GPU ENABLED CNN BASED COVID-19 HRCT IMAGES CLASSIFIER

Preethi S V[1] Samyuktha G[2] Varsha Devi K[3] Kavin Raj A C[4] Pranav Shankar Ramalingam[5]	**Karpagam G R [6]**
Students Department of Computer Science and Engineering PSG College of Technology Coimbatore, Tamil Nadu, India.	Professor Department of Computer Science and Engineering PSG College of Technology Coimbatore, Tamil Nadu, India.
[1] preethisureshvidhya@gmail.com [2] samyukthagopalsamy@gmail.com [3] varshatech.2000@gmail.com [4] kavinraj.chellamuthu@gmail.com [5] srpranav15@gmail.com	[6] grk.cse@psgtech.ac.in

ABSTRACT

The Convolutional Neural Network based COVID-19 HRCT Images Classifier automatically classifies high resolution computed tomography(HRCT) lung images to diagnose COVID-19 and predict the severity of lung infection caused by the coronavirus infection. This automated HRCT Images Classifier takes the HRCT lung scans as input and is standard for the model. It is used to classify the scans and predict the severity score. With that information, the doctors can quickly analyze hundreds of CT scans for each patient. The model is trained using Deep learning with 30,000 HRCT images of lungs with varying lung infection severity. We have used a 3D U-Net architecture for segmenting the lung from the CT scans and 3D CNN architecture for classification and scoring of the severity of the CT scans. The objective is to leverage the CNN Deep Learning model to analyze CT scans automatically. Using the Image classifier would help ease the stress on the limited workforce of radiology experts and other resources and guide appropriate treatment protocols for COVID-19.

KEYWORDS

High Resolution Computed Tomography, Convolutional Neural Network, U-Net based Segmentation, COVID-19, RT-PCR

MOTIVATION

The ongoing COVID-19 pandemic has caused a bottleneck in the healthcare system around the world. The coronavirus is highly contagious as it spreads through air droplets and person to person contact. The standard method of diagnosing such viruses is the RT PCR (Reverse Transcription-Polymerase Chain Reaction) method. However, there are numerous issues in using RT-PCR. It is prone to produce many false-negative and false-positive results because of its low sensitivity and specificity. In this scenario, doctors can use medical imaging such as X-ray and Computed Tomography (CT) of the patients' lung to make the diagnosis, as the disease primarily targets the lung's epithelial cells. As CT scanners are widely available, they are considered to be functional and practical diagnostic tools. Artificial intelligence (AI) technologies can strengthen imaging tools' power and help medical specialists quickly analyze hundreds of CT images. Deep

learning provides state-of-the-art performance for detection, segmentation, classification, and prediction. Hence, deep learning offers a convenient tool for diagnosing and predicting lung infection severity in CT images for COVID-19 patients.

TERMS AND TERMINOLOGIES

GPU Computing: GPU computing accelerates CPUs' performance for general-purpose scientific and engineering computing using GPU (graphics processing unit) as a co-processor to the CPUs. The GPU accelerates applications running on the CPU by offloading some of the compute-intensive and time-consuming portions of the code.

Deep Learning: Deep learning models (which are organized in tens or hundreds of hidden layers consisting of a set of interconnected nodes) are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction.

Convolutional Neural Network: Convolutional Neural Network convolves learned features with input data and uses 2D convolutional layers, making the deep neural network architecture well suited to processing 2D data, such as images. 3D CNNs can be constructed using 3D convolution layers followed by 3D pooling layers.

Segmentation: The computer-based process of identifying the boundaries from surrounding thoracic tissue on computed tomographic (CT) images is called segmentation. This is a vital first step in radiologic pulmonary (lung-related) image analysis.

Lung Quantification: Lung quantification refers to calculating the ratio of the volume of the lung infected. In this context, lung quantification refers to assigning scores based on the severity of COVID-19 infection.

Binary Classification: Binary Classification refers to the process of classifying the images into two different groups based on some features. In this context, Binary Classification refers to classifying whether a patient is affected by COVID-19 or not by using CT scans.

EXISTING WORKS

Literature survey deals with the study of various research papers that aim to describe several approaches involved in lung segmentation, COVID-19 classification, and comparison of famous architectures for COVID-19 classification.

I. GPU enabled CNN Performance

[1] presents an implementation of a framework that uses GPU to accelerate the training of deep neural networks like CNNs for classification tasks. Since training CNNs can be very compute-intensive, it can take several days to train while using large datasets on the CPU. Using the proposed framework on two different networks - SimardNet and LeNet5, the authors conclude that classification on the GPU is 2 to 24 times faster than on the CPU depending on the network and GPU implementation scales very well compared to CPU version.

[2] also presents a detailed performance analysis that investigates the performance differences among seven different convolution implementations using various GPU evaluation metrics. The

authors stress on the fact that deep understanding of the algorithm, network and the hardware is important for accelerating training of deep neural networks while using GPU.

II. COVID-19 - CT Scan findings

[8] studies the time course of lung changes on chest CT of COVID-19 affected patients. 81 CT scans were obtained in the interval of 4 days from 21 confirmed COVID-19 patients. Based on the findings, it is concluded that the affected period can be split into 4 stages from symptom onset: early stage (0–4 days) where ground glass opacities can be seen in CT scans, progressive stage (5–8 days) where increased crazy paving pattern can be seen, peak stage (9–13 days) where lung involvement increased due to consolidation, and absorption stage (≥14 days) where gradual resolution of consolidation is seen without the crazy paving pattern.

III. Project Management

[7] illustrates the workflow that can be adapted to any deep learning project that requires automated medical image analysis. The checklist includes various steps in the following order: (i) defining the scope of the project, i.e., detection, segmentation, classification, monitoring, prediction or prognosis, (ii) building a team that includes radiologist, data scientist, etc., (iii) Obtain ethical approval, (iv) Data source identification and selection, (v) Data de-identification (anonymization), collection, curation and sampling which includes creation of training, validation and test datasets, (v) Model selection, training and evaluation, (vi) Choosing hardware, (vii) Regulatory, (viii) Clinical Adoption which includes deployment in clinical practice. Dataset building (data collection and curation of structured or unstructured data) is identified as one of the most time consuming parts in a project involving medical image analysis, along with model hyper-parameters fine tuning.

IV. COVID-19 Segmentation, Classification and Quantification

[9] proposes a fully automated 2 stage method to detect COVID-19 from CT images. Stage 1 involves image processing to discard CT scans in which the infection is not visible or not useful depending on count of dark pixels with a chosen threshold, followed by Stage 2 which performs classification using Feature pyramid network with deep convolutional neural network ResNet50V2. This paper uses a very large dataset of CT scans in the form DICOM images and patient's information is accessible via the DICOM files (16-bit grayscale). This problem is solved by converting the DICOM images to TIFF format, which holds the same 16-bit grayscale data and does not conclude the patients' private information.

[11] proposes a 2 stage segmentation followed by 3D classification using CT scans. In first stage, CT Images are sampled down from 512*512 to 128*128 so that the patterns of lung can be learned at a relatively low resolution. In the second stage, the bounding box was drawn with the lung field segmentation results. Finally, the 3D classification network outputs the diagnosis probability of three categories: NCP (novel coronavirus pneumonia), CP (common pneumonia) and normal controls.

[10] presents a deep learning based segmentation system that automatically quantifies infection and their volumetric ratios with respect to the lung using CT scans. The development uses human-in-the-loop workflow for manual delineation and correcting the auto-contoured results. The process is broken down into 2 stages. Stage 1: Chest CT scan is first fed into the DL-based segmentation system. Stage 2: Quantitative metrics are calculated to characterize infection regions in the CT scan, including but not limited to infection volumes and percentage of infection in the whole lung, lung lobes and broncho-pulmonary segments.

[6] proposes another fully automated method for evaluating CT scans for COVID-19 using three successively applied deep learning algorithms: (1) Pulmonary lobe segmentation and labelling using a relational two-stage U-Net architecture, (2) Lesion segmentation and CT severity score prediction using a 3D U-net using the nnU-Net framework (3) CO-RADS score prediction using 3D-inflated Inception architecture. Reference delineations of lung and lobar boundaries were automatically obtained using a commercial software (LungQ v1.1.1), followed by manual correction. Reference patterns in CT scans were identified using automatic methods by using thresholding and reviewed by a certified image analyst.

V. Comparison of Models

[3] compares 10 well known CNN architectures - AlexNet, VGG-16, VGG-19, SqueezeNet, GoogleNet, MobileNet-V2, ResNet-18, ResNet-50, ResNet-101, and Xception to distinguish infection of COVID-19 from non-COVID-19 groups. ResNet-101 and Xception showed the best performance with an AUC of 0.994.

[4] compared two types of state-of-the-art (SOTA) DL models: (1) 3D convolutional neural networks (CNNs), including DenseNet3D121, R2Plus1D, MC3 18, ResNeXt3D101, Pre- Act ResNet, and ResNet3D series, (2) 2D CNNs including DenseNet121, DenseNet201, ResNet50, ResNet101 and ResNeXt101. The performance analysis showed that 3D CNN models outperform 2D CNN models. Also, increasing the number of slices or depth of the model does not necessarily improve the model performance.

VI. Summary of Lessons learnt

- GPU computing model is very effective during CNN training where the computation is inherently parallel and involves a massive amount of floating-point operations, e.g., matrix and vector operations.
- Patient data stored in DICOM can be dismissed by converting the image to other formats or by using any commercial software for anonymizing.
- Common patterns seen in Chest HRCT scans of COVID-19 patients are
    - Ground Glass Opacity
    - Crazy paving pattern
    - Consolidation
- Segmentation of Chest HRCT images significantly improves classification accuracy.
- 3D classification is more accurate than 2D classification for detecting COVID-19.

CONTEXT OF THE RESEARCH

The input, approach and the platform to be used for the different phases of the process were selected. The platform used for building this model is Google Colab and the language preferred is Python. A total of 210 patients CT scan images were given in which 125 patients are affected by COVID-19, and 85 patients are not affected by COVID-19. Preprocessing the data refers to all the transformations on the acquired data before it is fed to the deep learning algorithm. This step involves anonymizing patient details, verifying data distribution, and sampling the required images for each patient. DICOM files of the patients usually contain Patient information like Patient ID, etc all of which were removed after anonymization using a professional DICOM editor called Sante DICOM Editor. Data Selection and Sampling was performed where only 64 slices are selected using the 3D volume formed by the patients' CT Scan before segmentation in order to prevent overfitting.

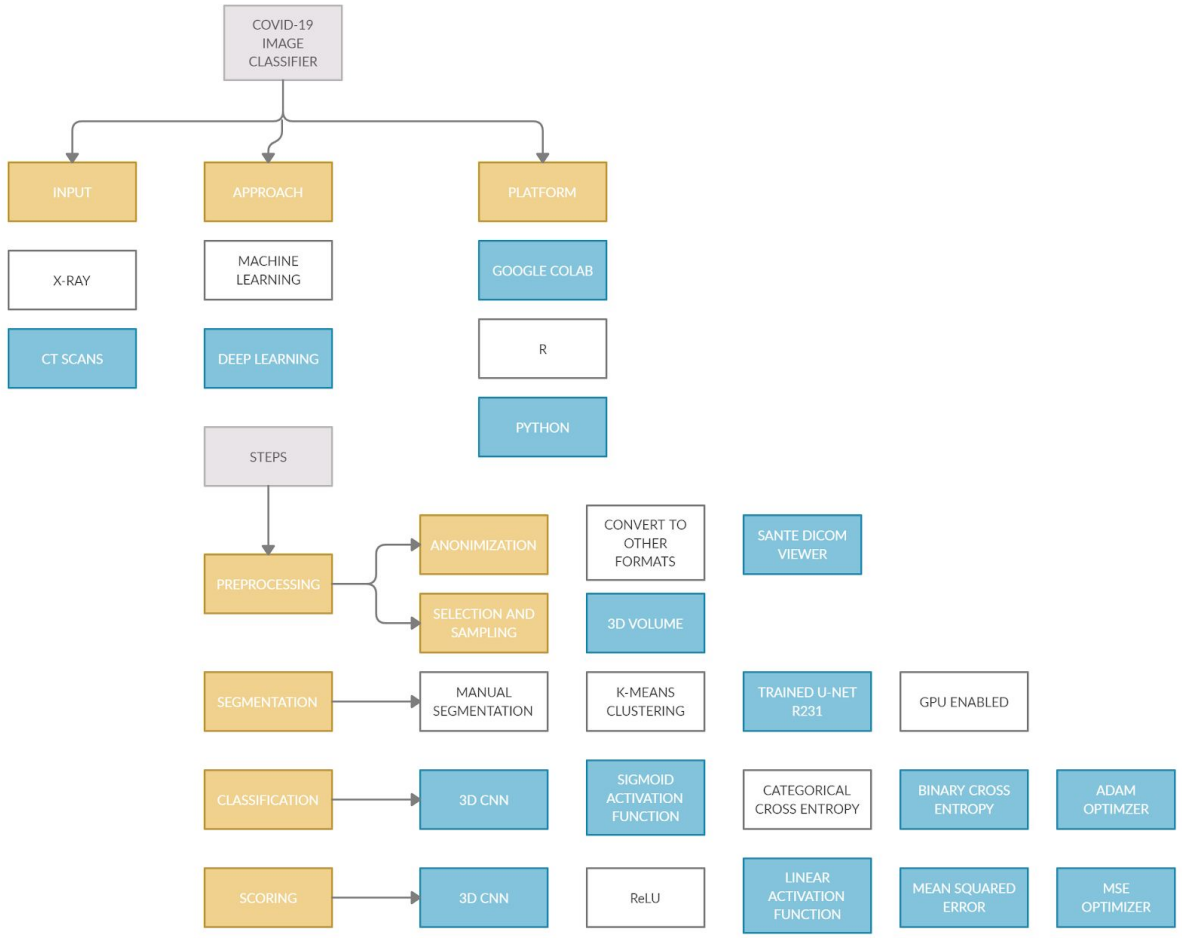

Fig. Context Diagram

Trained U-net(R231) model for lung segmentation was used for developing masks (GPU enabled). Time to develop masks was drastically reduced while using GPU when compared to CPU. A 3D convolutional neural network (CNN) is used for classification. It takes as input a sequence of 2D frames i.e. slices in a CT scan and the sigmoid activation function is used. Loss function used is Binary Cross Entropy since this is a binary classification. The model is trained with validation done at the end of each epoch with the ADAM optimizer function. A 3D convolutional neural network (CNN) is used for scoring as well. It takes as input a sequence of 2D frames i.e. slices in a CT scan and the linear activation function is used. Loss function used is Mean squared error (MSE) since this is similar to regression. The model is trained with validation done at the end of each epoch with MSE as the optimizer function.

PROPOSED WORK

The goal is to design and develop GPU enabled Convolutional Neural Network (CNN) based deep learning models to analyse Computed Tomography (CT) scans of the lung and use it to predict whether a patient is COVID-19 positive or negative and classify the severity of the disease (if applicable). The dataset is a collection of 125 COVID-19 positive and 85 COVID-19 negative patients' images provided by the Radiology department of PSG IMSR. The human ethics committee approval for the study was obtained from PSGIMSR. All images are High-Resolution DICOM images (512x512 resolution). The ratio chosen for the train to test split was 80:20. Anonymization of the DICOM

images of each patient was done using Sante DICOM Editor. All images were resized across height, width, and depth to 64 * 128 * 128 to prevent overfitting.

For analysis, only the lung region of the CT images was required, so a trained U-net(R231) model for lung segmentation was used for developing mask images. Each mask image contains only the lung segments (left and right lung). The creation of these mask images was accelerated using the GPUs available on Google Colab. The mask was then mapped on to the original image. A 3D Convolutional Neural Network (CNN) from [12] was used for classification. Its input is a sequence of 2D frames i.e. slices in a CT scan and the sigmoid activation function is used. The sigmoid activation function gives a single value in the range 0 to 1. This output denotes the probability of presence of COVID-19. The loss function used was Binary Cross Entropy since this is a binary classification. The model was trained with validation done at the end of each epoch with the ADAM optimizer function. Another 3D Convolutional Neural Network (CNN) from [12] was used for scoring. Its input is a sequence of 2D frames i.e. slices in a CT scan and the linear activation function is used. The linear activation function outputs a numerical value which can range from 0 to 100 which was then used to classify the severity of the disease as Low (0-19), Mild (20-39), Moderate (40-59), Severe (60-79) or Critical (80 and above). The loss function used was Mean squared error (MSE) since this is similar to regression. The training was performed using options like prefetch, exponential decay, and early stopping. The model was trained with validation done at the end of each epoch with MSE as the optimizer function.

Conceptual Architecture

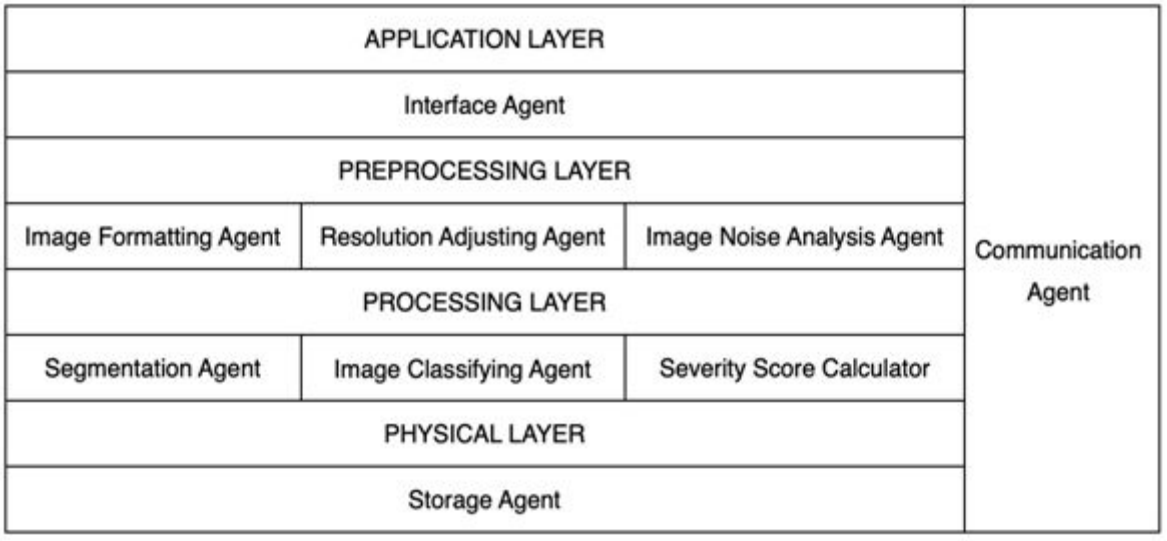

Fig. Conceptual Architecture

The conceptual design is modelled as a four-layer architecture.

Application Layer:

Interface Agent:

The Interface agent is the medium through which the medical professional will interact with the model. Through this agent, the CT images can be passed and the results of the processed images can be viewed.

Pre-Processing Layer:

Image Formatting Agent:

The Image Formatting Agent is responsible for anonymizing the DICOM files of each patient using a professional DICOM editor called Sante DICOM Editor.

Resolution Adjusting Agent:

The Resolution Adjusting Agent is responsible for downsizing the image slices and resizing those slices across height, width, and depth to 64 * 128 * 128.

Image Noise Analysis Agent:

The Image Noise Analysis Agent is responsible for checking for any noise in the images.

Processing Layer:

Segmentation Agent:

The Segmentation Agent is responsible for isolating the lungs from the given images.

Image Classifying Agent:

The Image Classifying Agent is responsible for detecting the presence of COVID-19 in the Lung images.

Severity Score Calculator:

The Severity Score Calculator is responsible for assigning a score for a COVID-19 positive patient.

Physical Layer:

Storage Agent:

Non-processed images, processed images, and results are compiled and stored using the Storage Agent.

Communication Agent:

The communication agent is common to all 4 layers and enables communications between all the four layers.

EXPERIMENTAL RESULTS

This section explains the results obtained on various steps that were performed for classifying and predicting the severity score from CT scans of the patients and performance analysis of the models on training and validation datasets. The accuracy and loss for both classification and scoring for the training and the validation sets are plotted. Since the validation set is class-balanced, accuracy provides an unbiased representation of the model's performance.

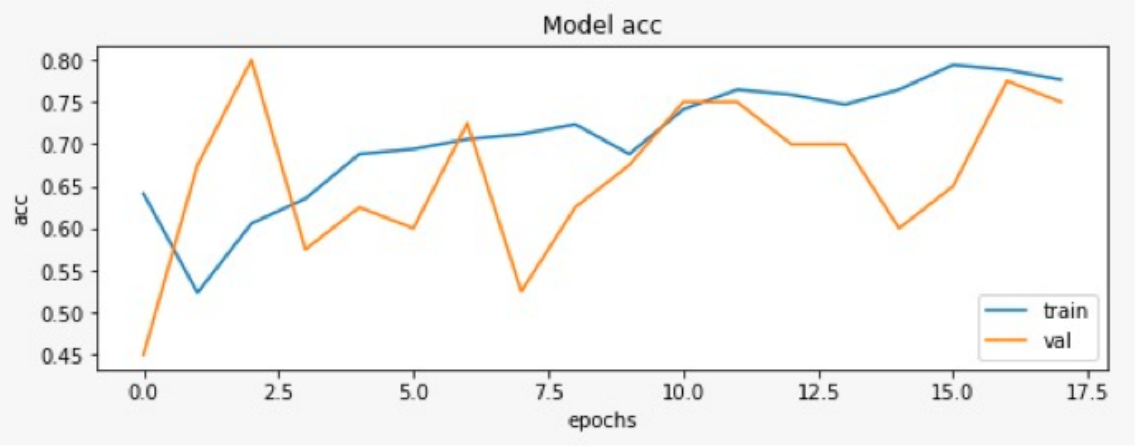

Fig. Classification Model Accuracy for Training and Validation dataset per epoch

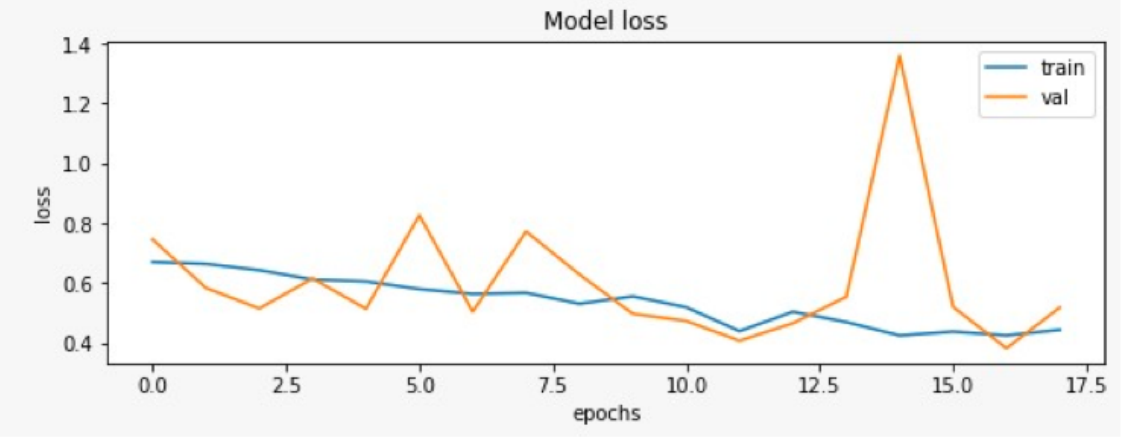

Fig. Classification Model Loss for Training and Validation dataset per epoch

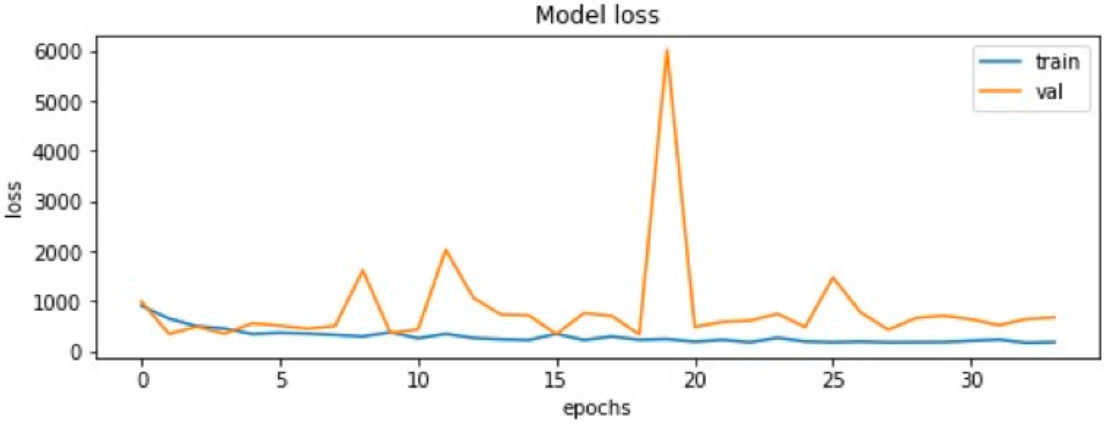

Fig. Scoring Model Loss for Training and Validation dataset per epoch

The following table shows the results of the 3D Scoring model for the test / validation dataset (which consists of a total of 40 samples) after converting the score (0 to 100) to 5 distinct classes - Low, Mild, Moderate, Severe and Critical.

Categorizing by Scoring	Number of samples	Percentage
Predicted class same as actual class	23	57.5
Predicted class differs from the actual class by 1 class	13	32.5
Predicted class differs from the actual class by 2 classes	3	7.5
Predicted class differs from the actual class by 3 classes	1	2.5
Predicted class differs from the actual class by 4 classes	0	0

Table. Output of Scoring Model for Validation dataset

The table above shows that the number of exactly classified severity is 23 out of 40 samples. However, if we consider the samples where the prediction differs from the actual class by a single class (±1), the number of correctly classified samples will be 36 out of 40 samples(23 + 13 = 36).

Hence, the accuracy (±1) of the scoring model on the test / validation dataset is 90% (57.5 + 32.5 = 90).The results of the models are tabulated below.

Model	Training Accuracy	Test Accuracy
3D CNN for Classification	78.82%	77.50%
3D CNN for Scoring	96.47%	90%

Table. Accuracy of the 3D Models

ASSESSMENT

This section describes the dataset used in this work, the various metrics used to evaluate the work along with the metric graphs.

The total data used for this work is : 210 patients CT scan images.

Training data used for this work is: 170 patients CT scan images.

Test data used for this work is: 40 patients CT scan images.

Each of these patients' CT scans contained nearly 350 to 800 slices of information.

METRICS

CONFUSION MATRIX

A confusion matrix of binary classification is a 2 * 2 table formed by counting the number of the four outcomes of a binary classifier. It is denoted as the following table.

Class		**Predicted Class**	
		Negative	Positive
Actual Class	Negative	True Negative (TN) = **12**	False Positive (FP) = **4**
	Positive	False Negative (FN) = **5**	True Positive (TP) = **19**

Table. Confusion matrix

True positives (TP) are data points which are identified as positive which are actually positive. False positives (FP) are data points which are identified as positive but which are actually negative. True negatives (TN) are data points which are identified as negative which are actually negative. False negatives (FN) are data points which are identified as negative but which are actually positive.

VISUALISING CONFUSION MATRIX

The confusion matrix for the GPU Enabled CNN Based COVID-19 HRCT Image Classifier is shown in Figure:

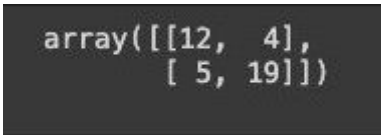

Fig. Confusion matrix

ACCURACY

Accuracy is calculated as the ratio of the number of all correct predictions to the total number of the samples in the dataset. The best accuracy is 1.0 and the worst is 0.0.

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

The accuracy of the classification model is 77.5% or **0.775**.

RECALL

Recall or Sensitivity or True Positive Rate (TPR) is the ratio of the number of correct positive predictions to the total number of positives. The best recall is 1.0 and the worst is 0.0.

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

The recall value of the classification model is **0.7916.**

PRECISION

Precision or Positive Predictive Value (PPV) is the ratio of number of correct positive predictions to the total number of positive predictions. The best precision is 1.0 and the worst is 0.0.

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

The precision value of the classification model is **0.8261.**

F1 SCORE

The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Hence F1 Score is given as follows

$$F_1 = \frac{2.Precision\,.\,Recall}{Precision + Recall}$$

The F1 score of the classification model is **0.8085**.

DISCUSSION AND CONCLUSION

ACKNOWLEDGEMENT

We sincerely thank **Dr. Maheshwaran V**, Associate Professor, Department of Radiology, PSG Institute of Medical Sciences & Research, for giving us this opportunity and valuable advice.

We are thankful to **Dr. Vinoth Kumar B**, Associate Professor, Department of Information Technology, PSG College of Technology for his technical advice.

REFERENCES

1. D. Strigl, K. Kocfler and S. Podlipnig, "Performance and Scalability of GPU-Based Convolutional Neural Networks," 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing, Pisa, 2010, pp. 317-324, https://doi.org/10.1109/PDP.2010.43

2. X. Li, G. Zhang, H. H. Huang, Z. Wang and W. Zheng, "Performance Analysis of GPU-Based Convolutional Neural Networks," 2016 45th International Conference on Parallel Processing (ICPP), Philadelphia, PA, 2016, pp. 67-76, https://doi.org/10.1109/ICPP.2016.15

3. Ardakani, A. A., Kanafi, A. R., Acharya, U. R., Khadem, N., & Mohammadi, A. (2020). “Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. Computers in Biology and Medicine”, 121, 103795. https://doi.org/10.1016/j.compbiomed.2020.103795

4. He X, Wang S, Shi S, et al. “Benchmarking Deep Learning Models and Automated Model Design for COVID-19 Detection with Chest CT Scans”, medRxiv; 2020. https://doi.org/10.1101/2020.06.08.20125963

5. Hofmanninger, J., Prayer, F., Pan, J. et al. “Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem.” Eur Radiol Exp 4, 50 (2020). https://doi.org/10.1186/s41747-020-00173-2

6. Lessmann, Nikolas & Sánchez, Clara & Beenen, Ludo & Boulogne, Luuk & Brink, Monique & Calli, Erdi & Charbonnier, Jean-Paul & Dofferhoff, Ton & Everdingen, Wouter & Gerke, Paul & Geurts, Bram & Gietema, Hester & Groeneveld, Miriam & Harten, Louis & Hendrix, Nils & Hendrix, Ward & Huisman, Henkjan & Išgum, Ivana & Jacobs, Colin & Ginneken, Bram. (2020). “Automated Assessment of CO-RADS and Chest CT Severity Scores in Patients with Suspected COVID-19 Using Artificial Intelligence”, Radiology. 202439. https://doi.org/10.1148/radiol.2020202439

7. Montagnon, E., Cerny, M., Cadrin-Chênevert, A. et al. “Deep learning workflow in radiology: a primer”, Insights Imaging 11, 22 (2020). https://doi.org/10.1186/s13244-019-0832-5

8. Pan, F., Ye, T., Sun, P., Gui, S., Liang, B., Li, L., Zheng, D., Wang, J., Hesketh, R. L., Yang, L., & Zheng, C. (2020). “Time Course of Lung Changes at Chest CT during Recovery from Coronavirus Disease 2019 (COVID-19)”. Radiology, 295(3), 715–721. https://doi.org/10.1148/radiol.2020200370

9. Rahimzadeh, M., Attar, A., & Sakhaei, S. M. (2020). “A Fully Automated Deep Learning-based Network For Detecting COVID-19 from a New And Large Lung CT Scan Dataset”, medRxiv. https://doi.org/10.1101/2020.06.08.20121541

10. Shan, F., Gao, Y., Wang, J., Shi, W., Shi, N., Han, M., Xue, Z., Shen, D., & Shi, Y. (2020). “Lung Infection Quantification of COVID-19 in CT Images with Deep Learning”, ArXiv, https://arxiv.org/abs/2003.04655

11. Zhang K, Liu X, Shen J, et al. “Clinically Applicable AI System for Accurate Diagnosis, Quantitative Measurements, and Prognosis of COVID-19 Pneumonia Using Computed Tomography”, Cell. 2020 Sep;182(5):1360. https://doi.org/10.1016/j.cell.2020.04.045

12. Zunair H., Rahman A., Mohammed N., Cohen J.P. (2020) “Uniformizing Techniques to Process CT Scans with 3D CNNs for Tuberculosis Prediction”, In: Rekik I., Adeli E., Park S.H., Valdés Hernández M..C. (eds) Predictive Intelligence in Medicine. PRIME 2020. Lecture Notes in Computer Science, vol 12329. Springer, Cham. https://doi.org/10.1007/978-3-030-59354-4_15

13. DICOM Processing and Segmentation in Python using Machine Learning method, Webpage: https://www.raddq.com/dicom-processing-segmentation-visualization-in-python/

14. Keras Documentation - Example for Three Dimensional Image Classification, Webpage: https://keras.io/examples/vision/3D_image_classification/

APPENDIX

Steps involved in building the model:

STEP 1: ANONYMIZATION

Input: The original patient scans in DICOM format along with DICOM header.

Output: Anonymised CT scans with patient details like name, age, etc., removed from the DICOM Header.

STEP 2: LUNG SEGMENTATION USING PRE-TRAINED U-NET R231 MODEL

Input: Anonymized patient scans with 600-700 slices in 512 x 512-pixel resolution

Output: Segmented scans with the shape 64 x 128 x 128

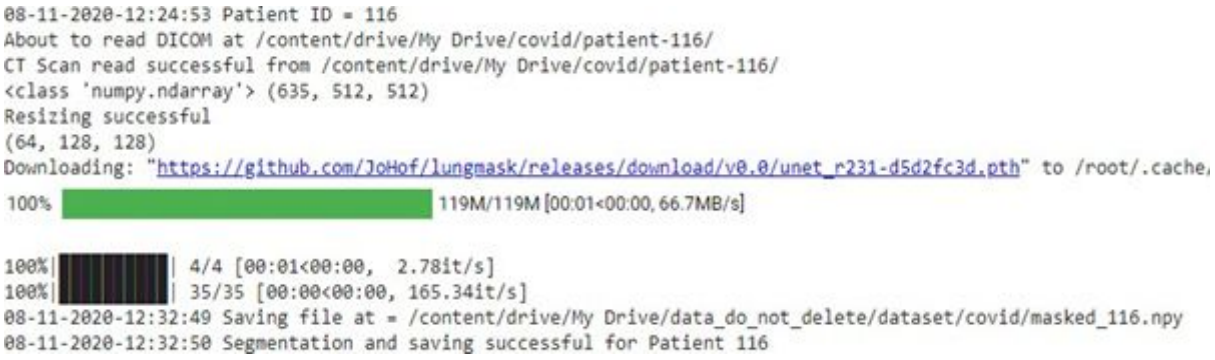

Fig. Segmentation Process

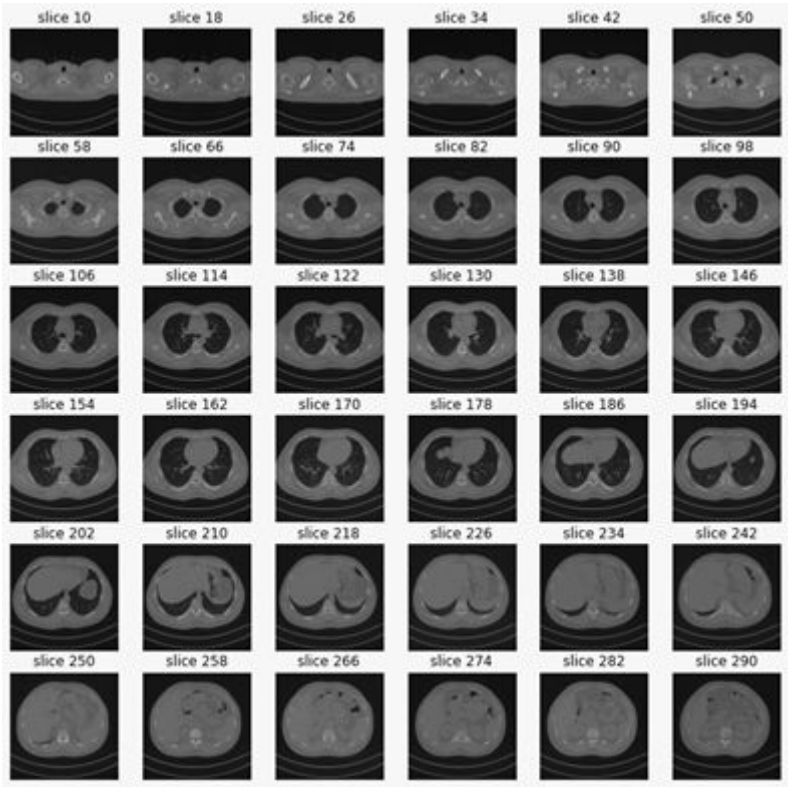

Fig. CT Slices before segmentation

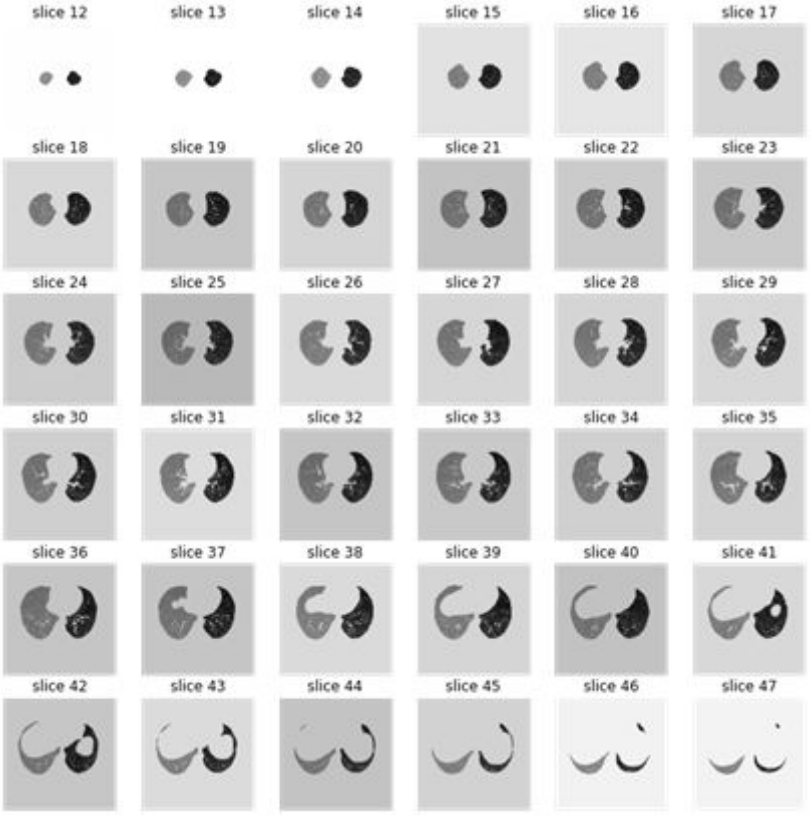

Fig. Segmented slices after applying mask

STEP 3: TRAIN 3D-CNN MODEL TO PERFORM BINARY CLASSIFICATION

Input: The original train dataset containing slices along with COVID test result (i.e positive - 1, negative - 0).

Output: Trained model after 18 epochs.

```
Epoch 1/100
WARNING:tensorflow:Callbacks method `on_train_batch_end` is slow compared to the batch time
85/85 - 14s - loss: 0.6710 - acc: 0.6412 - val_loss: 0.7463 - val_acc: 0.4500
Epoch 2/100
85/85 - 13s - loss: 0.6645 - acc: 0.5235 - val_loss: 0.5841 - val_acc: 0.6750
Epoch 3/100
85/85 - 14s - loss: 0.6432 - acc: 0.6059 - val_loss: 0.5154 - val_acc: 0.8000
Epoch 4/100
85/85 - 14s - loss: 0.6120 - acc: 0.6353 - val_loss: 0.6157 - val_acc: 0.5750
Epoch 5/100
85/85 - 14s - loss: 0.6057 - acc: 0.6882 - val_loss: 0.5143 - val_acc: 0.6250
Epoch 6/100
85/85 - 13s - loss: 0.5803 - acc: 0.6941 - val_loss: 0.8270 - val_acc: 0.6000
Epoch 7/100
85/85 - 14s - loss: 0.5641 - acc: 0.7059 - val_loss: 0.5046 - val_acc: 0.7250
Epoch 8/100
85/85 - 13s - loss: 0.5676 - acc: 0.7118 - val_loss: 0.7727 - val_acc: 0.5250
Epoch 9/100
85/85 - 14s - loss: 0.5312 - acc: 0.7235 - val_loss: 0.6290 - val_acc: 0.6250
Epoch 10/100
85/85 - 14s - loss: 0.5558 - acc: 0.6882 - val_loss: 0.4977 - val_acc: 0.6750
Epoch 11/100
85/85 - 14s - loss: 0.5195 - acc: 0.7412 - val_loss: 0.4738 - val_acc: 0.7500
Epoch 12/100
85/85 - 14s - loss: 0.4397 - acc: 0.7647 - val_loss: 0.4074 - val_acc: 0.7500
Epoch 13/100
85/85 - 13s - loss: 0.5045 - acc: 0.7588 - val_loss: 0.4661 - val_acc: 0.7000
Epoch 14/100
85/85 - 14s - loss: 0.4701 - acc: 0.7471 - val_loss: 0.5543 - val_acc: 0.7000
Epoch 15/100
85/85 - 14s - loss: 0.4249 - acc: 0.7647 - val_loss: 1.3606 - val_acc: 0.6000
Epoch 16/100
85/85 - 14s - loss: 0.4380 - acc: 0.7941 - val_loss: 0.5209 - val_acc: 0.6500
Epoch 17/100
85/85 - 14s - loss: 0.4253 - acc: 0.7882 - val_loss: 0.3824 - val_acc: 0.7750
Epoch 18/100
85/85 - 14s - loss: 0.4440 - acc: 0.7765 - val_loss: 0.5184 - val_acc: 0.7500
<tensorflow.python.keras.callbacks.History at 0x7f122e7e6ef0>
```

Fig. 3D CNN Classification Model Training of 18 epochs

STEP 4: TO SCORE THE SEVERITY OF LUNG INFECTION

Input: 64 x 128 x 128 scans which have COVID positive along with severity score

Output: Model trained to predict the severity score in the range of 1 to 100

```
Epoch 1/100
WARNING:tensorflow:Callbacks method `on_train_batch_end` is slow compared to the batch time (
85/85 - 20s - loss: 913.5384 - mse: 913.5384 - val_loss: 994.6662 - val_mse: 994.6662
Epoch 2/100
85/85 - 20s - loss: 655.0367 - mse: 655.0367 - val_loss: 350.4662 - val_mse: 350.4662
Epoch 3/100
85/85 - 19s - loss: 505.2814 - mse: 505.2814 - val_loss: 497.1805 - val_mse: 497.1805
Epoch 4/100
85/85 - 19s - loss: 459.2404 - mse: 459.2404 - val_loss: 353.0202 - val_mse: 353.0202
Epoch 5/100
85/85 - 19s - loss: 349.4583 - mse: 349.4583 - val_loss: 563.1483 - val_mse: 563.1483
Epoch 6/100
85/85 - 19s - loss: 373.1845 - mse: 373.1845 - val_loss: 515.8281 - val_mse: 515.8281
Epoch 7/100
85/85 - 19s - loss: 357.4106 - mse: 357.4106 - val_loss: 455.1203 - val_mse: 455.1203
Epoch 8/100
85/85 - 19s - loss: 330.5908 - mse: 330.5908 - val_loss: 510.1369 - val_mse: 510.1369
Epoch 9/100
85/85 - 19s - loss: 299.9180 - mse: 299.9180 - val_loss: 1621.2931 - val_mse: 1621.2931
Epoch 10/100
85/85 - 19s - loss: 383.1299 - mse: 383.1299 - val_loss: 368.1598 - val_mse: 368.1598
Epoch 11/100
85/85 - 19s - loss: 264.8126 - mse: 264.8126 - val_loss: 443.3730 - val_mse: 443.3730
Epoch 12/100
85/85 - 19s - loss: 350.0805 - mse: 350.0805 - val_loss: 2032.7949 - val_mse: 2032.7949
Epoch 13/100
85/85 - 19s - loss: 269.6869 - mse: 269.6869 - val_loss: 1070.5349 - val_mse: 1070.5349
Epoch 14/100
85/85 - 19s - loss: 243.5412 - mse: 243.5412 - val_loss: 737.8983 - val_mse: 737.8983
Epoch 15/100
85/85 - 19s - loss: 228.4870 - mse: 228.4870 - val_loss: 723.9706 - val_mse: 723.9706
Epoch 16/100
```

Fig. 3D CNN Scoring Model Training of first 15 of 34 epochs

STEP 5: TESTING OF SEVERITY SCORE

Input: Single Patient's CT scan

Output: Predicted Severity Score Percentage

```
Affected Score Percentage = 43.03343 %
Predicted CTSI =  10.0

Actual Score Percentage = 52 %
Actual CTSI =  13
```

Fig. Test scan and severity score