COVID-19 Detection In CT Scans

Michael Baskhairoun
McMaster University
1280 Main Street, Hamilton. Ontario
baskhaim@mcmaster.ca

Abstract

In this project, a program was written in python language to detect COVID-19 in CT Scans of the lungs using prepossessing, feature extraction and machine learning recognition methods. In this project, two methods of recognition were used, linear regression which achieved a miss classification rate of 24.15% (Accuracy = 75.85%), and neural networks which achieved a miss classification rate of 23.51%, the results, implementations, and differences of these two methods will be further discussed in the report.

1. Introduction

In 2020, the world faced a pandemic that later became known by everyone as the COVID pandemic. The pandemic had a drastic effect on everyone's lives in all countries of the world. Since the beginning of the pandemic, testing for the virus became crucial, as people needed to know weather, they have the virus or not and if there is a need for self isolation.

As the pandemic strengthened and the infections increased, testing methods like PCR and rapid testing became quickly available. The scarcity of tests led to the need of other readily available traditional methods like CT scans of the lungs. A CT scan shows cut sections of the lungs in which the lung's alveoli can be seen. A very common COVID symptom is pneumonia [1] which is the inflammation of the alveoli in the lungs. By looking at these CT scans, a professional physician or radiologist can compare the numbers and sizes of the inflamed alveoli and identify if the patient has COVID or not. [4]

Given the huge number of daily cases of COVID infections, a solution that will aid in the analysis and detection of COVID in CT scans would be very helpful if not crucial. This paper discusses an image processing program that uses machine learning (linear regression, and neural networks) to detect if a CT scan has Covid based on lung CT features that would indicate if a patient has covid symptoms (pneumonia). [3]

2. Related Works

Since identifying COVID using CT scans proved to be reliable in most cases, a lot of hospitals adopted software and programs that would be able to identify COVID by taking and analysing CT scans of patients. However, most of these programs are not available publicly and are privately owned. However, some papers about proposed methods to tackle the problem are available. Since, the problem is very new, there are very few implementations that are publicly available. [2]

One implementation that's available online, uses convolutional neural networks to compare images as a whole without extracting features. This method achieved very high accuracy results as high as 90% accuracy on the data sets used. However, this method proved to be unreliable since its only able to detect COVID from the same dataset but not from other datasets or other CT scans. This is due to the variances between different CT machines which can produce a lot of invariances and since the program is only trained on datasets from a very specific dataset, it won't be able to detect COVID in external scans. [2]

Another implementation uses datasets of 3D CT scans and random forests for classification. The implementation uses features of alveoli inflammation size. This implementation was able to achieve an accuracy of 87%. The dataset used for this implementation is very rich and uses original unprocessed 3D CT scans which have a color range of -1000 to 1000 which makes feature extraction much more efficient and reliable than regular images since the features to be extracted or masked would have a much more specific color range. However, such datasets were not available with the implementation, and they require specific programs to analyze and can not ne done through open cv and therefore they

could not be used in this project. [5]

3. Proposed Method

There are two methods that were implemented for this project, linear regression and neural networks. Both methods will use the same preprocessing and future extraction procedures. The data set used was taken from: https://www.kaggle.com/datasets/plameneduardo/sarscov2-ctscan-dataset

For preprocessing, the code will read all images (CT scans) in designated folders and assign labels and store them in a label array accordingly. The program will then proceed to call a function called maskAlveoli() which takes in the images and return them in binary format with only the alveoli visible. The function does this applying binary thresholding with the ranges of alveoli which was determined through trial and error. The cv2.floodFill() method is then used on every pixel in each image to take out any deformities and to make sure that any alveoli that are blended with the background are included in the mask (since the background of the CT is within the color range of alveoli).

For feature extraction, a function called getFeatures() that was implemented from scratch is called. The function goes through all the images in the dataset to extract the needed features (number of inflamed alveoli and average size of inflamed alveoli in each image). The function does so by assigning labels to each connected area of pixels which would indicate the probable existence of inflamed alveoli. This was done using the measure method from skimage library. The number of labels is stored as the feature Number of inflamed alveoli. The function then loops through the labels and measures their size and calculate the average size of alveoli in the CT by adding up all the size and dividing them by the number of labels.

For linear regression, the regressor is trained using the training set which is 3/4 of the entire set which is standardized. The regressor uses the linear regression formulas to train the model:

$$x = (x_1, X_2, ..., x_D)^T \rightarrow D{-}dimensional feature vector$$

$$t = (t^1, t^2, ..., t^{(i)}) \rightarrow labels set$$

In Linear regression, the predictor is a linear function of the features where:

$$f(x_1, ..., x_D) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_D x_D \quad (1)$$

Let **W**=$(w_0, w_1, ...w_D)^T$ denote the vector of parameters where w_0 is the bias or intercept.

using least squares as the loss function to minimize the training error and solving the gradient, the following equation is obtained to calculate the optimal feature parameters vector:

$$w = (X^T X)^{-1} X^T t \quad (2)$$

Equation (2) was used to train the model on the training set, and after the parameters vector was obtained, it was used in Equation (1) to get predictions for the test set.

After the regressor is trained, the classification threshold is then calculated using least loss on the test set. The prediction is then acquired for the entire test set using the trained regressor and used to calculate the misclassification rate and accuracy and plot the confusion matrix.

For the neural network, tensor flow was used from the keras library. The model is initialized to have 2 hidden layers with 20 units in each hidden layer and relu function as the activation function for each hidden layer. The label set for the neural network is modified to have an array of 2 elements instead of ones and zeros to label the dataset (0=[0,1] and 1= [1,0]) as this is the required for the keras library. The model is then trained using the training set and a batch size of 10 and a thousand epochs. The prediction is acquired and is used to calculate the accuracy of the model and plot the confusion matrix.

Another sub method was also used to extract the features using an orb detector and storing the key points as features. However, the method did not achieve good results since the orb detector could not detect any key points in a lot of images which led to the elimination of a lot of images from the dataset and thus hindering the results of the method. The code for this method will also be included in the submission.

4. Results

4.1. Pre-Processing

The maskAlveoli function was proven to work very well in masking the required features (alveoli). This was determined by inspecting sample output from the function. The following samples show the results:

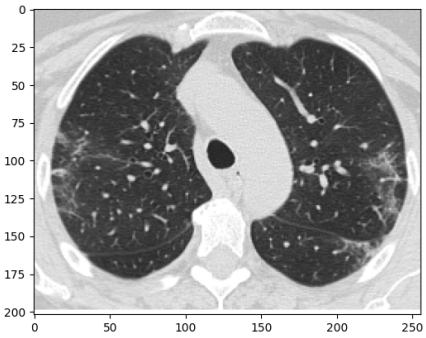

Figure 1. CT scan of an infected lung

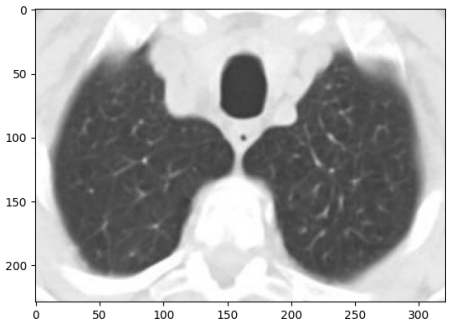

Figure 2. CT scan of a non-infected lung

From Figure 1, and Figure 2, it can be seen that the infected lungs have considerably more and larger visible alveoli, which justifies the proposed method of extracting these two features for recognition.
After applying the pre-processing function maskAlveoli(), the following results are obtained on two stages:

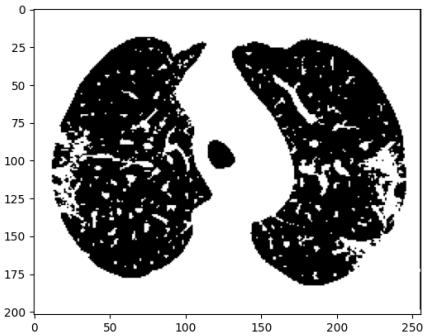

Figure 3. Infected lung after applying binary thresh holding using the color ranges of the alveoli (Stage 1)

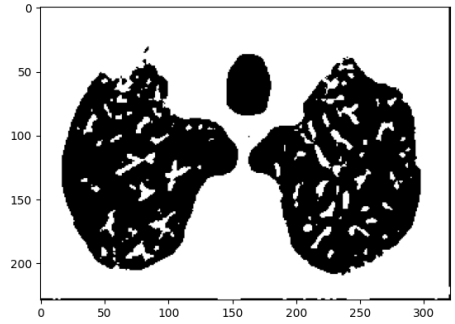

Figure 4. Non-infected lung after applying binary thresh holding using the color ranges of the alveoli (Stage 1)

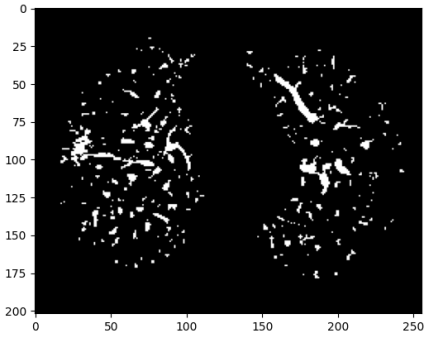

Figure 5. Infected lung after applying binary thresh holding using the color ranges of the alveoli (Stage 2)

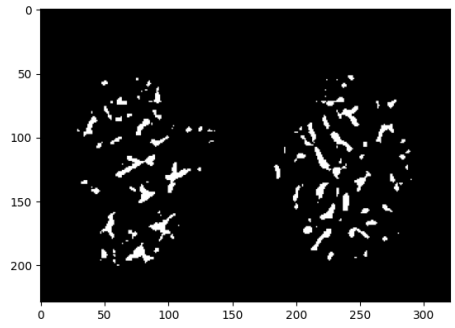

Figure 6. Non-infected lung after applying binary thresh holding using the color ranges of the alveoli (Stage 2)

Figures (5) and (6), represent the final output of the pre-processing stage of the program. The masked alveoli binary

images are then passed to the next stage which is feature extraction.

4.2. Feature Extraction

The pre-mentioned getFeatures() function extracts the features (number of visible alveoli, and average alveoli size) from the entire dataset and returns a two dimensional dataset that is ready for splitting, training then testing. The following table represents a sorted (based on label) random sample output of features from the data set:

Number of visible alveoli	Average size of alveoli	Label
316	6.813291	Covid
389	3.22108	Covid
465	3.974194	Covid
538	4.291822	Covid
463	4.816415	Covid
594	4.175084	Covid
512	4.289063	Covid
477	6.035639	Covid
495	4.30303	Covid
492	4.313008	Covid
443	3.586907	Covid
71	13.16901	Non-Covid
79	13.5443	Non-Covid
75	15.94667	Non-Covid
69	16.01449	Non-Covid
76	14.44737	Non-Covid
62	15.04839	Non-Covid
66	14.63636	Non-Covid
65	12.10769	Non-Covid
76	9.881579	Non-Covid

Table 1. Sample output for the getFeatures() function

From Table 1, it can be seen that the getFeatures() is very successful in extracting the features needed since the output data is very suitable for training the model and is linearly separable which is a key requirement for linear regression.

4.3. Recognition

For this stage of the program, two classifiers were trained and used, Linear regression and neural network. The following are the Confusion matrices output for both methods:

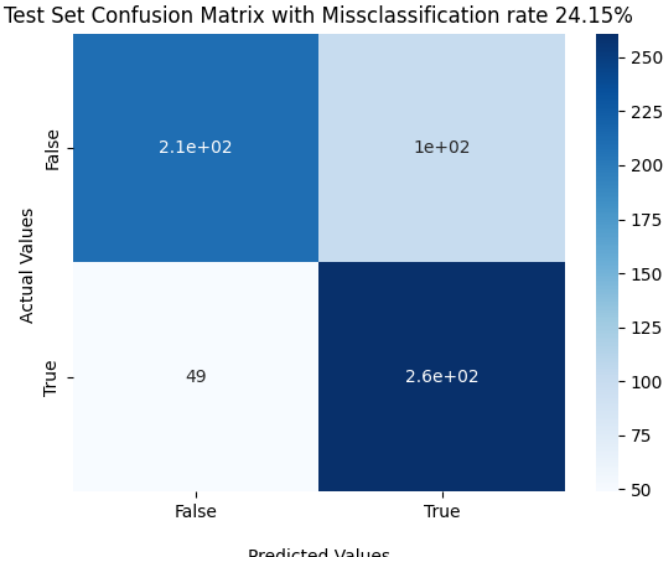

Figure 7. Confusion matrix for Linear regression using the test set

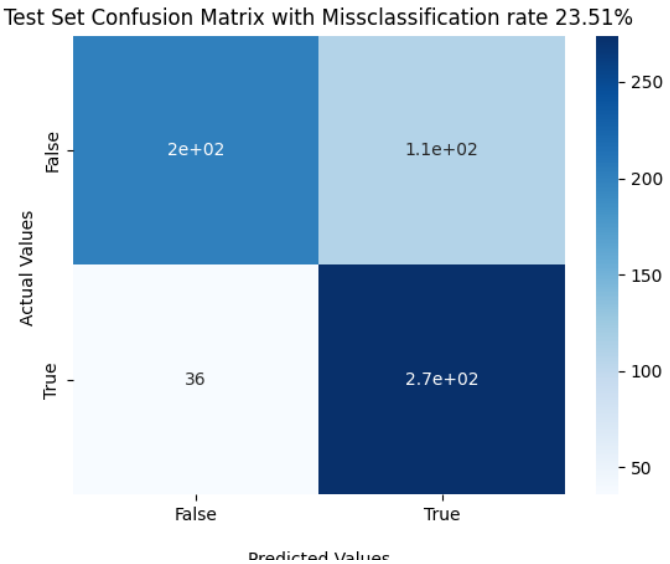

Figure 8. Confusion matrix for Neural Network model using the test set

From the confusion matrices in Figure(7) and Figure(8) it can be seen that both models have relatively high performance with a miss-classification rate of 24.15% for the linear regression and a miss-classification rate of 23.15% for the Neural network.

For the neural network method, it was found that the results were very close to those of the linear regression with a

slight increase in classification accuracy. However, the neural network model, minimizes the false negatives which in the case of this project is favourable since the cost of having false negatives is greater than false positives as falsely diagnosed COVID patients could result in the infection of more patients.

It was also found that as the number of epochs increases, the average accuracy increases as well. However, as the number of epochs increase, the run-time increases dramatically. It was found that the optimal number of epochs is 500 and best accuracy achieved at 10,000 epochs

4.4. Miss-Classifications

Since, both models had very similar performance which was unexpected since neural networks are supposed to have better results than linear regressors, and in aim to achieve better results, or point out why the models could not achieve better accuracy, a sample of miss-classified images was analyzed.

After careful studying of the operation of CT scan machine, it was found out that the CT scanners take a full 3D scan of the lungs, the machine then takes horizontal cut sections of the scan for display and analysis. Most of the miss-classified images had one common feature. The scans were of cut sections at the very tip of the lungs which had very minimal lung tissue visible in them and therefore, feature extraction was not able to perform optimally since the scans had little to no features for extraction. The following figures show samples of miss-classified images for clarification.

Figure 9. Miss-classified image from the test set

Figure 10. Miss-classified image from the test set

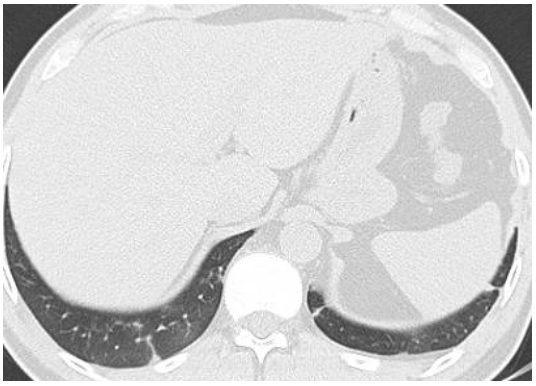

Figure 11. Miss-classified image from the test set

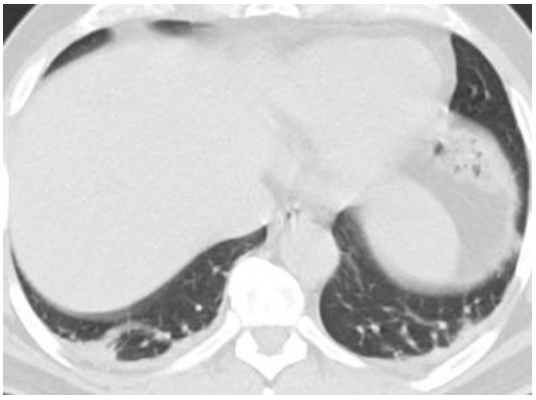

Figure 12. Miss-classified image from the test set

Figure 13. Miss-classified image from the test set

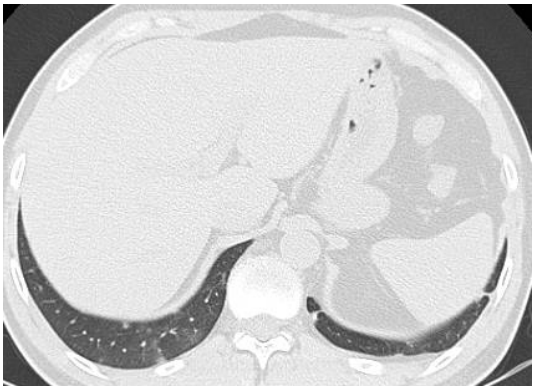

Figure 14. Miss-classified image from the test set

Figure 15. Miss-classified image from the test set

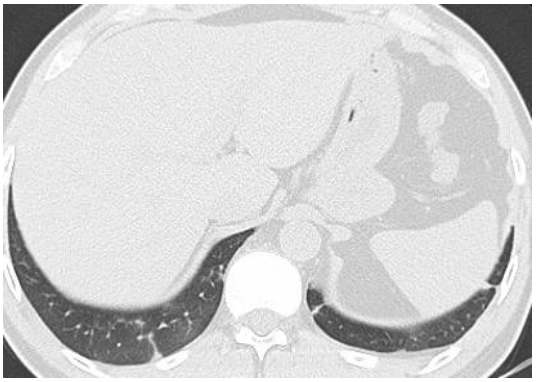

Figure 16. Miss-classified image from the test set

Figure 17. Miss-classified image from the test set

5. Conclusion

In conclusion, The pre-processing method and feature extraction method were very successful which was proven by the visual and data results. The classification models (linear regression, and neural networks) were also very successful in classifying the scans with miss-classification rates of 24.15% for linear regression and 23.51$ for the neural network. Even though, both models had very similar performance, the neural network model is preferred over the linear regression model as it minimizes the false negatives which in the case of this project is favourable since the cost of having false negatives is greater than false positives as falsely diagnosed COVID patient could result in the infection of more patients.

Moreover, it was found that the most of the miss-classified images, were miss-classified due to defects in the scans given in the data set as they had little to no features to extract. Therefore, it would be sound to conclude that with the proper dataset, much better results could be achieved with no changes to the program itself.

References

[1] American Lung Association. Pneumonia symptoms and diagnosis. 2021. 1

[2] Stephanie A. Haromn and Thomas H.Sanford. Artificial intelligence for the detection of covid-19 pneumonia on chest ct using multinational datasets. 1

[3] Imaging Technology News. Canon medical launches ct solution for patients with viral infectious diseases. 2020. 1

[4] Infection prevention and control Canada. Pandemic coronavirus(covid-19). 2022. 1

[5] Feng Shi. Large-scale screening of covid-19 from community acquired pneumonia using infection size-aware classification. 2