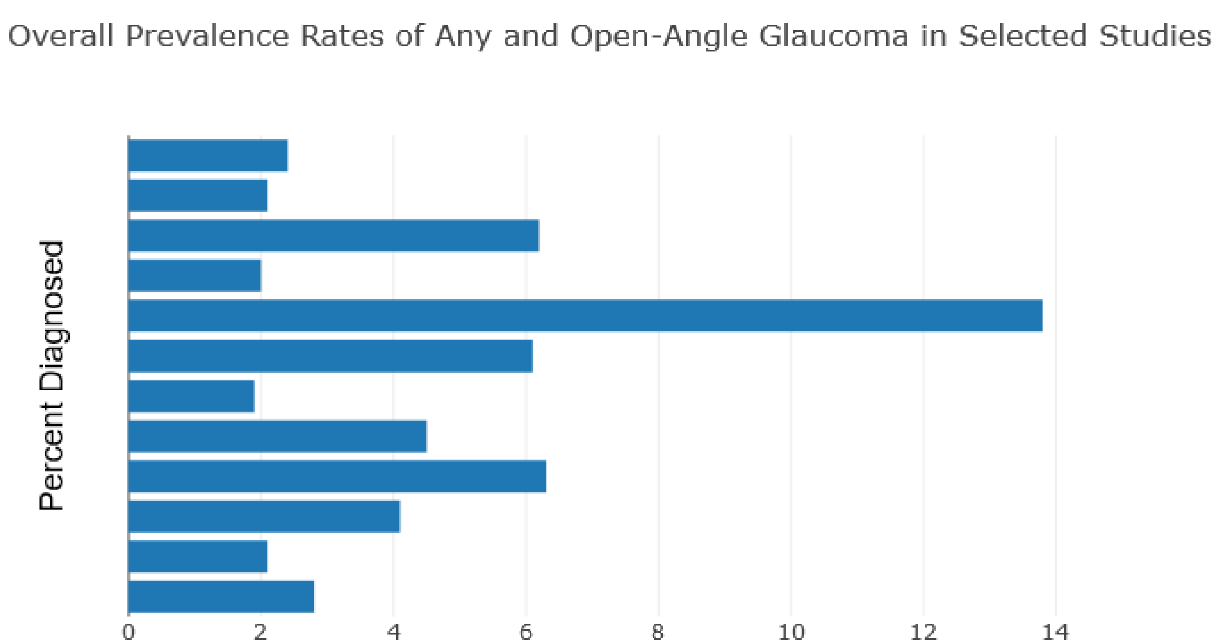


Explanation: From the years 2005 to 2011 the overall crude prevalence of Glaucoma nationwide appears to be relatively stable over time hovering between 4.8 and 6.6 percent among US adults over the age of 40.

[5] *For best viewing of my results and interactive graphs follow this link:* https://nbviewer.jupyter.org/github/pmleffers/Glaucoma/blob/3d07c76b765dde9c627960bddd442970ca532130/Glaucoma%20Analysis.ipynb

Meta-Analysis

	Author	Title	Data_Source	Sample_Size	percent
0	Park D, Mansberger SL, et al.	Prevalence of Age-Related Macular Degeneration...	Telemedicine Screening Program	424.0	2.8
1	Gupta P, Zhao D, et al.	Prevalence of Glaucoma in the United States: T...	NHANES 2005-2008	4798.0	2.1
2	Maa AY, Evans C, et al.	Veteran Eye Disease After Eligibility Reform:...	Atlanta VA Medical Center Chart Review	658.0	4.1
3	Cassard SD, Quigley HA, Gower EW, et al.	Regional Variations and Trends in the Prevalen...	Medicare Claims	NaN	6.3
4	Kim E, Varma R.	Glaucoma in Latinos/Hispanics.	LALES	6142.0	4.5
5	EDPRG	Prevalence of Open-Angle Glaucoma Among Adults...	EDPRG	NaN	1.9
6	Mansberger SL, Romero FC, et al.	Causes of Visual Impairment and Common Eye Pro...	Northwest AIAN	288.0	6.1
7	Lee PP, Feldman ZW, Ostermann J, et al.	Longitudinal Prevalence of Major Eye Diseases	National Long-Term Care Survey	NaN	13.8
8	Quigley HA, West SK, Rodriguez J, et al.	The Prevalence of Glaucoma in a Population-Bas...	Proyecto VER	4774.0	2.0
9	Haronian E, Wheeler NC	Prevalence of Eye Disorders Among the Elderly ...	UCLA MEC	431.0	6.2
10	Klein B, Klein R, et al.	Prevalence of Glaucoma: The Beaver Dam Study	BDES	4926.0	2.1
11	Tielsch JM, Sommer A, et al.	Racial Variations in the Prevalence of Primary...	BES	5308.0	2.4



Citations(s)

Varma, Rohit et al. "An assessment of the health and economic burdens of glaucoma." American journal of ophthalmology vol. 152,4 (2011): 515-22. doi:10.1016/j.ajo.2011.06.004
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3206636/

Coleman AL, Kodjebacheva G. Risk factors for glaucoma needing more attention. Open Ophthalmol J. 2009;3:38–42. Published 2009 Sep 17. doi:10.2174/1874364100903020038
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2759104/

NORC | Published Examination based Prevalence of Major Eye Disorders. JUNE 22, 2018.
http://www.norc.org/PDFs/VEHSS/EyeConditionExamLiteratureReviewVEHSS.pdf

** Population estimates collected from Census Reporter*
https://censusreporter.org/profiles/01000US-united-states/