COVID-19 Diagnosis based on CT Scan Images

Het Jagani
015261415
hethareshkumar.jagani@sjsu.edu

Akash Rupapara
015266511
akashjaysukhbhai.rupapara@sjsu.edu

Keya Patel
015280876
keyagirishkumar.patel@sjsu.edu

***Abstract*—Coronavirus 2019 (COVID-19) has spread all over the world and caused deaths in thousands of number. One of the important reason this spread is less accurate and late diagnosis of the disease. Also most of the diagnosis methods can have good chance of giving false negative test results. There has been great efforts in developing computational methods which can analyse the lung's image and diagnose the disease. Due to privacy reasons often these works are difficult to reproduce because the CT scan data is not available publicly. Besides this large amount of data is required for training models to predict COVID-19 accurately. We built the dataset by combining multiple publicly available datasets which resulted into hundreds of lungs CT Scan images. We applied various data pre-processing and data mining techniques on these datasets and trained various classification models. Specifically we used RESNET-50 pre-trained model for feature extraction from images. Then we trained SVM, KNN and DNN on the extracted features by applying PCA and without PCA. Our approach achieves highest accuracy of 89% and highest F1 score of 87.7% in diagnosing COVID-19 from CT scans even though the dataset is limited.**

***Index Terms*—COVID-19, CT, diagnosis, Data Preprocessing, Data Mining, Support Vector Machines, K-Nearest Neighbours, DNN, Resnet-50, Transfer Learning**

I. Introduction

Coronavirus 2019 (COVID-19) is a highly contagious disease and has infected around 190 million people and caused around 4 million deaths worldwide as of July 2021. One of the major factor contributing in spreading of disease is the inefficiency and shortage of tests. Current method of testing is reverse transcription polymerise chain reaction (RT-PCR) test. The RT-PCR test takes about 4-5 days to obtain results and there is considerable change to receive a false negative test result. Besides this factor RT-PCR test kits were in great shortage in earlier stages. This kind of inefficiency causes the rapid and unstoppable spread of disease world wide.

To lessen the effects of these inefficiencies great amount of resources and research are devoted towards finding robust and reliable test methodology. Studies have indicated that computed tomography (CT) scans provides clear radiological findings in COVID-19 patients and it can serve as efficient method for tests due to reasonable availability of CT scan machines. The images taken from CT machine is analysed by medical professional and he/she can diagnose COVID-19 disease in patient. Further efforts are being made to automate this process of analysis to alleviate the burden on medical professionals of reading CT Scans. This is possible due to advancements in data mining and AI fields. While these works have shown good results, still there are certain limitations to these works. The CT scan datasets used in such works are not available publicly. Due to privacy reasons such dataset is not made public for research and development of advanced AI models. Also, such a work requires large amount of datasets for accurate prediction.

Identify applicable funding agency here. If none, delete this.

In this work, we have addressed the above discussed problems by combining two publicly available datasets from [1] and [2]. By combining these dataset we have 1348 COVID and 1626 NON-COVID images. Still the number of images for such a task is not sufficient so we have applied some data augmentation techniques to augment the dataset and as a result we have total of 9434 COVID images and 11382 NON-COVID images. Still for such image processing task this amount is reasonable but not great because such data mining tasks are data hungry. To address the data deficiency we have used pretrained RESNET-50 model which can help retrieve more relavant and useful features from images. Here we have done Principal Component Analysis on those extracted features and trained various classification models for diagnosis task. Also we have used raw feature data obtained as output of feature extractor directly for training of classification models as part of experimentation.

II. Literature Review

Since the starting of outbreak of Covid-19, there have been lot of research and efforts made to develop accurate deep learning models to perform diagnosis of Covid-19 based on medical images of CT Scans and Chest X-Ray Images. An Early-screening model containing multiple CNN models to classify CT scans of patients with and without COVID-19 by Wu et al. [3]. By modifying existing inception transfer learning model, Xu et al. developed deep learning algorithm to provide clinical diagnosis replacing pathogenic tests [4]. When experimented to find sensitivity of detecting covid-19 from chest CT was found to be 97% by Ai et al [5].

Currenly, due to privacy issues large datasets of CT-scans are not available to develop model by training on images [6]. Hence, to deal with lack of image sources, we need to combine images from different sources to form unified dataset to get more accurate results. Collectively, mixing dataset will produce largest dataset available to train model with best of our knowledge [7]. An infection-size-aware model based

on random forest known as (iSARF) which is desigened to automatically categorizes medical images into groups with different ranges of infected lesian sizes [8].

Transfer learning is done by taking standard pre-trained neural networks as base along with their weights such as resnet, and begin training on that using current dataset. Transfer learning has given productive results and are widely used in medical domain for recognition and classification tasks such as tumor classification [9], detection of pneumonia [10], retina disease classification [11] and many more.

III. Methods

Fig 1 shows the workflow of the analysis done on the dataset.

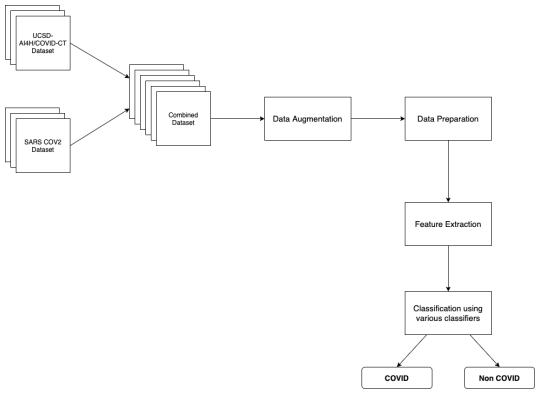

Fig. 1. Block Diagram

A. Data Preprocessing

As the number a result of less number of images in any single dataset, we have decided to combine two datasets and create a single dataset for training. But still the number of images in resultant dataset are less for such a data intensive task. So to address this problem we have used data augmentation techniques to increase the number of images in dataset to a reasonable number. Here we have applied six transformations on each image so that the number of images becomes seven times of the original. Following are the transformations applied on each image:

- Flip Left Right
- Flip Up Down
- Rotate 20^o right
- Rotate 20^o left
- Adding Gaussian random noise
- Adding Salt and Pepper noise

Before applying all the transformations images are resized to 256x256 to keep consistent image size. For all data augmentation tasks tensorflow.image library is used. So after augmentation we have total 9434 COVID images and 11382 NON-COVID images for further data mining purpose.

B. Data Preparation

After data preprocessing we have 20614 images of size 256x256 in total. Loading this dataset for training is very memory intensive task and also we have observed that we do not need such large size images for training purpose. The CT images are usually non colored images so there is also no need to preserve all three channels (i.e. RGB) of images. So the data preparation step comprises of loading the whole image dataset, preparing input samples and output labels, split into train and test datasets, converting images to gray scale, resizing the image and save those arrays such that they can be loaded easily.

For loading dataset we have used keras' image_dataset_from_directory function which expects the dir structure of /label/images. We have kept 80% images in training set and 20% in test set. After that we convert all images to gray scale and resize each image to 64x64. Fig 2 shows some of the images with labels.

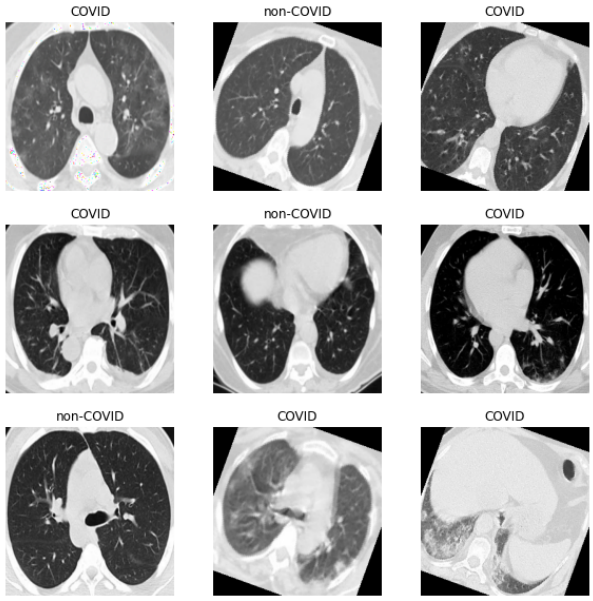

Fig. 2. Sample Images

After all the processing we save the arrays generated to .npy files which can be easily loaded for further training purpose.

C. Experiments

As this is image classification task, we have used CNN for extracting higher level features from images. But here we have not trained a CNN from scratch because we have limited amount of images. Training CNN would require much more data to gain reasonable performance. So we chose to do use a pre-trained model as a feature extractor and do further processing on those extracted features. Here pretrained RESNET-50 from keras library with imagenet weights is used for feature extraction task.

As part of experimentation we have done PCA on image training dataset and regenerated images by preserving 70% principal components and then used it for further training purpose. Fig 3 shows the images after applying PCA on them.

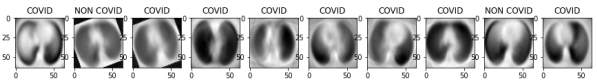

Fig. 3. Sample Images after applying PCA

For classification model we have flattened output of RESNET-50 model and attached a Dense layer with 512 unit and relu activation, two Dense layers with 256 units and relu activation and a Dense layer with 1 unit and sigmoid activation as output. In this model we froze the layers of RESNET-50 so that it was only used as feature extractor. This model had 4,392,193 trainable parameters for last 3 attached layers. Also the same model was also trained without applying PCA on initial dataset. For both experiments RMS Prop with learning rate of 0.0001 as optimizer and Binary Cross Entropy as loss function is used. It was found that PCA causes loss in information in images so model without PCA performed better.

Training the DNN model is quiet time consuming and is compute intensive task. So, we have tried using K-Nearest Neighbours (KNN) algorithm for this classification task. KNN is pretty simple and intuitive algorithm for classification task. It takes very low time for prediction and it does not require training it just tags the new test data based on previous training examples. In this experimentation we have extracted feature from the feature extractor CNN model and used these features for fitting KNN classifier. One of the cons of KNN classifier is Curse of Dimensionality and so we have applied PCA on extracted features and retained 80% components so that dimensions of features reduced from 8192 to 86 which is good for KNN classifier. Fitting and evaluating test dataset took about 9 seconds without PCA and about 1.6 seconds with PCA. This values are quiet low as compared to training other classifiers. We have used KNN with K = 5 and classification was done based on distance as weights when making decision about class.

The downside of KNN algorithm is that it is not robust and does not generalize well when we want to use the trained model to classify large number of test examples. Also KNN does not perform well in imbalanced datasets. Many studies have shown that Support Vector Machines perform well for medical diagnosis classification tasks. So as part of third experimentation we have trained a SVM classification model. SVM algorithm takes too long time to run because it projects data in higher dimensions to fit a large margin classifier. But by applying PCA on features the dimension reduces so training SVM on those features improves the time of training to great extend. In this case already the dimensions of our feature vectors is too high so SVM does not improve much on accuracy measure then previous methods. For SVM we have trained two models, one with PCA and one without doing PCA on features. We have chosen Radial Basis Function (RBF) kernel for training SVM classifier.

IV. Results

Table I shows all the scores of different models which are trained on the dataset. Here all the scores are calculated on testing dataset. It can be seen that accuracy of DNN method yeilds best results and it takes around 7.77 minutes to train. Also KNN with PCA algorithm is fastest and gives accuracy of 87.24% and F1 Score of 85.64%. Applying PCA on training dataset and transforming train and test datasets to lower dimension takes about 144.6 seconds.

TABLE I
Performance scores of all methods

	Accuracy	F1 Score	Precision	Recall
DNN	90.2	88.72	90.17	87.32
KNN	89.02	87.72	86.67	88.79
KNN with PCA	87.24	85.64	85.12	86.18
SVM	87.05	84.9	87.52	82.42
SVM with PCA	86.26	83.98	86.55	81.55

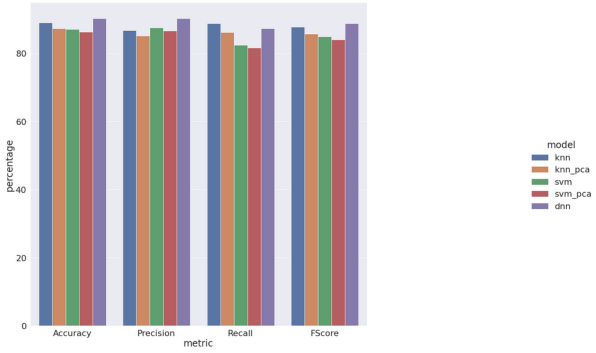

Fig. 4. Bar chart of Performance scores of different models

Fig 4 shows that DNN model gives best F1 Score which is 88.72% and also it gives best accuracy which is 90.2%. Here the dataset we have is not so skewed number of Covid and Non Covid samples does not differ too much so accuracy can be considered as valid measure for evaluating models. While training DNN we kept 20% of our training data as validation dataset so that we can know if the models are overfitting on training data. Fig 5 shows the plot of loss while training for 10 epochs. Here we can observe that validation loss is also decreasing with each training step so the model is not overfitting.

Table II shows the time taken by each model to train. Fig 6 shows the confusion matrices for all the methods that are implemented for classification task. Here we can observe that number of false positives and false negatives are relatively low as compared to true positives and true negatives. which is a good thing for classification model.

TABLE II
TIME TAKEN TO RUN ALGORITHMS

Model	Time in seconds
PCA	144.64
DNN	466.14
KNN	9.03
KNN with PCA	1.59
SVM	526.2
SVM with PCA	12.59

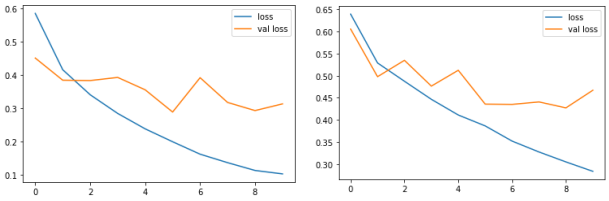

Fig. 5. Left: loss plot of DNN, Right: loss plot of DNN with PCA on features

V. CONCLUSION

In this project we have achieved maximum of 90.2% accuracy on the COVID-19 diagnosis task which is reasonable but not good enough to use in real world. Here we have achieved good performance then [1] which is the base paper we used for datset. We are able to achieve this result because of combination of two datasets and augmenting the dataset which increased the number of samples to significant amount. It is observed that such image classification task is very data hungry and Deep Neural Network models perform better if we have significant amount of data to train.

REFERENCES

[1] X. He, X. Yang, S. Zhang, J. Zhao, Y. Zhang, E. Xing, and P. Xie, "Sample-efficient deep learning for covid-19 diagnosis based on ct scans," *medRxiv*, 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/04/17/2020.04.13.20063941

[2] E. Soares, P. Angelov, S. Biaso, M. H. Froes, and D. K. Abe, "Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification," *medRxiv*, 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/05/14/2020.04.24.20078584

[3] X. Xu, X. Jiang, C. Ma, P. Du, X. Li, S. Lv, L. Yu, Q. Ni, Y. Chen, J. Su *et al.*, "A deep learning system to screen novel coronavirus disease 2019 pneumonia," *Engineering*, vol. 6, no. 10, pp. 1122–1129, 2020.

[4] S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao, J. Guo, M. Cai, J. Yang, Y. Li, X. Meng *et al.*, "A deep learning algorithm using ct images to screen for corona virus disease (covid-19)," *European radiology*, pp. 1–9, 2021.

[5] C. Zheng, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, W. Liu, and X. Wang, "Deep learning-based detection for covid-19 from chest ct using weak label," *MedRxiv*, 2020.

[6] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi, "Covid-19 image data collection: Prospective predictions are the future," *arXiv preprint arXiv:2006.11988*, 2020.

[7] X. He, X. Yang, S. Zhang, J. Zhao, Y. Zhang, E. Xing, and P. Xie, "Sample-efficient deep learning for covid-19 diagnosis based on ct scans," *medrxiv*, 2020.

[8] F. Shi, L. Xia, F. Shan *et al.*, "Large-scale screening of covid-19 from community acquired pneumonia using infection size-aware classification. arxiv e-prints [preprint] 2020."

[9] B. Q. Huynh, H. Li, and M. L. Giger, "Digital mammographic tumor classification using transfer learning from deep convolutional neural networks," *Journal of Medical Imaging*, vol. 3, no. 3, p. 034501, 2016.

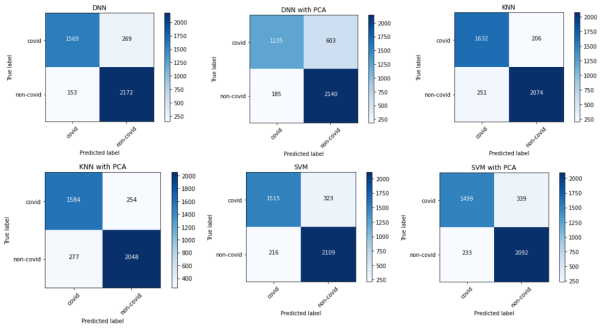

Fig. 6. Confusion Matrices

[10] V. Chouhan, S. K. Singh, A. Khamparia, D. Gupta, P. Tiwari, C. Moreira, R. Damaševičius, and V. H. C. De Albuquerque, "A novel transfer learning based approach for pneumonia detection in chest x-ray images," *Applied Sciences*, vol. 10, no. 2, p. 559, 2020.

[11] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging," *arXiv preprint arXiv:1902.07208*, 2019.