Diagnosing Diabetic Retinopathy

Project Category: Life Sciences

Blanco, Ignacio
iblanco@stanford.edu

Gupta, Jai
jaigupta@stanford.edu

Kaur, Amrita
amritak@stanford.edu

October 3, 2020

1 Introduction

Diabetic retinopathy (DR) eye disease is a medical condition where the retina is damaged due to diabetes. It is estimated to affect over 93 million people worldwide, and it is the leading cause of blindness in the developing world. Additionally, 80 % of those patients who have had diabetes for 20 years or more develop DR eye disease. This disease is hard to diagnose, because it is initially asymptomatic, until it is too late and the patient loses their vision. Currently, the detection of DR is a time-consuming and manual process that requires specialists and several days to yield a diagnosis, which leads to lost follow up, miscommunication, and delayed treatment. However, as the number of individuals with diabetes continue to grow, the current methods and techniques become insufficient.

The need for an automated and thorough detection process has been pointed out, and previous efforts have made some progress with image classification, pattern recognition, and machine learning, using digital color fundus photographs. Nonetheless, we propose using cutting-edge techniques that can optimize and improve the existing methods to obtain a more accurate and faster diagnosis. After obtaining results, we will compare our models with existing methods to finetune and optimize the decision-making process of the algorithms.

2 Dataset

To diagnose the presence and severity of diabetic retinopathy, we will be considering the following dataset. This dataset [1] contains 82GB of high-resolution retina images taken using fundus photography. For each patient, photos of both the left and right eyes are included. The images in this dataset come from a wide variety of clinics over an extended period of time, and therefore were taken using different types of cameras, greatly influencing the visual appearance of the photos. While some photos display the eye as one may see it anatomically, others are inverted due to the use of a microscope condensing lens.

Within the labeled training data (32GB), a clinician has rated the presence and severity of diabetic retinopathy (DR) in each eye using the following scoring system:

- 0 for no DR
- 1 for mild DR
- 2 for moderate DR
- 3 for severe DR
- 4 for proliferative DR

Using the unlabeled data (50GB) and our models, we will be seeking to input the retina images and output a score prediction from 0-4, corresponding to the presence and severity of diabetic retinopathy within the eye.

3 Modeling

3.1 Related Work

There have been many attempts in this area. Kaggle [5] hosts this dataset and some of the leading solutions use Finetuning on top of Inception V3, Resnet-50 and VGG and the use of attention in the final layers to focus on the right part of the image.

3.2 Method 1: Fine Tuning ImageNet models

First proposal is to fine tune existing models that have been trained on ImageNet dataset on our dataset. Training a model on a large dataset then fine tuning it to a smaller dataset performs great due to transfer of knowledge from the large dataset. This sort of transfer learning has shown great result in the field of both language and vision, and we will try to experiment with the same by finetuning the penultimate layers.

Experiments
There are three such candidate models that we wish to try and compare:

- Inception V3
- Resnet-50
- Resset-101

For each of the above models, we plan to use existing checkpoints from publicly available sources and then perform the finetuning. The aim here would be to compare the complexity (no of operations/parameters) of the model and the achieved accuracy after the fine-tuning.

If time permits, we will also try comparing the effect of fine-tuning multiple penultimate layers and observe how that affects accuracy.

3.3 Method 2: Generative approach for modelling from unlabeled dataset

One thing to note about this dataset is that we have a large amount of unlabelled data as well. The question we ask here is whether we can make use of this additional unlabelled dataset to improve our prediction.

GANS provides a powerful technique to model the distribution of a dataset to differentiate between fake (generated) images and real images (from dataset). While this does not directly help us with our problem, we hypothesize that this can help provide an inductive bias in the model that can help in the real problem.

Experiment
Essentially, we will have a generator, and a discriminator. The generator will produce fake images, and the discriminator has to produce a probability distribution over $k+1$ classes. Where k is the number of classes from the original dataset, and the additional class denotes that the provided image is fake (created by the generator). We base our work on the Semi-Supervised learning approach from [2]. [3] achieved at the time state-of-the-art result using this method, including MNIST. Hence, we think this might be a good direction to explore.

While Method 1 above has its own potential, we think this method might have significant advantage and we will use results from Method 1 as baseline to compare against.

Metrics
For the experiments with pretrained models, as well as the experiment with GANS, we will use accuracy as our primary metrics which determines the number of cases where we guessed the class correctly. To better understand the results, we will also look at precision, recall and AUC metrics.

References

[1] EyePACs (2015, Feb). Diabetic Retinopathy Detection. Version 1. Retrieved October, 2020 from https://www.kaggle.com/c/diabetic-retinopathy-detection/data.

[2] Augustus Odena *Semi-Supervised Learning with Generative Adversarial Networks*

[3] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen *Improved Techniques for Training GANs*

[4] Mayo Clinic (2018,May) Diabetic Retinopathy – Symptoms and Causes. Retrieved October, 2020 from https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes.

[5] Kaggle *https://www.kaggle.com*