COVID-19 Detection with X-ray Image Classification Using Cloud Computing Platforms

https://youtu.be/N7E0nHKPrrc

Final Report

Team 27

Fathima Said Mohamed Thotathil | Anh L Tran | Ron He

Abstract

In this project, we demonstrate novel methods to classify X-ray images of patients using Deep Learning Pipelines (DLP) processing techniques such as PyTorch in Amazon Sagemaker and Pyspark in DataBricks. We observed an improvement in the accuracy and performance over the current state-of-the-art techniques, hereby establishing these methodologies as first-in-class neural network architectures for performing binary and multi-class classification on X-ray Images. Furthermore, we managed to host the best performed model in AWS and build a web application to allow users to upload an image and obtain a response message with covid-19 prediction based on the uploaded X-ray images.

Introduction and Background

Since the novel coronavirus (COVID-19) spreads rapidly around the world, there have been more than 50 million cases reported worldwide in which more than one million people have died[1]. Due to the highly contagious nature and the severe consequences of COVID-19, it is important to detect the active cases as early as possible in order to prevent the spread of the virus and to help healthcare professionals provide quick and effective treatments for infected individuals. Currently, reverse transcription polymerase chain reaction (RT-PCR) is the most adapted method for the diagnosis of COVID-19. However, this method is time consuming. Chest X-ray image analysis by healthcare professionals is another effective COVID-19 detection method.

To elaborate the X-ray image method for rapid COVID-19 detection, several artificial intelligence (AI) techniques that use deep learning models have been proposed and built. Among contemporary deep learning architectures, Convolutional Neural Network (CNN) is at the core of most of the state-of-the-art computer vision solutions for a wide variety of tasks and has yielded many substantial gains in various benchmarks. Various studies on Covid-19 X-ray classification use CNN concepts mainly to build models. For example, Narin et al. (2020) applied the transfer learning method by using five pre-trained CNN based models (ResNet50, ResNet101, ResNet152, InceptionV3 and Inception-ResNetV2) to classify images as COVID-19, normal (healthy), viral pneumonia and bacterial pneumonia [1,2,3]. These models were built from the ImageNet dataset, which has millions of images from multiple categories, and have numerous convolution layers and maximum pooling steps. They used these models on the chest X-ray images dataset created by Dr. Joseph Cohen et al [3, 4]. These models yielded high accuracy

[1] https://coronavirus.jhu.edu/us-map

(more than 95%) for binary classification (COVID vs. No-Findings) and multi-class classification (COVID vs. No-Findings vs. Pneumonia). Some studies also combined the CNN with other machine learning models such as Support Vector Machine and Decision tree for increasing the accuracy of COVID19 detection in classification tasks [7, 8].

Analyzing a large scale of X-ray images from patients requires lots of storage space and computation effort. To meet such high biomedical analysis demand, both Hadoop and Spark frameworks provide optimal and efficient architecture solutions. Spark architecture has been demonstrated as the most complete because it facilitates the implementation of algorithms with its embedded libraries [10]. The Spark framework functions by distributing the tasks onto multiple worker nodes. It has been shown that the proposed system achieves high-throughput and effective activity, such as image reconstruction [11]. One of the Experimental results indicate that the proposed PMIR algorithm fully inherits the parallelism of the MIR algorithm, resulting in the fast reconstruction of images. The proposed parallel algorithm performs an average of 28.56 times faster than sequential Python implementation [12].

Amazon Web Services provides comprehensive cloud computing services Specifically, Amazon SageMaker is efficient for building, training, and deploying the model via SageMaker hosting services [13].

Databricks is an alternative platform that also uses Amazon Web Services to train and deploy machine learning models. It is a web-based platform for working with Spark that provides automated cluster management and IPython-style notebooks.

In our project, we used both PyTorch and PySpark frameworks with two pretrained models (i.e. ResNet50 and InceptionV3) for binary classification (COVID vs. No-findings) and multi-class classification (COVID vs. No-findings vs. Pneumonia) settings.

Approach and Implementation

Image Dataset

During the model building phase, we initially used "X-Ray Image DataSet" from Ozturk et al. [6] study that is publicly available on Github. This data set contains 1,127 images in three classes: 127 "Covid-19", 500 'No_findings', and 500 "Pneumonia". During the model selection and tuning phase, in order to improve the model performance and generalize the models for big data, we combined other two X-Ray image datasets from Cohen JP et al. [4] and Chowdhury et al. [5] studies (see table A in the appendix for details about each dataset). The final data set consists of 4412 frontal chest X-ray images in three classes: 698 "Covid-19", 1851 'No_findings', and 1863 "Pneumonia".

For model training and testing purposes, all of the images with diagnostic labels were merged into a single data set, and then randomly split so that 80% of the data set were for training and the remaining 20% for testing.

Models

ResNet-50, short for Residual Networks, is a 50-layer-deep classic convolutional neural network. It has 5 stages each with a convolution block of 3convolution layers and an identity block of 3 convolution layers. Like other ResNet models, ResNet-50 uses skip connection to add the output from an earlier layer to a later layer to mitigate the vanishing gradient problem.

InceptionV3 is another powerful convolutional neural network that was developed from Googlenet, a deep learning model from Google. It mainly focuses on reducing computation and increasing optimization by applying different techniques such as factorized convolutions, regularization, dimension reduction, and parallelized computations. Thus, it has approximation of an optimal local sparse structure and processes visual or spatial information at various scales and then aggregate.

We used the ResNet-50 and InceptionV3 pretrained models. These two models have established pretrained weights and biases from the ImageNet classification training and have many well-built convolutional, RELU, and Dropout layers.

PyTorch and Amazon SageMaker

PyTorch natively supports ResNet50 and InceptionV3 pretrained models. In our first approach, we used PyTorch to build the classification models. We adapt the concept of transfer learning, in which the models use all but the last fully-connected layer as a fixed feature extractor and then train a classifier on top of it [9].

Because the data include images in various sizes and type formats, we resized all the images to the size of 224x224 for ResNet-50 and 299x299 for InceptionV3, normalized them with 0.485 mean and 0.229 standard deviation, and converted them into multidimensional tensors using torchvision.transform package. Then we shuffled up the dataset images and divided them into batches of 20 using the DataLoader function to feed the PyTorch-based machine learning model.

In addition to the aforementioned two pretrained models, for the PyTorch framework, we used a cross-entropy loss function to measure the deviation of the predictions of each class. We also used Stochastic Gradient Descent (SGD) as the optimizer with a learning rate of 0.001 that updates the values of the parameters of each layer and generates probabilities for the target class labels via its SoftMax function behind the scene. However, the results were not great. We then changed the optimization technique to Adaptive Moment Estimation (Adam) method with a learning rate of 3e-5. We applied the train, evaluation, and visualization functions from homework 5 on our dataset and models.

We used Amazon SageMaker's GPU powered integrated Jupyter authoring notebook instance to build and train both ResNet-50 and InceptionV3 models using custom PyTorch code . Since the ResNet-50 model performed better than the InceptionV3 model, we chose to host the ResNet-50 model using SageMaker hosting services to provide inferences. We used Amazon SageMaker Neo to compile and deploy the trained model on a GPU powered real time inference

instance. We used Amazon Simple Storage Service (S3) to store the dataset to train the model, the best trained model and the SageMaker Neo compiled model.

In order to allow the hosted model to be accessible from external applications, we integrated the hosted model with an Internet facing REST API built using Amazon API Gateway and Amazon Lambda function written in Python. API Gateway acts as the 'front door' to our serverless frontend web application built in React using AWS Amplify. The front end client application is a single page web application which provides users the ability to upload their Chest X-ray image. After a user uploads the image, the request is sent to the POST API hosted on API Gateway which then triggers the Lambda function. Then the Lambda function in turn forwards the request to the hosted model. The hosted model then provides the inference back to the Lambda function which is then forwarded to the API Gateway and finally to the client application (See Figure 1 in the Appendix for the AWS infrastructure diagram and Figure 2 for the Client Application wireframes).

PySpark and DataBricks

PySpark MLlib also natively supports ResNet50 and InceptionV3 pretrained models. Besides providing the same transfer learning experience, the Spark framework offers a memory-based platform giving the Big Data ecosystem a major boost. To host the PySpark notebook, we set up the DataBricks Cluster as following:

- Python 3 cluster running Databricks Runtime 5.5 LTS ML.
- A spark-deep-learning library with the Maven and Coordinate 1.4.0-spark2.4-s_2.11.
- Libraries with the Source option PyPI and Package tensorflow==1.12.0, keras==2.2.4, h5py==2.7.0, wrapt.

To train the model in PySpark, we normalized all the images to the size of 299x299 and then loaded all pre-processed images into Spark with a target column. After this we assigned a label to each of the images based on the actual diagnosis outcome of these images.

In addition to the aforementioned two pretrained models (ResNet50 and InceptionV3), we removed the last predicting layer of the pre-trained model and replaced them with our own predicting layers. Only Weights of the predicting layers are updated during the training. This transfer learning approach allows the training process to converge quickly.

Experimental Evaluation

Metrics

To evaluate our approaches and models, we calculated the precision, recall, F1-score, and accuracy. We also plotted confusion metrics.

PyTorch

For both ResNet50 and InceptionV3 models, we used 10 epochs to train the images and ran each model against the test datasets. Below is the summary of the results.

Table 1--Binary Classification Results

Model	Precision (Covid Image only)	Recall (Covid Image only)	F1-score (Covid Image only)	Overall Accuracy
ResNet-50 - sample dataset (SGD; 0.001 learning rate)	0.95	0.95	0.95	0.933
InceptionV3 - sample dataset (SGD; 0.001 learning rate)	0.93	0.91	0.92	0.912
ResNet-50 - full dataset (Adam; 3e-5 learning rate)	0.99	0.99	0.99	0.993
InceptionV3 - full dataset (Adam; 3e-5 learning rate)	0.99	0.99	0.99	0.992

Table 2-- Multiclass Classification Results

Model	Precision (Covid Image only)	Recall (Covid Image only)	F1-score (Covid Image only)	Overall Accuracy
ResNet-50 - sample dataset (SGD; 0.001 learning rate)	0.95	0.80	0.87	0.871
InceptionV3 - sample dataset (SGD; 0.001 learning rate)	0.88	0.84	0.86	0.833
ResNet-50 - full dataset (Adam; 3e-5 learning rate)	0.95	0.91	0.93	0.950
InceptionV3 - full dataset (Adam; 3e-5 learning rate)	0.85	0.91	0.88	0.910

As we can see from the above results, the Adam method with 3e-5 learning rate optimizer outperformed the SGD with 0.001 learning rate for both models, even when running on the full dataset. For the binary classification task, the models achieved a very high accuracy about 99%. For the multiclass classification task, the ResNet-50 model (95% accuracy) performed better than the InceptionV3 model (91% accuracy). Noticeably, in the multiclass confusion matrices

(see figures 3 and 4 in the appendix), the true positive rate for covid is 0.91 for both best models while the other classes have higher rates, except the normal class in InceptionV3.

PySpark

In PySpark training, weights of the predicting layers are updated during the training until the prediction outcomes converge. The performance of the models are summarized in Table 3 and Table 4.

Table 3--Binary Classification Results

Model	Precision (Covid Image only)	Recall (Covid Image only)	F1-score (Covid Image only)	Overall Accuracy
Resnet50 and Logistic Regression	1.00	1.00	1.00	1.000
Resnet50 and Random Forest	0.96	1.00	0.98	0.979
Resnet50 and Decision Tree	0.78	0.88	0.82	0.809
InceptionV3 and Logistic Regression	1.00	0.88	0.93	0.936
InceptionV3 and Random Forest	0.77	0.71	0.74	0.745
InceptionV3 and Decision Tree	0.67	0.58	0.62	0.638

Table 4-- Multiclass Classification Results

Model	Precision (Covid Image only)	Recall (Covid Image only)	F1-score (Covid Image only)	Overall Accuracy
Resnet50 and Logistic Regression	1.00	0.88	0.94	0.851
Resnet50 and Random Forest	0.73	0.65	0.69	0.716
Resnet50 and Decision Tree	0.56	0.59	0.57	0.627
InceptionV3 and Logistic Regression	0.94	0.88	0.91	0.731
InceptionV3 and Random Forest	0.57	0.71	0.63	0.582
InceptionV3 and Decision Tree	0.44	0.47	0.46	0.433

From the results from PySpark machine learning platforms, the Resnet50 with Logistic Regression last layer performs with highest accuracy (see figures 5 and 6 in the appendix for confusion matrices). From both Binary and Multiclass Classification results, this model performs above 0.85 accuracy. It is very likely that the Resent50 pre-trained model captures the images and converts them into two not well-separated boundaries. Under such output, trees are susceptible to overfitting the training data, so that logistic regression's simple linear boundary generalizes better.

The above results all indicate that Resnet50 is better performed than InceptionV3 to predict the patients' outcome. It is very likely due to the fact that Resnet50 uses skip connection to propagate information over layers and enhances the detection of smaller objects in the image, while the InceptionV3 model suffers from the vanishing gradients problem.

Discussion

Based on the results, we observed that both the approaches yielded high accuracy when using the ResNet50 pretrained model compared to the InceptionV3. In addition, the accuracy of binary classification was higher than that of multiclass classification.

The PyTorch approach used the transfer learning method with pretrained models. By applying the cross-entropy loss function and Adam optimizer with a learning rate of 3e-5, we were able to achieve good results with above 90% accuracy for both classification tasks. Specifically, the binary classification yielded about 99% accuracy and multiclass classification yielded 95% accuracy along with other high-score metrics such as precisions, recalls, and F1-score.

The PySpark approach extracted the features from the pretrained models and then trained the Logistic Regression parameters until the outcome converges. It also results in much faster training with a high accuracy.

Similar to previous studies, the multiclass classification task is challenging to yield strong results compared to the binary classification [5-7]. Specifically, with the addition of the viral (non-covid pneumonia) class, the performance of the models drops significantly. Another challenging part was adapting to the AWS cloud computing environment; our team needs to learn all the cloud storage and hosting methods and counters lots of bugs when implementing our machine learning methods. Because not all the environments were able to run our codes developed in other platforms, we had to stick with Databricks to run PySpark codes and extract models from there. The next challenge is to select the machine learning architecture. We spent a significant amount of time training the models, looked at the mis-classified data to tune and choose the best model. Our next step is to further fine-tune the training parameters using cross validation. We will be expecting higher prediction accuracy from our trained models.

Conclusion

In this project, we demonstrated two novel methods (i.e. PyTorch and PySpark) of classifying X-ray images of patients using Deep Learning Pipelines (DLP) processing techniques

utilizing two different cloud computing architectures: Amazon SageMaker and Databricks. Both approaches produced high accuracy with data pipeline efficiency and reduced runtime.
The current results demonstrate that these innovative methodologies could be used to obtain accurate predictions for binary and multi-class classification on X-ray Images. Using the best performed model as a foundation, we successfully built a web service that allows users to upload X-ray images and response with highly accurate prediction outcomes.

References

[1] Ilyas, M., Rehman, H., & Naït-Ali, A. (2020). Detection of Covid-19 From Chest X-ray Images Using Artificial Intelligence: An Early Review. *arXiv preprint arXiv:2004.05436*.

[2] Shi, F., Wang, J., Shi, J., Wu, Z., Wang, Q., Tang, Z., ... & Shen, D. (2020). Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19. *IEEE reviews in biomedical engineering*.

[3] Narin, A., Kaya, C., Pamuk, Z. (2020). Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks. arXiv preprint arXiv:2003.10849.

[4] Cohen, J. P., Morrison, P., & Dao, L. (2020). COVID-19 image data collection. *arXiv preprint arXiv:2003.11597*.

[5] Chowdhury, M. E., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., ... & Reaz, M. B. I. (2020). Can AI help in screening viral and COVID-19 pneumonia?. *arXiv preprint arXiv:2003.13145*.

[6] Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O., & Acharya, U. R. (2020). Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Computers in Biology and Medicine*, 103792.

[7] Sethy, P. K., Behera, S. K., Ratha, P. K., & Biswas, P. Detection of coronavirus Disease (COVID-19) based on Deep Features and Support Vector Machine.

[8] Yoo, S. H., Geng, H., Chiu, T. L., Yu, S. K., Cho, D. C., Heo, J., ... & Min, B. J. (2020). Deep learning-based decision-tree classifier for COVID-19 diagnosis from chest X-ray imaging. *Frontiers in medicine*, *7*, 427.

[9] Minaee, S., Kafieh, R., Sonka, M., Yazdani, S., & Soufi, G. J. (2020). Deep-covid: Predicting covid-19 from chest x-ray images using deep transfer learning. *arXiv preprint arXiv:2004.09363*.

[10] Kouanou, A. T., Tchiotsop, D., Kengne, R., Zephirin, D. T., Armele, N. M. A., & Tchinda, R. (2018). An optimal big data workflow for biomedical image analysis. *Informatics in Medicine Unlocked*, *11*, 68-74.

[11] Mishra, Sanjukta (2018). A REVIEW ON BIG DATA ANALYTICS IN MEDICAL IMAGING, International Journal of Computer Engineering and Applications, Volume XII, Issue I, Jan. 2018.

[12] Ullah, R., & Arslan, T. (2020). PySpark-Based Optimization of Microwave Image Reconstruction Algorithm for Head Imaging Big Data on High-Performance Computing and Google Cloud Platform. *Applied Sciences*, *10*(10), 3382.

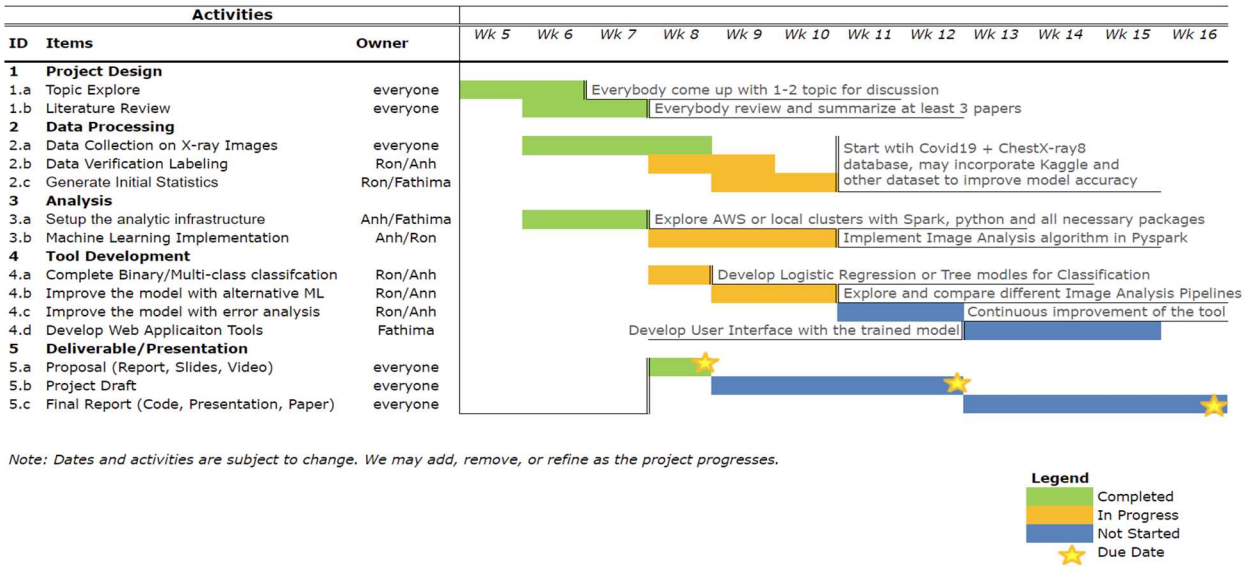
[13] Mishra, Abhishek (2019). Machine learning in the AWS Cloud: add intelligence to applications with Amazon SageMaker and Amazon Rekognition, Sep. 2019.

[14] Lombog, M. (2020, May 21). Build a React App Using AWS Amplify in Simple Steps (Part 1). Retrieved November 15, 2020, from https://medium.com/better-programming/build-a-react-app-with-authentication-using-aws-amplify-49db1dfdc290

[15] Amazon. (2020.). Deploying pre-trained PyTorch vision models with Amazon SageMaker Neo. Retrieved from https://github.com/aws/amazon-sagemaker-examples/blob/master/sagemaker_neo_compilation_jobs/pytorch_torchvision/pytorch_torchvision_neo.ipynb

Team Contributions

All the team members have contributed equally throughout the project life cycle. Below is the timeline of our team's activities.



Note: Dates and activities are subject to change. We may add, remove, or refine as the project progresses.

Appendix

Table A-- COVID-19 datasets

Index	Data Source - Author	Amount	Link
1	COVID Chest x-ray dataset (multiple sources) - Cohen JP et al. [4]	866 X-ray images total but only 782 are frontal view. Among 782 frontal view images, 18 No findings, 191 Pneumonia-No covid, and 478 Pneumonia-Covid	https://github.com/ieee8023/covid-chestxray-dataset
2	COVID 19 dataset (Covid19 developed by Cohen + ChestX-ray8 database provided by Wang) – Ozturk et al. [6]	127 Covid19 (these images are also in the #1 dataset), 500 no finding, 500 pneumonia	https://github.com/muhammedtalo/COVID-19
3	COVID 19 Kaggle dataset (images from Italian Society of Medical and Interventional Radiology (SIRM) COVID-19 DATABASE, Novel Corona Virus 2019 Dataset developed by Joseph Paul Cohen) - Chowdhury et al. [5]	219 Covid19 (include 158 images from #1), 1341 Normal, 1345 Viral Pneumonia	https://github.com/tawsifur/COVID-19-Chest-X-ray-Detection

Figure 1 -- AWS Target Architecture

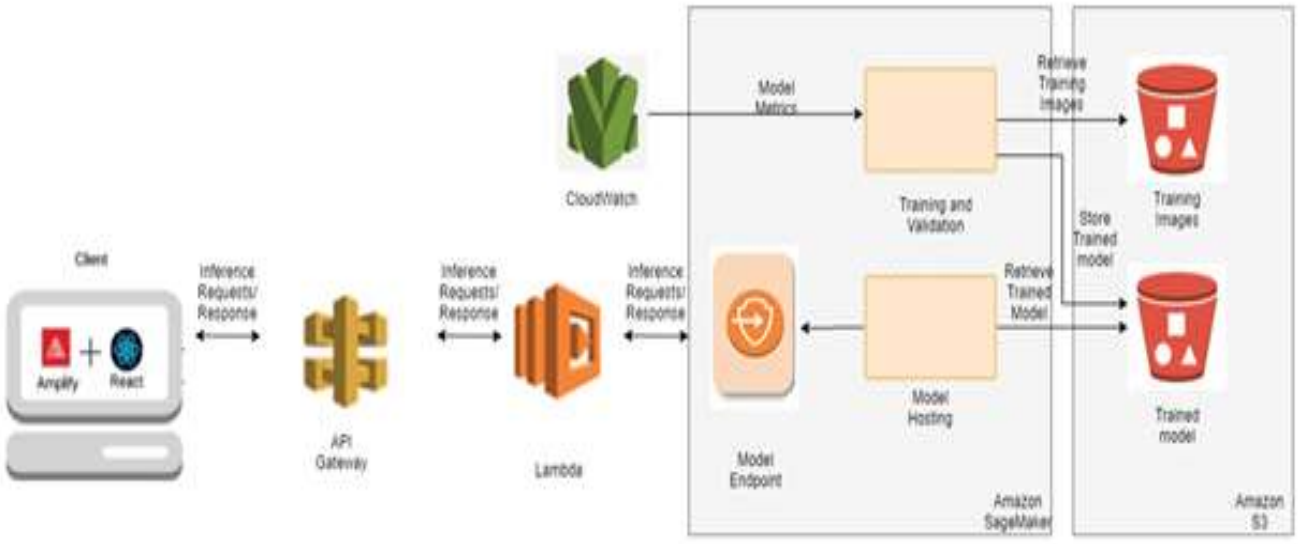
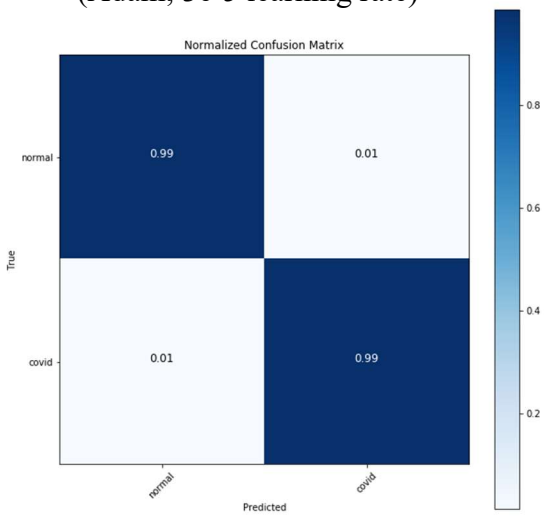


Figure 2 -- Client Application

Figure 3 -- Binary Classification Confusion Matrices for Testing Data using PyTorch - best models

ResNet-50 - full dataset
(Adam; 3e-5 learning rate)



InceptionV3 - full dataset
(Adam; 3e-5 learning rate)

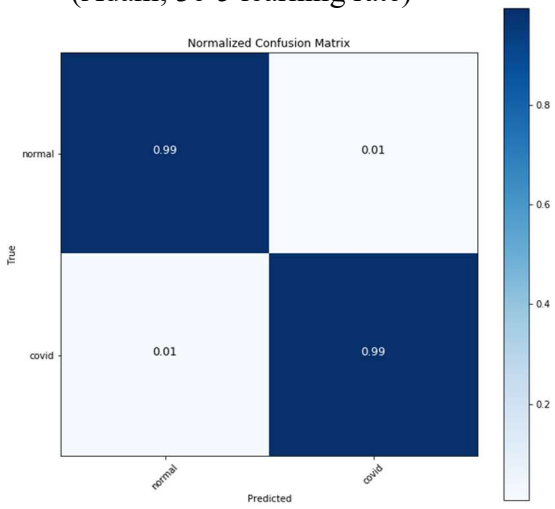
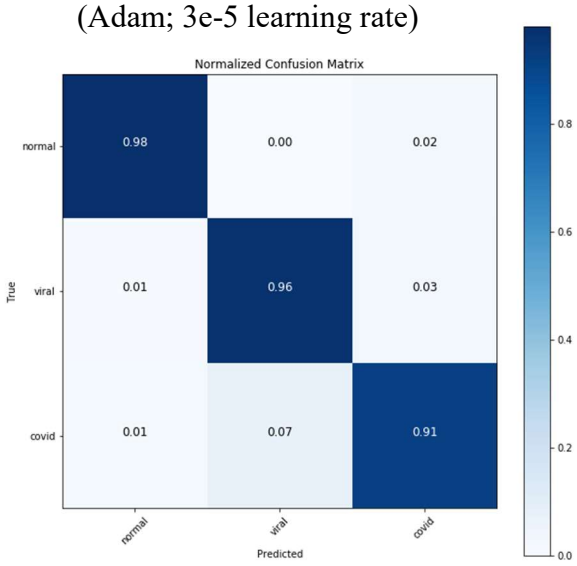


Figure 4 -- Multiclass Classification Confusion Matrices for Testing Data using PyTorch - best models

ResNet-50 - full dataset
(Adam; 3e-5 learning rate)



InceptionV3 - full dataset
(Adam; 3e-5 learning rate)

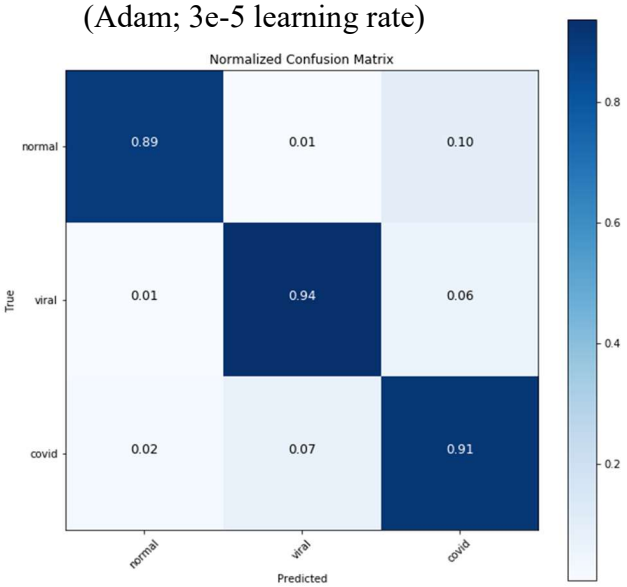
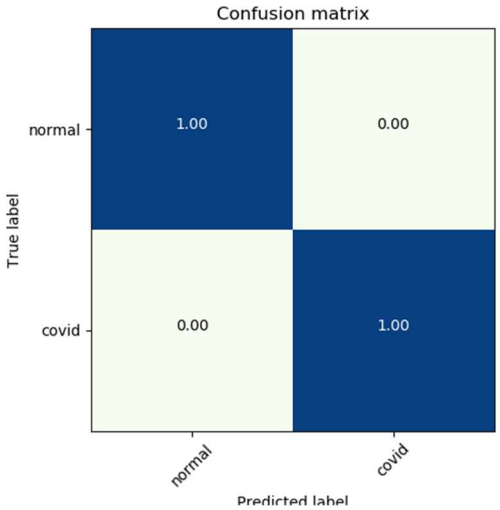


Figure 5 -- Binary Classification Confusion Matrices for Testing Data using PySpark

Resnet50 and Logistic Regression



InceptionV3 and Logistic Regression

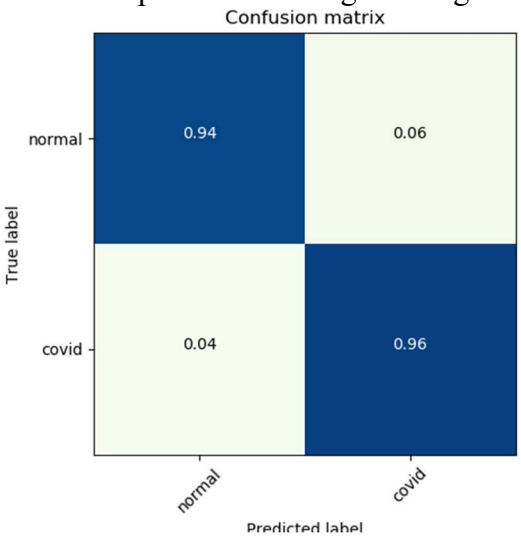
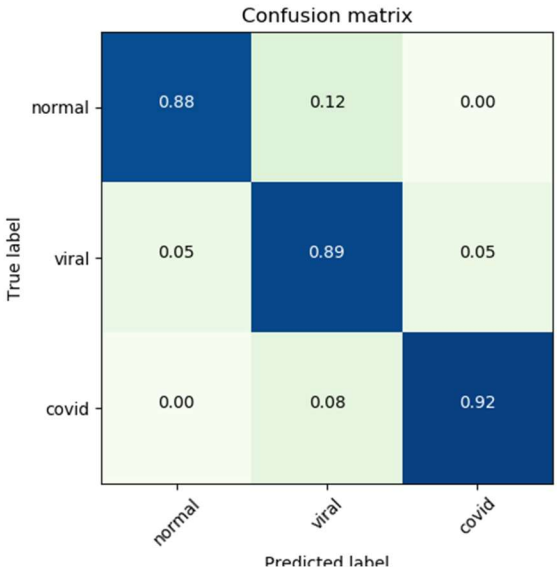


Figure 6 -- Multiclass Classification Confusion Matrices for Testing Data using PySpark

Resnet50 and Logistic Regression



InceptionV3 and Logistic Regression