

Integration of Deep and Ensemble Learning for Detecting COVID-19 in Computed Tomography Images

Ali Haidar (✉ a.haidar@unsw.edu.au)

University of New South Wales <https://orcid.org/0000-0001-5092-949X>

Lois Holloway

University of New South Wales

Research Article

Keywords: COVID-19, Deep Learning, Ensemble Learning, Feature Extraction

Posted Date: November 16th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-82263/v2>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Integration of Deep and Ensemble Learning for Detecting COVID-19 in Computed Tomography Images

Ali Haidar^{1,2}, Lois Holloway^{1,2,3,4,5,6}

¹South Western Sydney Clinical School, University of New South Wales, Liverpool, NSW, Australia

²Ingham Institute for Applied Medical Research, Liverpool, NSW, Australia

³Centre for Medical Radiation Physics, University of Wollongong, Wollongong, NSW, Australia

⁴Centre for Oncology Education and Research Translation, Wollongong, NSW, Australia

⁵Liverpool and Macarthur Cancer Care Centres, Liverpool, NSW, Australia

⁶Institute of Medical Physics, University of Sydney, Camperdown, NSW, Australia

Corresponding author: Ali Haidar (a.haidar@unsw.edu.au)

Abstract. *This paper presents an approach for detecting covid-19 in Computed Tomography (CT) images by integrating deep convolutional neural networks and ensembles of decision trees. The proposed approach consisted of three steps. In the first step, the CT images slices were collected and processed. In the second step, a deep convolutional neural network was trained to predict covid-19 in the CT images. In the third step, deep features were extracted and were used to train an ensemble of decision trees. Six types/packages of ensembles of decision trees were investigated: extreme gradient boosting (XGBoost), bagged decision trees (BDT), random forest (RF), adaptive boosting decision trees (Adaboost), gradient boosting decision trees (GBDT), and dropouts meet multiple additive regression trees (DART). The accuracy, sensitivity, specificity, f1-score, precision, and area under the ROC curve (AUC) were calculated to compare the models against each other. The proposed approach revealed the highest performance with a RF that reported 0.87 accuracy, 0.87 f1-score, and 0.90 AUC. The developed models revealed similar performance when compared to previously published models. This highlights the efficiency of combining deep networks with ensembles of decision trees for detecting covid-19.*

Keywords: COVID-19; Deep Learning; Ensemble Learning; Feature Extraction

1. Introduction

Covid-19, a new pandemic that caused more than 600,000 deaths by July 2020 worldwide, has emerged in the last quarter of 2019 in the city of Wuhan, China. This pandemic has affected almost every aspect in human life and caused huge loss to different sectors.

Covid-19 can infect individuals if being in contact with an affected patient. Although several research groups are investigating possible cures and vaccines, covid-19 will remain in communities. Several methods have been followed in covid-19 diagnosis such as the reverse-transcriptase polymerase chain reaction (RT-PCR). Most of the tests are currently conducted for individuals who have symptoms or may have been in contact with a reported positive case. In addition, covid-19 can be detected (or at least show an indication) by interpreting medical images, such as chest X-ray and CT images. With enough data, diagnostic models can be trained to automatically detect covid-19. Although these approaches will not be used as the main method for covid-19 diagnosis, it can still be utilized in the medical sector. These detection approaches could assist in identifying covid-19 in patients admitted to scans not directly related to covid-19 tests, such as CT scans for cancer patients. This helps in warning health-care workers and the patient family members, where various actions might be considered. Since imaging systems are already well established, embedding

such diagnosis models to automatically detect the occurrence of such viruses is not considered that complicated and does not require expensive resources. In addition, the imaging techniques can be sometimes used and investigated, where the PCR may not show positive results as reported in [1].

Deep convolutional neural networks (CNNs) have been widely utilized in image related tasks since the emergence of Alexnet, a CNN that achieved the lowest error in ImageNet challenge in 2012 [2-4]. Subsequently, this type of neural networks has been investigated in medical related areas such as diagnosis, contouring, and prediction tasks [5-9]. A CNN was used to predict seizures in [9]. Diamant et al. utilized deep CNNs to predict outcome measures in head and neck cancer [10]. McCloskey et al. utilized CNNs to detect sleep apnea in wavelet spectrogram images [11]. A CNN was used to classify breast cancer in [12]. CNNs were also used to detect thyroid cancer based on different image types [13,14].

Furthermore, several methods have been proposed to detect/segment covid-19 by using CNNs [15-18]. Zhoa et al. developed a covid-19 dataset by collecting various images from different resources and combined the covid-19 images with other normal lung CT scans [19]. A detection study has been conducted by utilizing deep learning pre-trained convolutional neural network models [20]. The highest accuracy reported was 86%, with an f1-score of 85% and an AUC of 0.94. Other types of datasets were collected and made publicly available as the chest X-ray image dataset [21]. Transfer learning was also applied to detect covid-19 in chest X-ray images [22]. Zeng et al. utilized the genetic algorithm to optimize the dropout layer in a deep learning model to distinguish pneumonia and covid-19 [23]. Furthermore, four transfer learning models were also utilized to detect covid-19 in [24]. This shows the applicability of such techniques in detecting covid-19.

Interestingly, multiple machine learning algorithms can be combined to perform over the same task [25]. This is known as ensemble learning, where multiple algorithms are trained, and their output/predictions are combined. Ensembles of machine learning algorithms can lead to better accuracy and enhanced performance, if well trained. The ensembles can be generated by combining homogenous or heterogenous classifiers. Ensembles of decision trees are homogenous ensembles that have reported reasonable performance when used in medical related tasks (outcome prediction, secondary cancer prediction, etc.) [26,27].

Both deep learning and ensembles of decision trees have reported reasonable accuracies when deployed in various areas. Hence, a combination of the two techniques might reveal reliable performance in detecting covid-19. In this paper, we investigated a new approach for detecting covid-19 by establishing a machine learning model that extracts features from the CT images using a deep convolutional neural network, then uses the deep features to train an ensemble of decision trees. The rest of the paper is organized as follows; Section 2 details the collected dataset and describes the proposed approach. Section 3 details experimental setup, results, and comparative analysis. Section 4 derives a conclusion.

2. Materials and Methods

2.1 Data

A dataset that consisted of CT images slices corresponding to covid-19 and non-covid-19 patients was collected from GitHub [19] (28/04/2020). The authors provided a scheme that shows the images distribution over training, validation, and test cohorts, which was followed when building the models developed in this study. The same training technique was also followed by not applying any cross-validation and keeping the validation cohort as it is. The images distribution over the training, validation, and test cohorts is shown in Figure. 1.

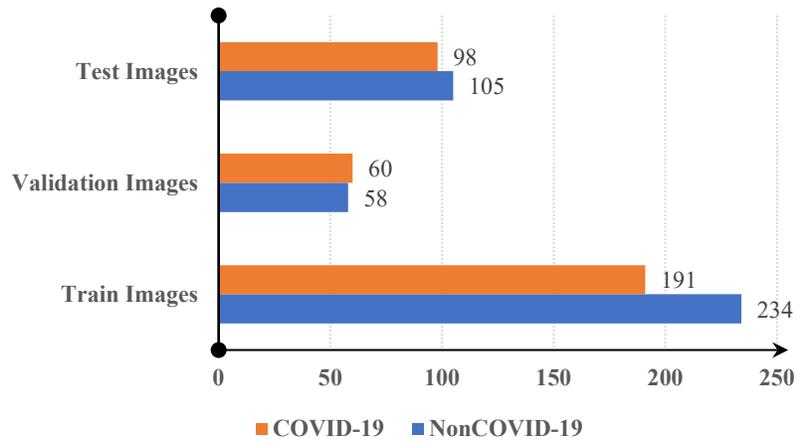


Figure. 1 Images distribution over the three cohorts (training, validation, and test).

2.2 Methods

Three steps were followed to develop the proposed approach as shown in Figure 2. In the first step, the images were collected and processed. In the second step, a deep learning model was trained to detect covid-19 in CT image slices. For each image, the deep learning model was trained to distinguish between images with normal lungs and images with covid-19. In the third step, features were extracted from the deep learning model, then an ensemble of decision trees was trained to classify covid-19.

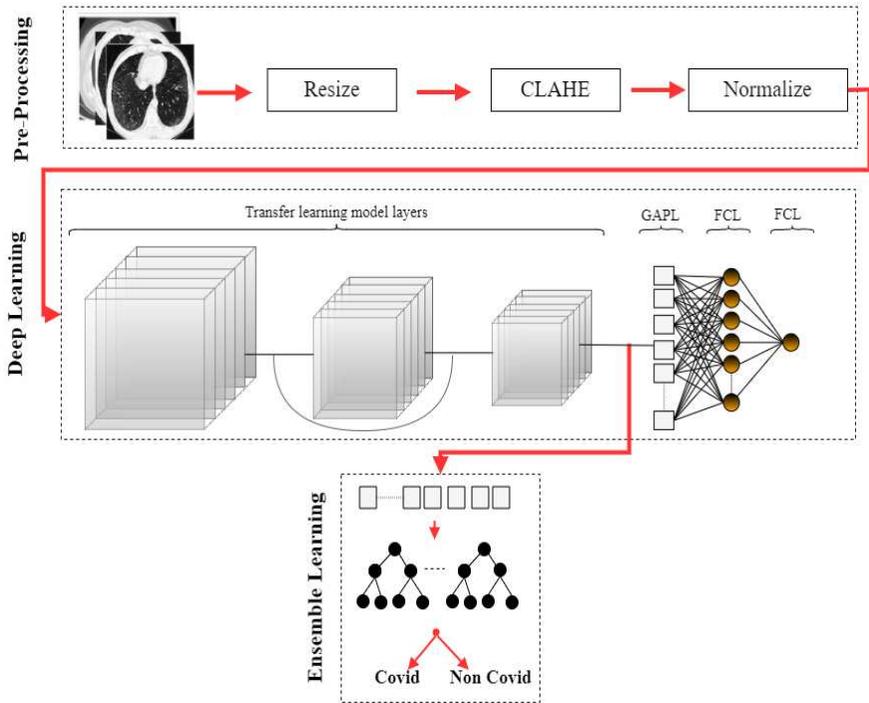


Figure. 2 Integration of Deep and Ensemble Learning for Detecting COVID-19.

Pre-processing

The collected images were resized into $224 \times 224 \times 3$, which is the default size used in many pre-trained deep learning models (DenseNet121, DenseNet169, DenseNet201, etc. In addition, the images were processed by applying the contrast limited adaptive histogram equalization (CLAHE) to enhance local contrast in lung images (e.g. Figure 3) [28].

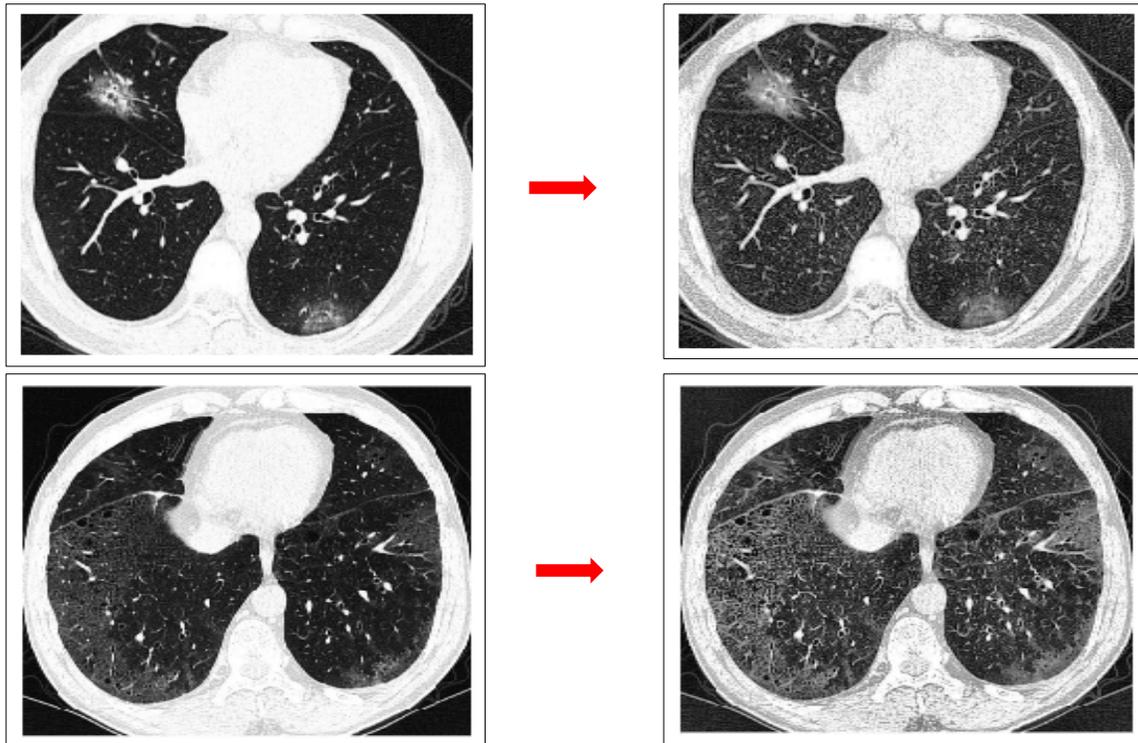


Figure. 3 CLAHE example (left side: original image; right side: processed image).

Deep Learning

The CNN consists of multiple types of layers that process the image pixels to learn and extract meaningful features. The CNN consists of convolutional, pooling, and dropout layers. Fully connected layers are added to enhance the learning process after the main layers.

Two approaches can be followed to develop a CNN. In the first approach, the deep CNN is built from scratch, where combinations of convolutional, pooling, and dropout layers are stacked together. In the second approach, models pre-trained on different types of images are tuned to match the current task requirements, then trained using the dataset images [29]. This approach is referred to as transfer learning, where knowledge is transferred from a different domain to the area under consideration, which is in this case the detection of covid-19 in lung CT images.

We utilized the second approach by using a deep learning model previously trained on the ImageNet dataset. Several types of transfer learning models have been released and made available: DenseNet169, DenseNet121, Xception, ResNet, etc [30-32]. These pre-trained models vary by their layers, number of layers, connections between layers, size, etc. The performance of these algorithms differs based on the problem under consideration. The DenseNet169 has reported reliable accuracy in previous methods [20].

Hereafter, it was considered as the feature extractor model in this study. It consists of 169 layers and 14,307,880 parameters.

The last fully connected layer was removed from the DenseNet169. A global average pooling layer (GAPL) and two new fully connected layers (FCL) were added. The first fully connected layer consisted of 13 neurons with the sigmoid activation function. The last fully connected layer consisted of one neuron with a sigmoid activation function. Table 1 details the parameters used to train the deep learning model. The “adam” optimizer was selected with a small learning rate to avoid quick convergence to a local optimum due to the limited small size of the dataset [33]. The early stopping flag was selected to 4 to avoid overfitting. In addition, the model with the lowest loss over the validation dataset was tracked while training. 16 CPUs were utilized in Katana (a UNSW higher performance computing system) to train the deep learning model.

Table 1 Transfer learning model parameters.

Training parameters	DenseNet169 values
Optimizer	Adam
Learning rate	0.00001
Epochs	15
Batch size	4
Loss function	Binary cross-entropy
Early stopping	4

Ensembles Learning

Deep neural networks are efficient in extracting features from images by the way it processes pixels (convolution, pooling). We propose extracting features from the developed deep learning model and training an ensemble of decision trees as shown in Figure. 2. Each image was mapped into a set of features equal to the size of the output from the selected layer for extraction. The training and validation features were used to build the ensemble, while the extracted features from the hold-out sample were used for assessment. The features were extracted from the last global average pooling layer (GAPL). 1664 features were extracted from the global average pooling layer.

An ensemble consists of two main components: base classifiers and a fusion method. The base classifiers are the algorithms used to generate the ensemble. The fusion method determines the technique the outputs of each base classifier are combined (average, weighted average, stacking, etc.). The base classifiers might belong to the same/different types of algorithms. For e.g. neural networks can be combined with support vector machines and decision trees to form up an ensemble. In addition, multiple decision trees can be combined to form up an ensemble. Several types of ensembles of decision trees were investigated: extreme gradient boosting (XGBoost), bagged decision trees (BDT), random forest (RF), adaptive boosting decision trees (Adaboost), gradient boosting decision trees (GBDT), and dropouts meet multiple additive regression trees (DART). These algorithms belong to the same group of ensembles but vary with training methods. Boosting is an ensemble technique that updates the misclassified samples weights while sequentially creating new classifiers and adding it to the ensemble. XGBoost, Adaboost, GBDT, and DART are variations of ensembles that differ based on the boosting algorithms. Bagging is another form of ensembles, where random subsamples with replacement are taken from the dataset for training each base classifier [34]. For performing a prediction, the probability (average or majority vote) given by each base classifier (decision tree) is taken to release the final probability [34]. In random forest, random subsamples are taken from the dataset to train the base classifiers (decision trees). However, in this case, the samples and features

are subsampled. Since these ensembles use decision trees as base classifiers, they held similar hyper-parameters that includes the learning rate, maximum depth, subsample, etc.

2.3 Performance Metrics

To assess the performance of the proposed approach, six evaluation metrics were used: accuracy, precision, sensitivity, specificity, f1-score and AUC. True positives (TP) represents the number of covid-19 samples that were correctly classified, true negatives (TN) represents the number of non-covid-19 samples that were correctly classified, false positives (FP) represents the number of covid-19 samples that were misclassified as non-covid-19, false negatives (FN) represents the non-covid-19 samples that were misclassified as covid-19. Accuracy is the total number of samples that were correctly classified. Precision is the number of the correctly classified covid-19 samples divided by the total number of covid-19 samples. Sensitivity (recall) is the total number of correctly classified covid-19 samples divided by the total number of samples classified as covid-19. Specificity is the number of the correctly classified non-covid-19 samples divided by the total number of samples classified as non-covid-19 samples. The f1-score measures the robustness of the model. The AUC measures how well the model can separate the two classes (covid-19 and non-covid-19) based on different thresholds. Accuracy, precision, sensitivity, specificity, and f1-score are represented in the following equations:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$precision = \frac{TP}{TP + FP}$$

$$sensitivity = \frac{TP}{TP + FN}$$

$$specificity = \frac{TN}{TN + FP}$$

$$f1 = \frac{precision * sensitivity}{precision + sensitivity} * 2$$

3. Experiments and Results

The deep learning neural network was trained to minimize the error between the actual and predicted classes by using the 'binary_crossentropy' loss function. Keras and Tensorflow were used to develop and train the DenseNet169 [35,36]. Due to the limited size of the dataset, data augmentation was applied. The rotation range was selected to +30/-30. The zoom range and shear range were selected to 0.2. The results were generated as probabilistic values since the sigmoid activation function was used in the last layer. It was noticed that the model started to overfit after few epochs and was stopped by the early stopping flag. This is justifiable taking into consideration the small number of samples in the dataset. The developed model was evaluated over the hold-out sample that consisted of 203 images. Accuracy, f1-score, AUC of the trained DenseNet169 are shown in Table 2.

Table 2 DenseNet169 accuracy, f1-score, and AUC

Model	Acc	F1-score	AUC
DenseNet169	0.82	0.80	0.89

Six ensembles of decision trees were developed by using the extracted features from the developed DenseNet169. The xgboost package was used to implement the XGBoost [37]. RF, Adaboost, and BDT were implemented using the scikit-learn python package [38]. GDBT and DART were developed using the lightgbm package [39]. Random search was followed to tune the hyper-parameters of the ensembles. Several parameters were tuned including the number of estimators, maximum depth, learning rate, etc. Table 3 represents the accuracy, sensitivity, specificity, precision, f1-score and AUC for each of the developed ensembles over the hold-out sample.

Table 3 Accuracy, sensitivity, specificity, precision, f1-score, and AUC for each model.

Model	Acc	Sensitivity	Specificity	Precision	F-1 score	AUC
DenseNet169-XGBoost	0.87	0.87	0.87	0.87	0.87	0.90
DenseNet169-RF	0.87	0.90	0.85	0.85	0.87	0.90
DenseNet169-BDT	0.85	0.89	0.82	0.82	0.85	0.90
DenseNet169-Adaboost	0.72	0.63	0.80	0.8	0.71	0.84
DenseNet169-LGBM	0.84	0.82	0.86	0.86	0.84	0.90
DenseNet169-DART	0.87	0.85	0.89	0.89	0.87	0.90

The developed ensembles received comparable accuracy, except for the DenseNet169-Adaboost. This could be linked to fact that Adaboost ensemble uses decision stumps. With this high number of features, multiple level decision trees might be needed. The combined models revealed better accuracy compared to the deep network alone in terms of accuracy, f-score, and AUC. Hence, the combination of deep learning and ensembles enhanced the overall performance. The highest sensitivity was obtained with the DenseNet169-RF ensemble (0.90). We analyzed the misclassified images, where 6% of the covid-19 images were misclassified by all the models including the transfer learning model. In addition, 6% of the non-covid-19 images were misclassified as covid-19 by all the developed models.

The true positives, true negatives, false positives, and false negatives are represented in Table 4. The DenseNet169-RF revealed the lowest number of misclassified cases (26) followed by DenseNet169-XGBoost and DenseNet169-DART (27). The DenseNet169-Adaboost had the highest number of misclassified images (57).

Table 4 True positives, True negatives, false positives, and false negatives obtained over the test sample for each of the developed models.

Model	True Positives	True Negatives	False Positives	False Negatives
DenseNet169-XGBoost	85	91	14	13
DenseNet169-RF	88	89	16	10
DenseNet169-BDT	87	86	19	11
DenseNet169-Adaboost	62	84	21	36
DenseNet169-LGBM	80	90	15	18
DenseNet169-DART	83	93	12	15

The effectiveness of the transfer learning model in extracting meaningful and informative features was analyzed. Significance t-tests over the hold-out sample extracted features were applied, assuming that the

features values follow a normal distribution. The hold-out sample consisted of 98 covid-19 images and 105 non-covid-19 images. The last 7 images were removed to make the samples equal. Each feature values were grouped based on each class (covid or non-covid). For each feature, two groups were generated: the first group contains the values of the feature for the covid-19 images, while the second group contains the values of the feature for the non-covid-19 images.

The initiated null hypothesis was “*the two groups in each feature are similar and are not significantly different*”. In other words, there is no significance difference between the covid-19 image extracted features value and non-covid-19 image extracted feature values. The t-tests were conducted over each feature groups, with a p-value of 0.05. The null hypothesis was rejected for 1203 features out of the 1664 extracted features. 72% of the extracted features contained groups that are significantly different. This shows the effectiveness of transfer learning approaches in extracting such meaningful representations for each class in CT images slices.

The DenseNet169-DART revealed the lowest number of false positives. However, the DenseNet-RF showed the lowest number of misclassified images. With the RF, the features contributing to prediction were summarized and were arranged in descending order. As expected, the top 10 and 50 contributing features used by the RF in covid-10 classification were among the 72% significant features (1203 features), only 4% of the top 100 important features were not significant features, and 10.5% of the top 200 contributing features were not significant.

The top 10 performing features in the DenseNet169-RF were grouped based on each class. Boxplots of each feature are shown in Figure 4. Different ranges of values were obtained with each feature. It can be noticed that in most of the features, the distribution varied for each class. For 7 features out of the 10, the data ranges between the 25th and the 75th percentile were different. Hereafter, the combination of deep learning models with ensembles of decision trees can be considered in such tasks. We assume that these representations were implicit descriptors of the differences between covid-19 and non-covid-19 images. This kind of algorithms can be used in the healthcare sector, while taking CT scans for another types of diseases such as cancer. Probabilities in CT image slices may alert the healthcare workers and the patient’s family about the possible infection.

The top three performing models (DenseNet169-XGBoost, DenseNet169-RF, and DenseNet-DART) were compared to deep learning models developed in [20]. Several models were generated following different methods of training. The best performing model revealed 0.86, 0.85, and 0.94 accuracy, f1-score, and AUC respectively. As shown in Table 5, the three models revealed better accuracy and f1 score, while showing 4% decrease in AUC. It should be also mentioned that the AUC in the three models did not vary much compared to the developed transfer learning model used in this approach.

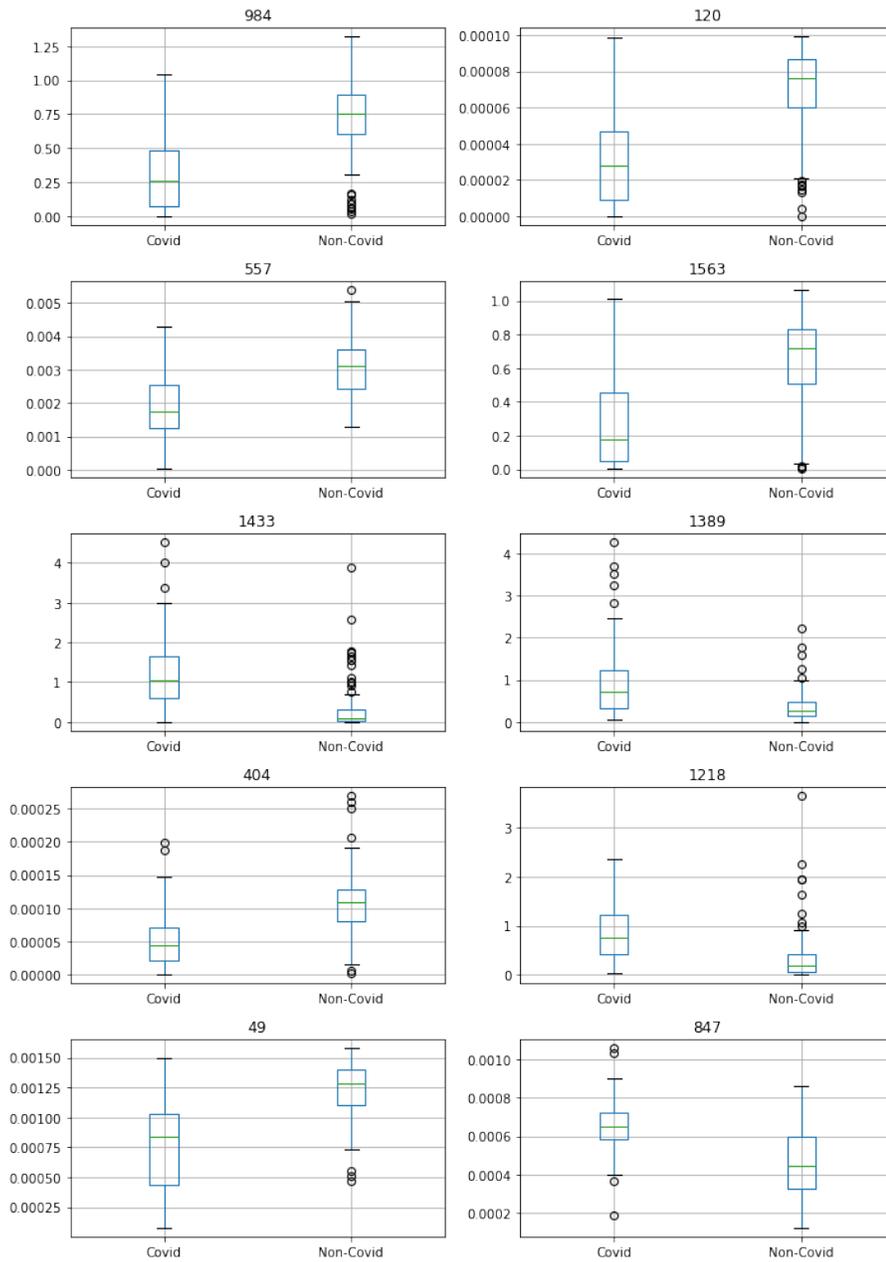


Figure 4 Boxplots for each feature based on each class over the test sample.

Table 5 Comparison to models developed by He et al [20].

Model	Accuracy	f1-score	AUC
ResNet-50 (method 1)	0.69	0.63	0.66
ResNet-50 (method 2)	0.83	0.83	0.91
ResNet-50 (method 3)	0.75	0.78	0.85
ResNet-50 (method 4)	0.81	0.81	0.88
ResNet-50 (self-trans)	0.84	0.83	0.91
DenseNet-169 (method 1)	0.70	0.63	0.77
DenseNet-169 (method 2)	0.86	0.85	0.91
DenseNet-169 (method 3)	0.74	0.76	0.86
DenseNet-169 (method 4)	0.83	0.82	0.88
DenseNet-169 (self-trans)	0.86	0.85	0.94
<i>DenseNet169-XGBoost (ours)</i>	0.87	0.87	0.90
<i>DenseNet169-RF (ours)</i>	0.87	0.87	0.90
<i>DenseNet169-DART (ours)</i>	0.87	0.87	0.90

*method 1, method 2, method 3, method 4, and self-trans refers to multiple training methods proposed in He et al [20]

4. Conclusion

This paper presented an approach for detecting COVID-19 in CT images by using deep learning and ensembles of decision trees. Deep features were extracted from the CT images using deep learning convolutional neural networks and were used to train the ensembles of decision trees. The proposed approach revealed comparable performance to previously published models, which highlighted the efficiency of combining deep networks with ensembles of decision trees for detecting covid-19. The extracted deep features were investigated, and significant features were highlighted. One of the key limitations to this study is the limited number of training samples. Future work will include assessing the proposed approach over other covid-19 datasets and using other deep learning models such as the Xception, DenseNet121, ResNet, etc.

Acknowledgements

This research includes computations using the computational cluster Katana supported by Research Technology Services at UNSW Sydney.

Funding (information that explains whether and by whom the research was supported)

None

Conflicts of interest/Competing interests (include appropriate disclosures)

None

Availability of data and material (data transparency)

Public repository on GitHub (<https://github.com/UCSD-AI4H/COVID-CT>)

Code availability (software application or custom code)

https://github.com/ayhaidar/covid_ensembles

References

1. Feng H, Liu Y, Lv M, Zhong J (2020) A case report of COVID-19 with false negative RT-PCR test: necessity of chest CT. *Japanese journal of radiology*:1-2
2. Krizhevsky A, Sutskever I, Hinton GE Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, 2012. pp 1097-1105
3. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009. pp 248-255
4. Zhang L, Verma B (2019) A deep neural network and rule-based technique for fire risk identification in video frames. *Pattern Analysis and Applications* 22 (1):187-203. doi:10.1007/s10044-018-0756-6
5. Shen S, Han SX, Aberle DR, Bui AA, Hsu W (2019) An interpretable deep hierarchical semantic convolutional neural network for lung nodule malignancy classification. *Expert Systems with Applications* 128:84-95. doi:<https://doi.org/10.1016/j.eswa.2019.01.048>
6. Jin T, Cui H, Zeng S, Wang X Learning Deep Spatial Lung Features by 3D Convolutional Neural Network for Early Cancer Detection. In: *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 29 Nov.-1 Dec. 2017 2017. pp 1-6. doi:10.1109/DICTA.2017.8227454
7. Feng X, Qing K, Tustison NJ, Meyer CH, Chen Q (2019) Deep convolutional neural network for segmentation of thoracic organs-at-risk using cropped 3D images. *Medical Physics* 46 (5):2169-2180. doi:10.1002/mp.13466
8. Ying X, Yu Z, Yu R, Li X, Yu M, Zhao M, Liu K Thyroid Nodule Segmentation in Ultrasound Images Based on Cascaded Convolutional Neural Network. In: *International Conference on Neural Information Processing*, Cham, 2018. Neural Information Processing. Springer International Publishing, pp 373-384
9. Truong ND, Nguyen AD, Kuhlmann L, Bonyadi MR, Yang J, Ippolito S, Kavehei O (2018) Convolutional neural networks for seizure prediction using intracranial and scalp electroencephalogram. *Neural Networks* 105:104-111. doi:<https://doi.org/10.1016/j.neunet.2018.04.018>
10. Diamant A, Chatterjee A, Vallières M, Shenouda G, Seuntjens J (2019) Deep learning in head & neck cancer outcome prediction. *Scientific Reports* 9 (1):2764. doi:10.1038/s41598-019-39206-1
11. McCloskey S, Haidar R, Koprinska I, Jeffries B Detecting Hypopnea and Obstructive Apnea Events Using Convolutional Neural Networks on Wavelet Spectrograms of Nasal Airflow. In, Cham, 2018. *Advances in Knowledge Discovery and Data Mining*. Springer International Publishing, pp 361-372
12. Bardou D, Zhang K, Ahmad SM (2018) Classification of breast cancer based on histology images using convolutional neural networks. *IEEE Access* 6:24680-24693. doi:10.1109/ACCESS.2018.2831280
13. Li X, Zhang S, Zhang Q, Wei X, Pan Y, Zhao J, Xin X, Qin C, Wang X, Li J, Yang F, Zhao Y, Yang M, Wang Q, Zheng Z, Zheng X, Yang X, Whitlow CT, Gurcan MN, Zhang L, Wang X, Pasche BC, Gao M, Zhang W, Chen K (2019) Diagnosis of thyroid cancer using deep convolutional neural network models applied to sonographic images: a retrospective, multicohort, diagnostic study. *The Lancet Oncology* 20 (2):193-201. doi:10.1016/S1470-2045(18)30762-9
14. Ko SY, Lee JH, Yoon JH, Na H, Hong E, Han K, Jung I, Kim EK, Moon HJ, Park VY (2019) Deep convolutional neural network for the diagnosis of thyroid nodules on ultrasound. *Head & neck* 41 (4):885-891
15. Oh Y, Park S, Ye JC (2020) Deep learning covid-19 features on cxr using limited training data sets. *IEEE Transactions on Medical Imaging*
16. Wang L, Wong A (2020) COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images. arXiv preprint arXiv:200309871
17. Fan D-P, Zhou T, Ji G-P, Zhou Y, Chen G, Fu H, Shen J, Shao L (2020) Inf-Net: Automatic COVID-19 Lung Infection Segmentation from CT Scans. arXiv preprint arXiv:200414133

18. Butt C, Gill J, Chun D, Babu BA (2020) Deep learning system to screen coronavirus disease 2019 pneumonia. *Applied Intelligence*. doi:10.1007/s10489-020-01714-3
19. Zhao J, Zhang Y, He X, Xie P (2020) COVID-CT-Dataset: a CT scan dataset about COVID-19. arXiv preprint arXiv:200313865
20. He X, Yang X, Zhang S, Zhao J, Zhang Y, Xing E, Xie P (2020) Sample-Efficient Deep Learning for COVID-19 Diagnosis Based on CT Scans. medRxiv:2020.2004.2013.20063941. doi:10.1101/2020.04.13.20063941
21. Cohen JP, Morrison P, Dao L (2020) COVID-19 image data collection. arXiv preprint arXiv:200311597
22. Apostolopoulos ID, Mpesiana TA (2020) Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*:1
23. Zeng Z, Wang B, Zhao Z (2020) Research on CNN-based Models Optimized by Genetic Algorithm and Application in the Diagnosis of Pneumonia and COVID-19. medRxiv
24. Chowdhury ME, Rahman T, Khandakar A, Mazhar R, Kadir MA, Mahbub ZB, Islam KR, Khan MS, Iqbal A, Al-Emadi N (2020) Can AI help in screening viral and COVID-19 pneumonia? arXiv preprint arXiv:200313145
25. Zhang C, Ma Y (2012) *Ensemble machine learning: methods and applications*. Springer,
26. Wang R, Weng Y, Zhou Z, Chen L, Hao H, Wang J (2019) Multi-objective ensemble deep learning using electronic health records to predict outcomes after lung cancer radiotherapy. *Physics in Medicine & Biology* 64 (24):245005
27. Ye X, Li H, Sakurai T, Shueng P-W (2019) Ensemble Feature Learning to Identify Risk Factors for Predicting Secondary Cancer. *Int J Med Sci* 16 (7):949-959. doi:10.7150/ijms.33820
28. Pizer SM, Amburn EP, Austin JD, Cromartie R, Geselowitz A, Greer T, ter Haar Romeny B, Zimmerman JB, Zuiderveld K (1987) Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing* 39 (3):355-368
29. Tajbakhsh N, Shin JY, Gurudu SR, Hurst RT, Kendall CB, Gotway MB, Liang J (2016) Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transactions on Medical Imaging* 35 (5):1299-1312. doi:10.1109/TMI.2016.2535302
30. Huang G, Liu Z, Maaten LVD, Weinberger KQ Densely Connected Convolutional Networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21-26 July 2017 2017. pp 2261-2269. doi:10.1109/CVPR.2017.243
31. Chollet F Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition, 2017*. pp 1251-1258
32. He K, Zhang X, Ren S, Sun J Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition, 2016*. pp 770-778
33. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
34. Breiman L (1996) Bagging predictors. *Machine learning* 24 (2):123-140
35. Chollet F (2015) Keras. GitHub. <https://github.com/fchollet/keras>.
36. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M TensorFlow: A System for Large-Scale Machine Learning. In: *OSDI, 2016*. pp 265-283
37. Chen T, Guestrin C Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016*. ACM, pp 785-794
38. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V (2011) Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12 (Oct):2825-2830
39. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y Lightgbm: A highly efficient gradient boosting decision tree. In: *Advances in Neural Information Processing Systems, 2017*. pp 3146-3154

Figures

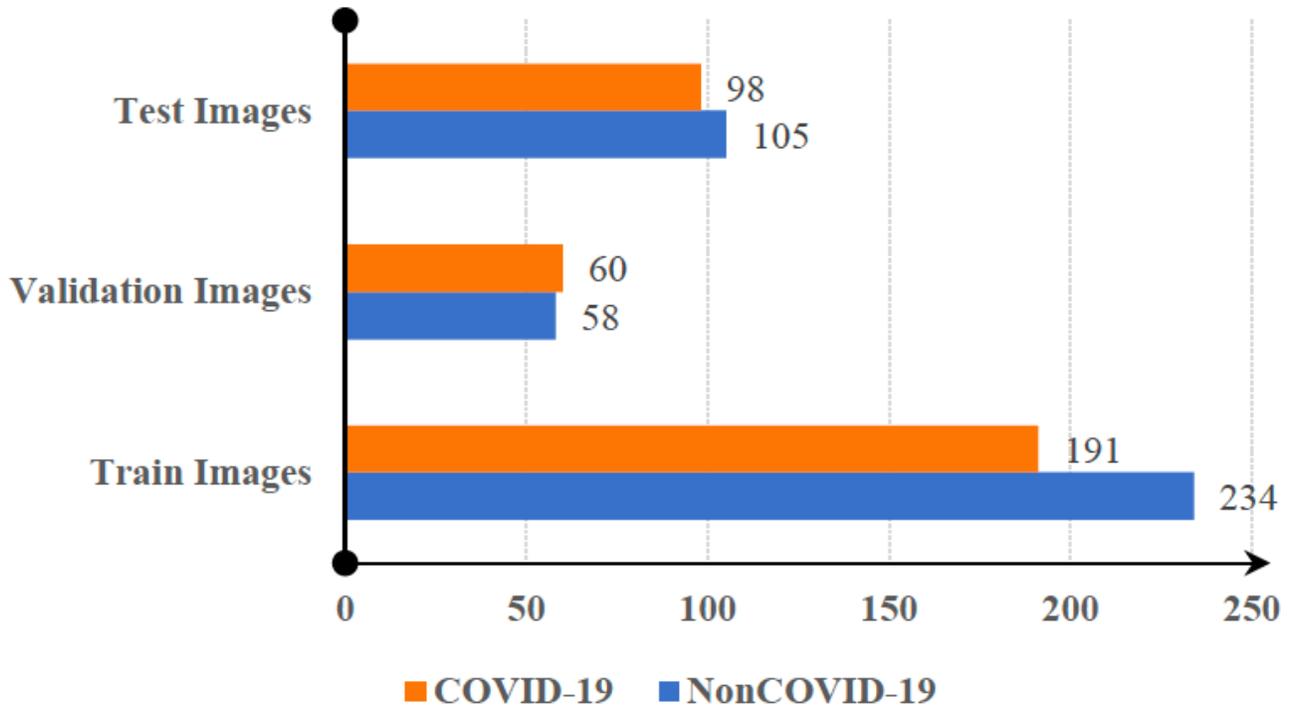


Figure 1

Images distribution over the three cohorts (training, validation, and test).

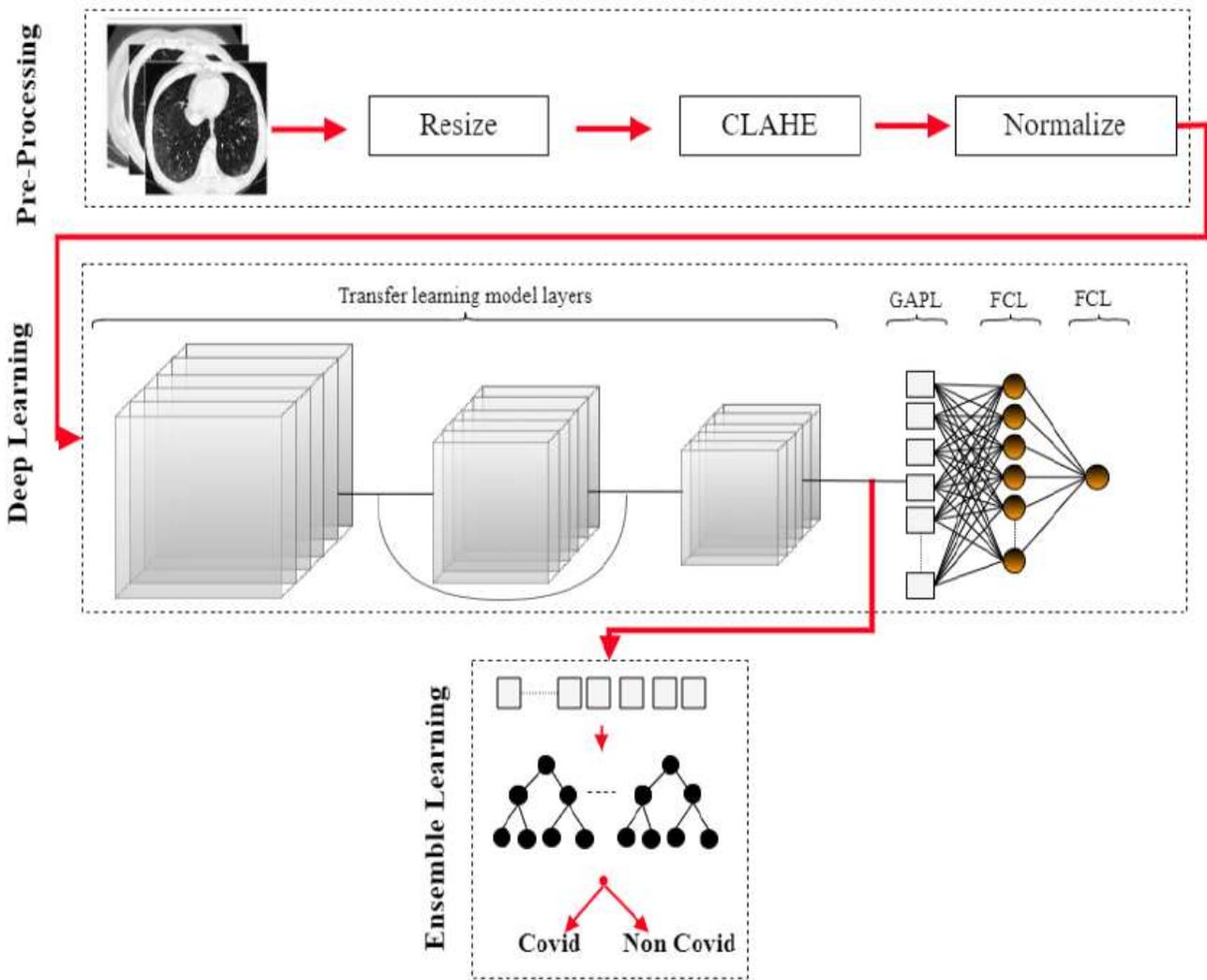


Figure 2

Integration of Deep and Ensemble Learning for Detecting COVID-19.

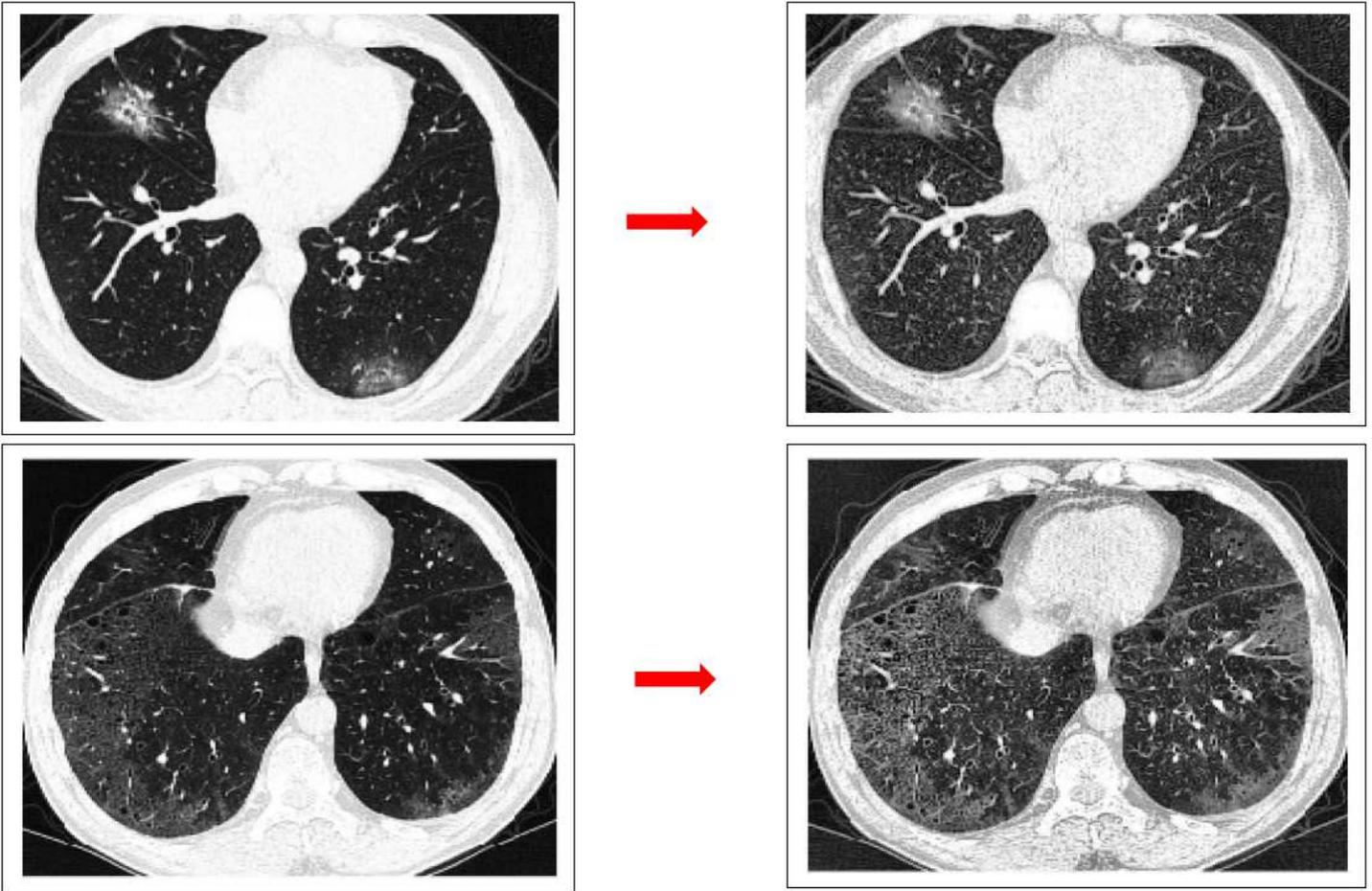


Figure 3

CLAHE example (left side: original image; right side: processed image).

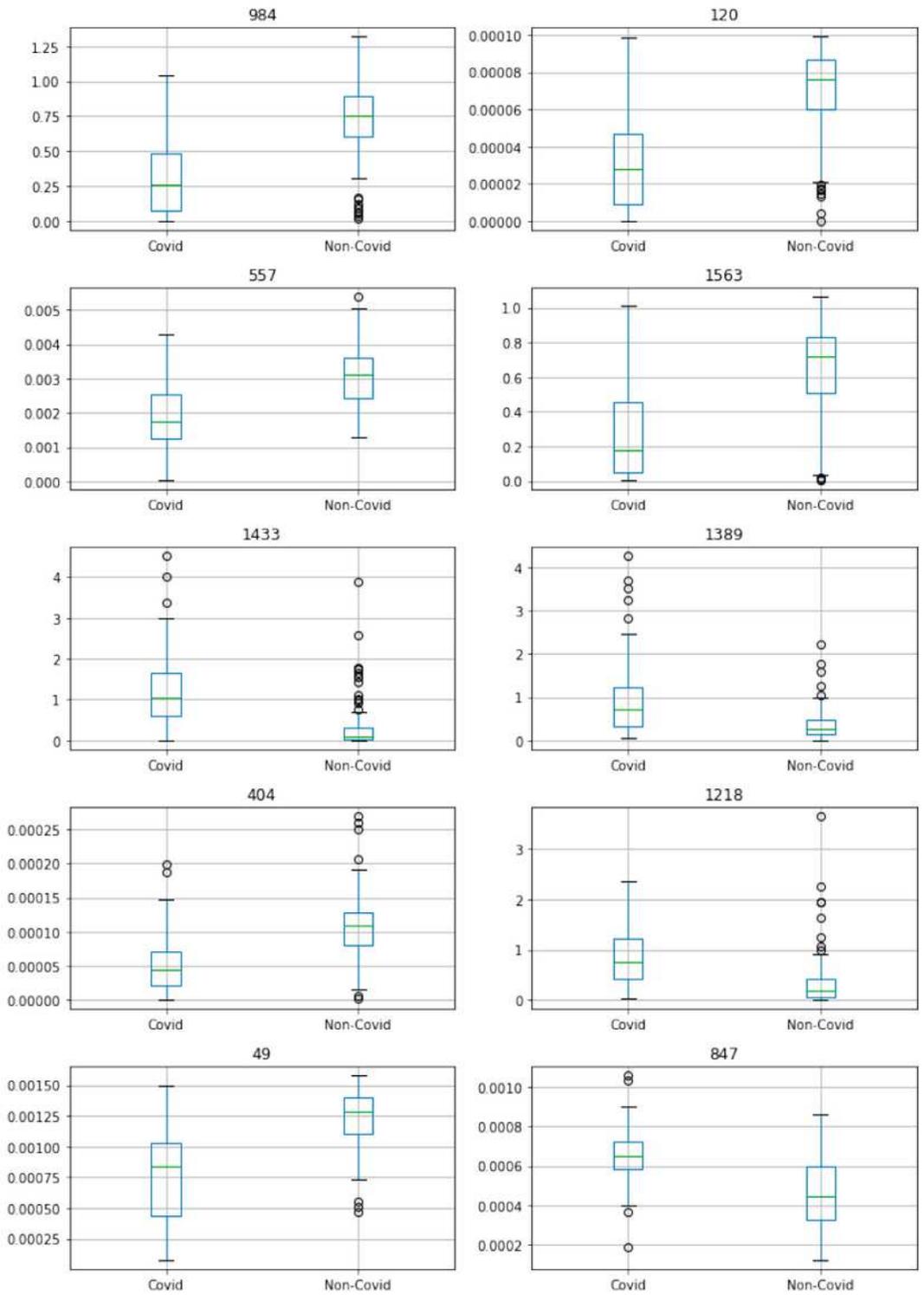


Figure 4

Boxplots for each feature based on each class over the test sample.