Improving performance on reduced tomography datasets

Project Report, FourthBrain

21 February 2021

Felipe Caballero, George Prounis, Gabriel Marcelo Santos Elizondo

SME:Sarfaraz Hussein, Carson Kent

Abstract

The COVID pandemic has shown the need for rapid-response and development in medicine. As new pathogens and diseases spread and reach hospitals and clinics, we gain valuable information, yet their analyses lag behind. Professionals lack the time to categorize, annotate, and publish each case. Tools from machine learning such as semi-supervised learning, transfer learning, and unsupervised-learning offer opportunities for early, expedited, analyses of these pathogens.

In our study, we explore the use of transfer learning and semi-supervised learning for the detection of retinal pathologies on a reduced set of data, simulating a medical scenario in which there is a lack of annotated data for new pathogens or diseases. Our analyses were conducted using Kermany, Zhang, and Goldbaum's OCT dataset (DOI: 10.17632/rscbjbr9sj.2). For our project, we split this dataset of 84k images into three pieces, a 90% unlabelled set (we removed the labels), a 5% test set, and a 5% train set which we further split into 4% train and 1% validation.

We created three different models, all of them trained on the last 34 layers of Resnet50:

- The baseline model uses 5% of the data for training.
- A COVID/OCT model. We trained it initially using the COVID dataset and used that as transfer learning to train the OCT 5% dataset.
- Finally, we used the baseline model to predict labels for 5% of the unlabelled set and then ran those newly predicted labels as additional training for the baseline to complete our semi-supervised approach.

The Semi-supervised approach showed an improvement over the baseline of 0.90% for f1 weighted average (f1 score reached: 92.80%). Both the Covid transfer learning and the semi-supervised approaches showed class specific recall improvement for Drusen: 2.34% improvement using the covid transfer learning and 4.68% improvement using the semi-supervised approach. Our findings show that semi-supervised learning can be used effectively to improve performance of a medical model and also suggest that using similar images to the target can help improve class specific recall.

Table of Contents

Team, about us



Felipe Caballero felipe@caballero.co	George Prounis george.prounis@gmail.com	Gabriel Marcelo Santos-Elizondo g.santose4@gmail.com

Felipe

Since I was little I felt an attraction to technology, I learned to program on a Sharp calculator and since then I've been committed to develop technology driven solutions. I graduated from Universidad Adolfo Ibañez in Santiago de Chile. After finishing university, I started a web development company with a friend and his sister. We did many websites for different companies. I led a team of 5 people from sales to development. We also did a couple of back office projects to keep track of people's information on call centers. I'm looking to start a career in Artificial Intelligence, I've done several courses on Coursera and FourthBrain is my hands-on approach before starting a new position.

George

After receiving my PhD in Neuroscience from Cornell University, studying developmental systems and social behavior, I did post-doctoral research on the neural basis of risk-taking and decision-making at UC Berkeley. Two years ago, I pivoted this decade of behavioral analysis into a career in data science and machine learning. I currently work as a data scientist & ML engineer for RedflagAI, a social

intelligence software start-up (https://www.redflagai.co/), and am grateful for the opportunity to strengthen my understanding of machine learning, from its theoretical foundations to deployment, as a FourthBrain student.

Gabriel

While completing my undergrad in Biology at Cal Poly SLO, I got the opportunity to lead a research team in the study of Northern Elephant Seals. During this time, I developed embedded devices and machine learning procedures to facilitate and hasten our data acquisition and analyses. I received further embedded systems training from Thom Maughan at the Monterey Bay Aquarium Research Institute (MBARI) where I open sourced a built guide for an underwater great white shark camera. The Shark Cafe Cam (https://www.mbari.org/intern-technology/). As my thirst for finding creative solutions to ecological problems grew, I sought to develop a greater understanding of machine learning. That drive led me to Fourth Brain and to getting hired at SWCA as a Machine Learning Engineer. Now, I seek to continue that path of learning in a new position with a company passionate about innovation.

Data Overview

Retinal OCT

From the original Samsung project proposal, the data came from Kaggle[1]. We found that this dataset was twice the advertised size and we looked for alternatives, we ended up using a dataset from Mendeley[2] which was 5.8 GB, this dataset contains the same images but not twice the size. The dataset comes from 'Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification' (Kermany et al. 2018). The Mendeley dataset we used (as the Kaggle one) contains a total of 84,484 files, of which 7,357 are duplicates. We ended with 77,127 usable files. We used a Pandas DataFrame to store this files' information and to do further work.

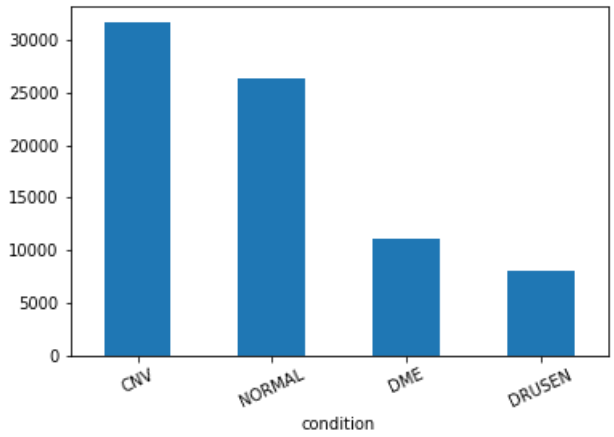

Figure 1: Bar graph depicting amount of images per OCT retinal pathology class.

We observed a slight class imbalance between CNV and NORMAL vs. DME and DRUSEN. We also reviewed images from each class to better understand the data and visually seek variation between pathologies. See Figure 2 for examples.

[1] https://www.kaggle.com/paultimothymooney/kermany2018

[2] https://data.mendeley.com/datasets/rscbjbr9sj/2

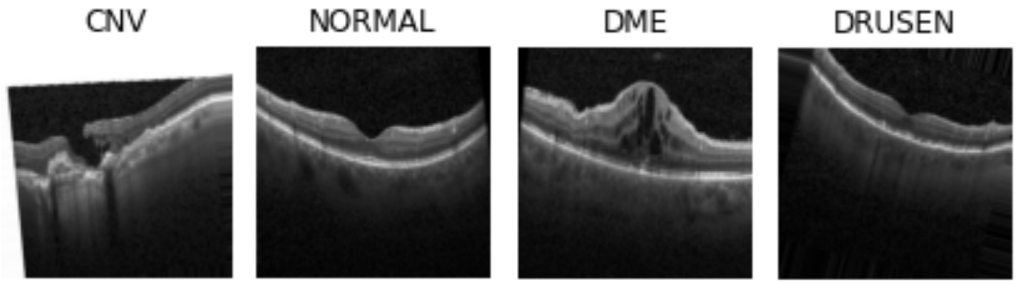

Figure 2: Example images pulled from the dataset for each OCT retinal pathology class.

We did several models (discussed later), in all of them we used 5% of the data as test set. For the full baseline we used the remaining 95% for training (of which 20% was used for validation). For all the dataset reduced models we used 5% for training (of which 20% was used for validation).

COVID-CTscan

The data was retrieved from a Kaggle dataset named 'COVID-CTset: A Large COVID-19 CT Scans dataset'[3]. The original dataset comes from 'A fully automated deep learning-based network for detecting COVID-19 from a new and large CT scan dataset' (Rahimzadeh et al. 2021). The subset we implemented was proven to achieve a 98%+ accuracy using a modified ResNetV2, so we expected it would be adequate for our application. This subset includes 11,621 total images (COVID: 2,155 (95 patients) / Normal: 9,466 (282 patients)).

[3] https://www.kaggle.com/mohammadrahimzadeh/covidctset-a-large-covid19-ct-scans-dataset

Objective

Improve performance on reduced dataset

Our objective with this project was to improve the performance of a baseline model trained on retinal OCT images while simulating a low data scenario. We had the following ideas on how to achieve this:

1. **Transfer Learning:** Look for similar datasets that we could use to train our model that would transfer well to the retinal OCT images model.
2. **Semi-Supervised Learning:** Use label spreading to produce new labels for unlabelled data, then use that newly labeled data to re-train our baseline model.
3. **Combine both:** Our tentative third idea was to combine both aforementioned methods if we saw both improved performance metrics from the baseline.

We did not end up combining the methods (idea 3), as performance of idea 1 did not show improved results.

Model Architecture Selection

Model	Val_acc	Val_loss	f1_score
Resnet50_.01_Adam_8	0.8396	0.4423	0.734184
Inception_V3_.01_Adam_8	0.8108	0.5646	0.788707
VGG16_.01_Adam_8	0.8234	0.5583	0.825868
Resnet50_.01_SGD_8	0.8126	0.5229	0.769536
Resnet50_.01_Adagrad_8	0.8126	0.5756	0.774433
Resnet50_.1_Adam_8	0.8342	3.7931	0.779204
Resnet50_.001_Adam_8	0.836	0.4436	0.806301
Resnet50_.01_Adam_2	0.836	0.5089	0.767981
Resnet50_.01_Adam_16	0.8414	0.5441	0.770273
Resnet50_.01_Adam_32	0.8342	0.5345	0.773052
Resnet50v2_.01_Adam_16	0.8378	0.4695	0.770538

Figure 3: Summary of model selection tests. Models vary by architecture and hyperparameters.

We started exploring model architectures that worked well on medical image slices for different teams. In Mallick's post[4] on LearnOpenCV 'Transfer Learning for Medical Images,' we found three models to be the most successful: Vgg16, Inceptionv3, and Resnet50.

We then extracted 4% of our total dataset for comparing these three models. We chose 4% as the lowest amount of data that would highlight differences in performance and make it easier to select a model. This low amount of data makes training the models faster.

We first tested the three model architectures against each other and compared their validation accuracy after 5 training epochs. We found Resnet50 performed the best.

[4] https://learnopencv.com/transfer-learning-for-medical-images/

Once we had an architecture, we then tested varying hyperparameters and selected those with the highest validation accuracy after training.

The best validation accuracy was obtained using:

- Resnet50.
- Optimizer: Adam.
- Learning rate: 0.01.
- Batch size: 16.

We chose this architecture and hyperparameter combination as our baseline.

We followed a scientific approach which means we tried changing as little variables as possible throughout our work. All the models we tried used:

- Data augmentation.
- Pre processing (Resnet50 needed).
- 30 epochs.

Models

Baseline

Having selected an architecture and hyperparameters we moved on to training the baseline model. We froze all but the last 34 layers of the model (the last convolutional block), the rest of the layers use the weights from Resnet50's Imagenet classification. We removed Resnet50's last dense layer, we added our own with a four neuron output to classify our 4 conditions.

We used 5% of the data as test set and 5% of the remaining 95% for training (0.05 * 0.95). This was done to simulate a low data scenario. 20% of the training data was used for validation (0.02 * 0.05 * 0.95).

We evaluated this model on the 5% test set and recorded performance.

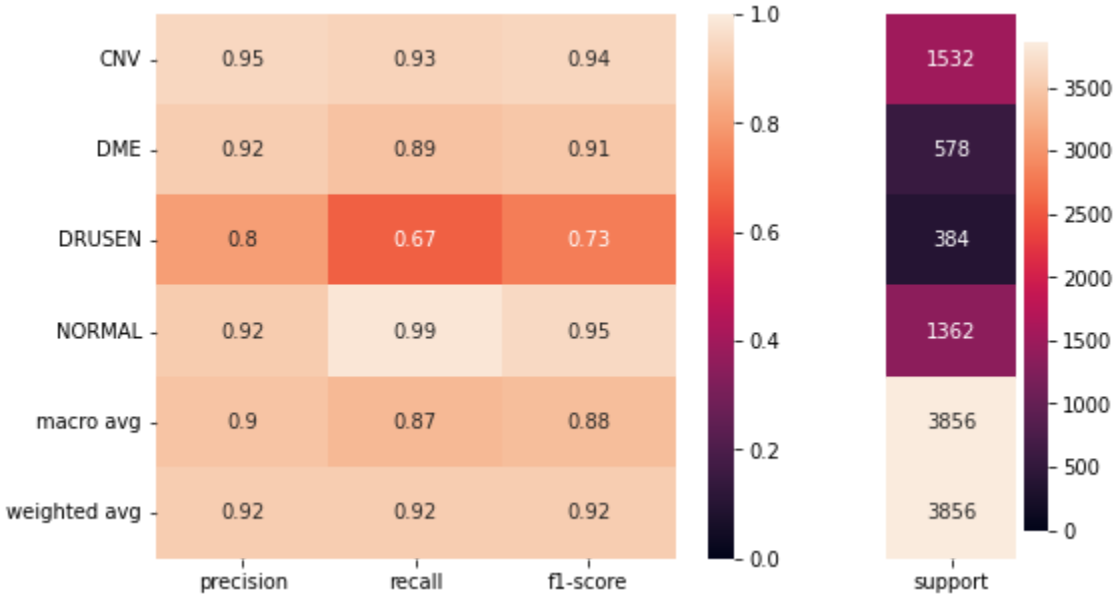

Figure 4: Heatmap of performance of 5% baseline on 5% test set.

Experimental A: Transfer Learning

We first looked to improve our baseline model by performing transfer learning from a model of matching architecture. We first created a model to classify a two-class

COVID-19 tomography dataset (described previously), with the final dense layer being replaced to predict two classes (Normal Lungs or with COVID). This COVID-CTscan Baseline Resnet50 was trained with a 20% validation split, and metrics were tested on 5% of the data, achieving 94% accuracy.

We used the COVID model to transfer learning into an OCT model (changing the last dense layer of 2 outputs to 4 outputs for the OCT conditions). We kept all but the last 34 layers frozen.

For the training on the OCT data we used 5% of the data as test set and 5% of the remaining 95% for training (0.05 * 0.95). This was done to simulate a low data scenario. 20% of the training data was used for validation (0.02 * 0.05 * 0.95).

In Figure 5 the results are shown.

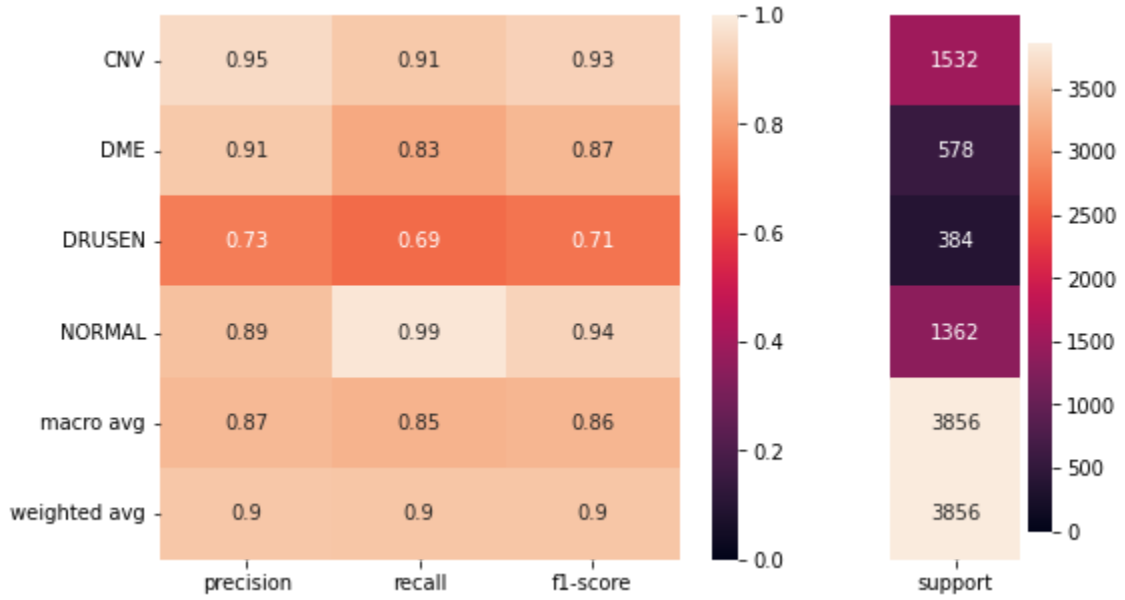

Figure 5: Heatmap of performance of COVID->OCT transfer model on 5% test set.

Experimental B: Semi-Supervised Learning

We split the dataset of 84k images into three pieces, a 90% unlabelled set (we removed the labels), a 5% test set, and a 5% train set which we further split into 4% train and 1% validation.

First we generated an unlabelled dataset by taking 5% of the data from the 90% unlabeled set. We then removed the last softmax layer from our baseline model so it would output probability vectors for each class instead of just one class. Next, we

ran our baseline model on the unlabelled images to predict new labels. We mapped these extracted probability vectors to see their spread versus that of labeled images.

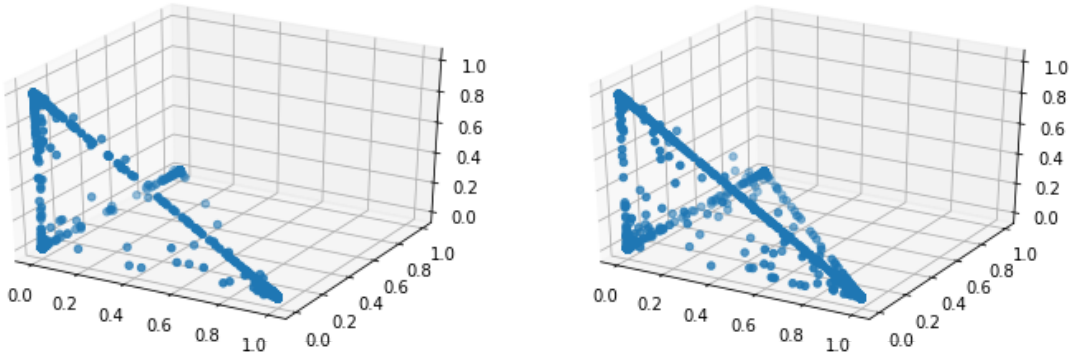

Figure 6: 3D graph of the probability vectors for the labeled data (left) versus the unlabelled data (right).

We can observe slightly messier vectors for the unlabelled images, which shows how there is more uncertainty in the label predictions.

We then used these probability vectors to fit a label spreading model from Sklearn. We used a KNN kernel, gamma of 25, and max iterations of 20. The generated labels were then concatenated with our train set labels. Before training the model, we unfroze the last 34 layers to repeat our procedure on the baseline and ensure unbiased testing. The concatenated labels were passed to our semi-supervised model alongside their respective images. We trained this semi-supervised model for 30 epochs and then evaluated it on the 5% test set.

The final average accuracy for the semi-supervised model was 92.82%.

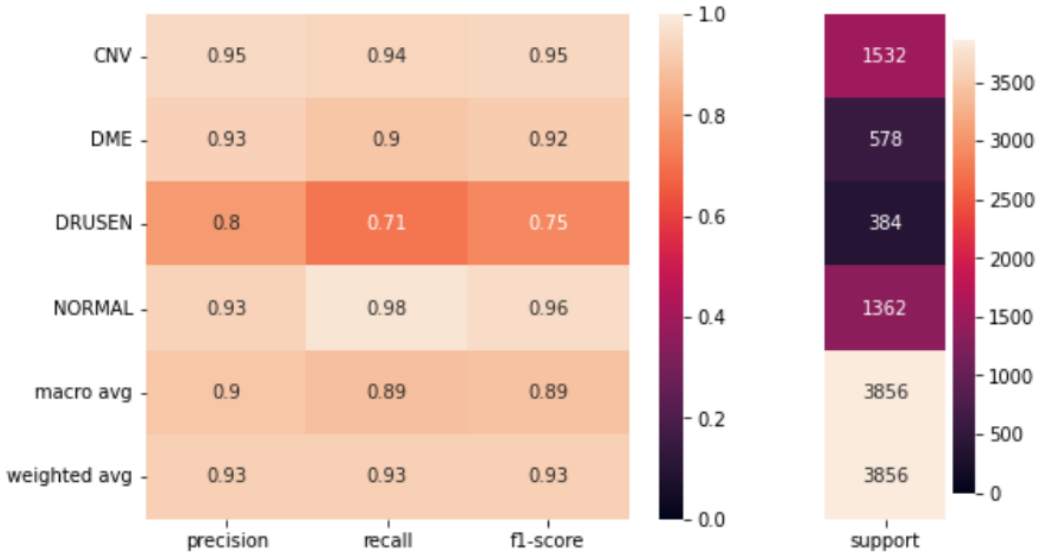

Figure 7: CNS Heatmap of performance of the semi-supervised model on the 5% test set.

System design

For the development we did the procedure:

- Exploratory Data Analysis, where we reviewed the data and found some duplicates.
- Created a baseline model using Resnet50.
- Create the COVID to OCT transfer learning model.
- Created the Semi Supervised model
- Created a deployment using the resulting models.

As a note, while developing we did try many more models and experiments but we mention here only the relevant ones.

For our deployment we created a website using Flask. We uploaded our three models to this website and it predicts a condition from within an image for each one of the models.

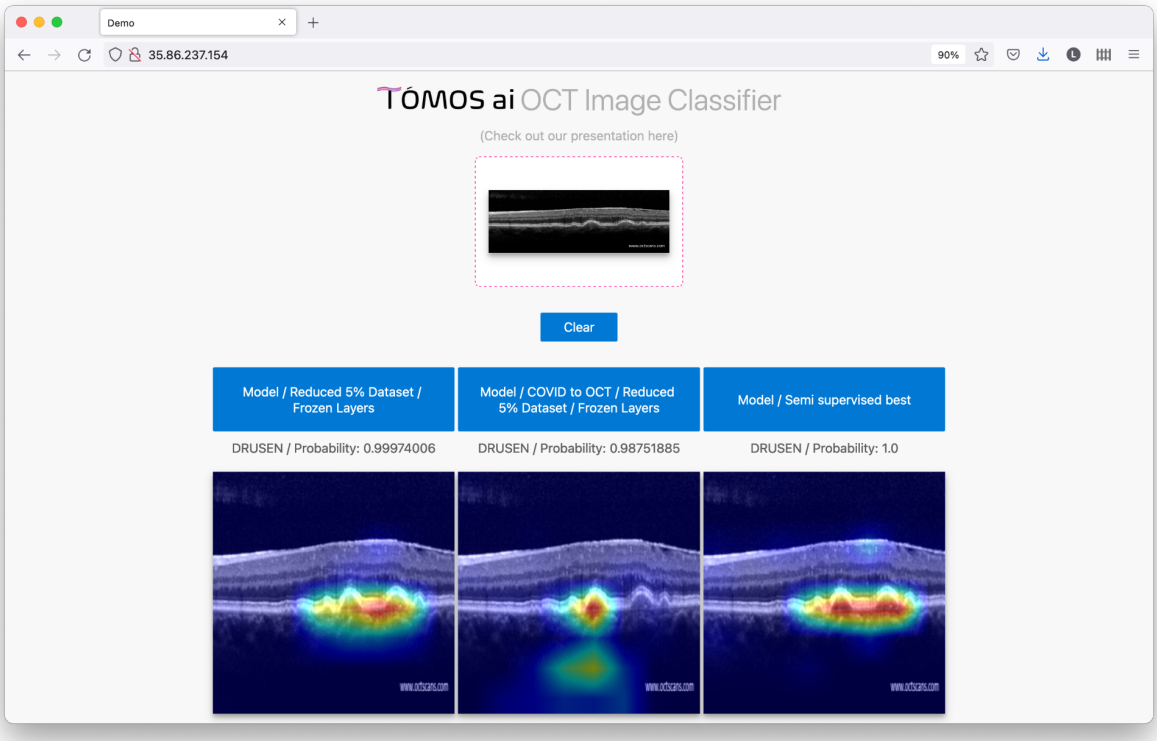

Figure 7: Deployment website depicting our three models predicting the upper image's condition.

Results

The semi-supervised model showed an improvement of 0.9% in F1 weighted average over the baseline model.

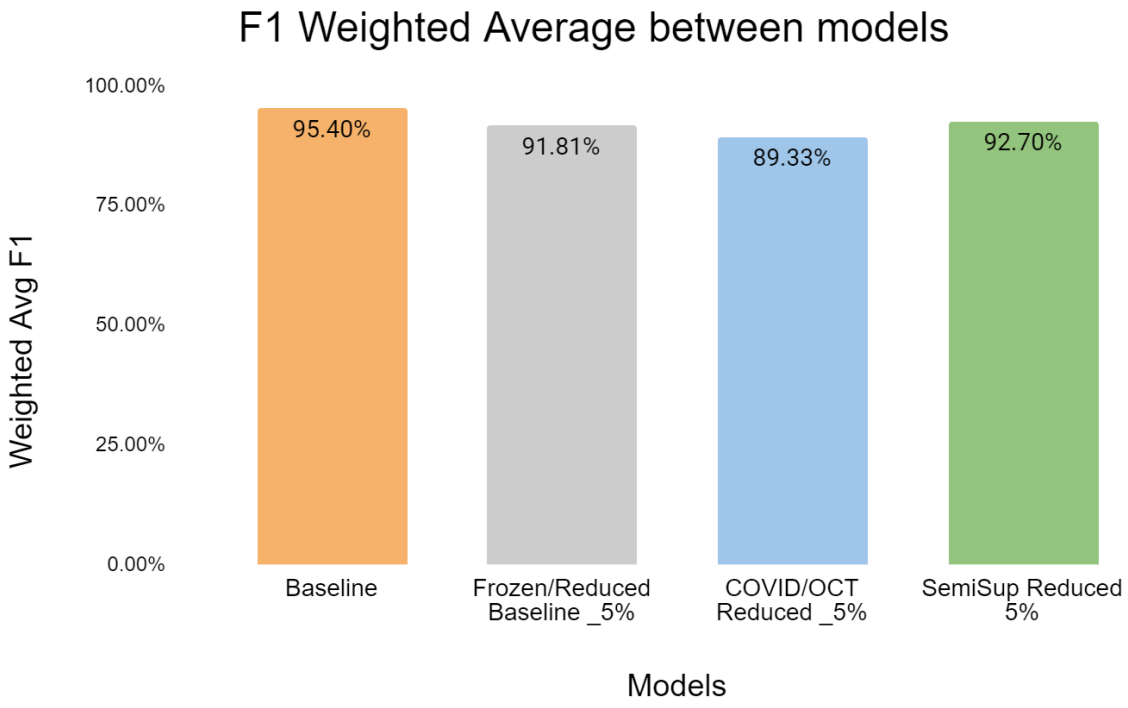

Figure 8: Bar graph of F1 weighted average between 5% train set trained models.

Conclusion & Future Directions

We improved performance on a limited (5%) labeled dataset with a semi-supervised learning approach (+0.9% F1).

To improve the model further, we would like to try performing co-training, freezing varying amounts of layers for transfer learning, and testing Vgg16. To better highlight performances between models, we could also further reduce the train set from 5% to 4% or 3%.

Overall, through the work completed in this project, we learned how to take a specific Dataset, run it through a Machine Learning pipeline specific to that data, and improve the performance catering the pipeline to the data.

References

1) Kermany D, Zhang K, Goldbaum M (2018). Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification. *Mendeley Data.* https://www.kaggle.com/paultimothymooney/kermany2018

2) Rahimzadeh M, Attar A, Sakhaei SM (2021). A fully automated deep learning-based network for detecting COVID-19 from a new and large CT scan dataset. *Biomedical Signal Processing and Control*. 68. https://www.kaggle.com/mohammadrahimzadeh/covidctset-a-large-covid19-ct-scans-dataset

3) Mallick S (2021). Transfer Learning for Medical Images. *LearnOpenCV.* https://learnopencv.com/transfer-learning-for-medical-images/