

A Deep Learning Framework for Pulmonary Disease Classification Using Volume-Rendered CTs

Supplementary Material

September 9, 2025

1 Expanded Related Work

We complement in this section the related work to our proposal that uses CT images as inputs to COVID-19 classification models [1, 2, 3, 4, 5, 6, 7, 8, 9]. DeCoVNet [2] uses a weakly-supervised 3D deep convolutional network to predict the probability of COVID-19 following a binary classification approach. In the first step, the authors used a U-NET to segment the lung in the volume and create a 3D binary mask. The CT and its 3D lung mask are sent to DeCoVNet, which consists of three stages: 3D convolution, 3D residual blocks, and a progressive classifier. Ternary models soon followed binary models to allow the separation of COVID-19 from normal cases and other types of CAPs. An example is COVNet [3], which uses a 3D deep learning framework that takes 3D slices as input, generates features, combines them using a max-pooling operation, and finally generates a probability score for each class. Like DeCoVNet, COVNet also relies on a pre-processing step that uses a U-NET to create a segmented mask for the lung. Amyar et al. [7] proposes a ternary model that uses a shared encoder for the classification and lesion segmentation tasks, without requiring labeled segmentation data. Deep-chest [8] uses a VGG-CNN model that supports classification in four classes: COVID-19, pneumonia, lung cancer and normal cases. Serte and Demirel [9] proposed a binary 3D classification model that fuses image level predictions to classify CT volumes.

The ICASSP 2021 competition brought an opportunity to compare methods by providing a public dataset of CTs (COVID-CT-MD [10]) and a contest of performing classification in three possible classes: normal, COVID-19, or CAP. The six-best solutions were presented at the ICASSP conference [11, 12, 13, 14, 15, 16]. The first place (team TheSaviours [11]) has an accuracy of 90%. They describe a two-stage CNN model for detecting COVID-19 and Community-Acquired Pneumonia (CAP). The first stage is designed to detect infections (COVID-19 or CAP). The second stage performs the classification in three classes (CAP, COVID-19, and Normal). The second place (team IITDelhi [12]) has an accuracy of 88.89%. Their method follows a three-level approach. In the first level, they use a slice-level classifier that performs feature extraction from all the slices of the CT to learn different sizes of infection. The next level performs a patient-level classifier, using four classifiers to distinguish between infected and normal slices. The last level uses an ensemble-learning that combines the scores of the previous level classifiers. The third place (team LLSCP [13]) has an accuracy of 87.78%. They use a multi-stage progressive learning approach composed of a 3D Resnet module, an ensemble binary classifier, and a final combining stage. The remaining entries achieved 85.56% [14], 81.11% [15] and 80% [16].

For comparison purposes, we chose TheSaviours [11] (ICASSP 2021 winner), DeCoVNet [2], and COVNet [3] approaches due to their outstanding performance and the public availability of code or pre-trained models.

2 Supplementary Results

In this section, we present and discuss additional results obtained both during the development of our pipeline and through its application to public and private datasets. For reference, an overview of our end-to-end pipeline for CT image analysis is shown in Figure 1. The following sections provide a more detailed discussion of the experiments related to the selection of views, transfer functions, and backbone architectures. We then present in greater detail our results for COVID-19 classification, addressing both binary and ternary classification tasks.

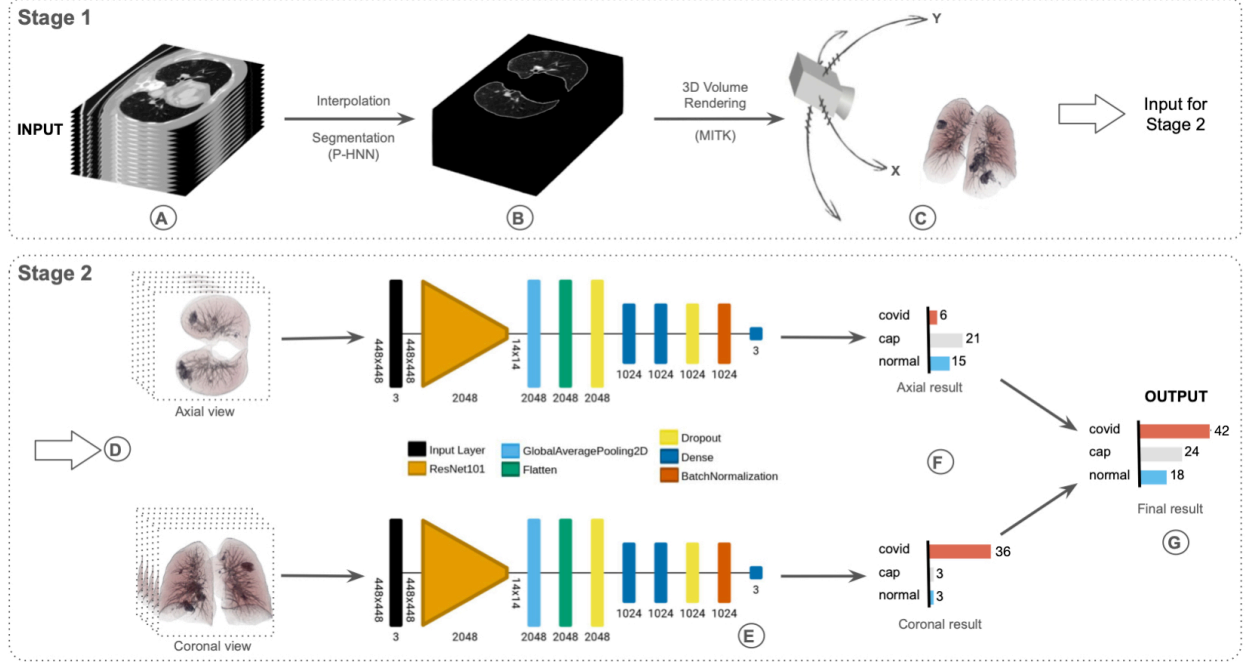


Figure 1: Two-stage CT-VR pipeline. Stage 1 prepares data to obtain the input images for model development. In Stage 2, DL models are trained to distinguish among the classes of interest (in this example COVID-19), and their outputs are combined to obtain a final patient-level classification.

2.1 Selection of Viewing Camera Positions

For the development of our work, volumetric visualization was used solely as a visual aid. By adjusting the transfer functions (TFs), we were able to highlight different lung regions and areas of interest, which became more prominent as the TF varied. Using the scripts we developed—based on the MITK project—we could freely position a camera around the lung, enabling screenshots and video captures from any desired viewpoint.

To simplify the setup, we focused on six representative views, chosen for their similarity to those commonly used in medicine: Axial, Coronal, and Sagittal. Specifically, we captured the Axial view from the top (Figure 2.1) and bottom (Figure 2.3), the Coronal view from the front and back (Figure 2.2 and 2.4), and the Sagittal view from the right and left sides (Figure 2.5 and 2.6).

Since our approach trains a separate model for each view, employing all six views required training six networks per experiment, which was computationally costly. Moreover, Sagittal views consistently underperformed compared to the others, likely due to the overlap of both lungs in those perspectives (Figures 2.5 and 2.6). Even when reduced to four networks, training remained

time-consuming. Performance metrics showed that using all four views yielded nearly the same results as training with just two—one Axial and one Coronal. We tested configurations with three views (two Axial and one Coronal, and vice versa), but results were biased by the duplicated perspective. Therefore, we ultimately selected only the Top Axial and Front Coronal views for subsequent experiments.

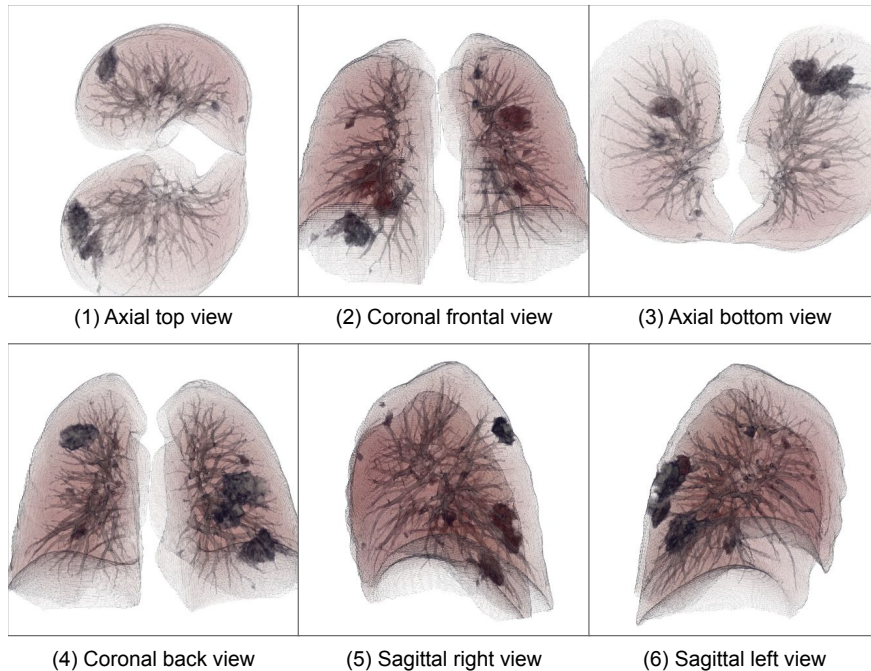


Figure 2: All six views of the lungs proposed by our approach CT-VR to analyze.

Another important consideration was the number of images (camera shots) to generate per view. Since images could be captured from any angle, the question became how many were ideal. Without a predefined metric, we tested sets of 168, 84, 42, and 21 images obtained with sweeping angles of 9° to 15° . Angles larger than 15° caused visible distortions, making it difficult to preserve Axial or Coronal characteristics (Figure 3). Final selection was guided by network performance: 84 and 42 images per view produced the best results, with minimal difference between them. To reduce training time, we chose 42 images per view as the standard configuration.

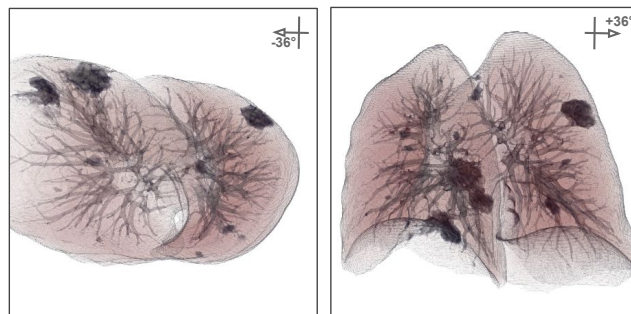


Figure 3: Image rendered with -36° (horizontal axis) in axial top view (left) and from $+36^\circ$ (horizontal axis) in coronal front view (right).

2.2 Selection of the Transfer Function

Volume rendering relies on transfer functions (TFs) that map Hounsfield unit (HU) values to color, transparency, and opacity, enabling the visualization of internal lung structures. Based on prior studies [17, 18], lung regions typically correspond to around -700 HU, while ground-glass opacities (GGOs) fall between -700 and -300 HU—guiding TF design in this study.

We used the MITK framework [19] to test different transfer functions (TFs) for generating volume-rendered images from the 3D segmented lungs. MITK provides a user interface that enables customization of TFs by adjusting opacity and color values across ranges of HU values, see Figure 4.

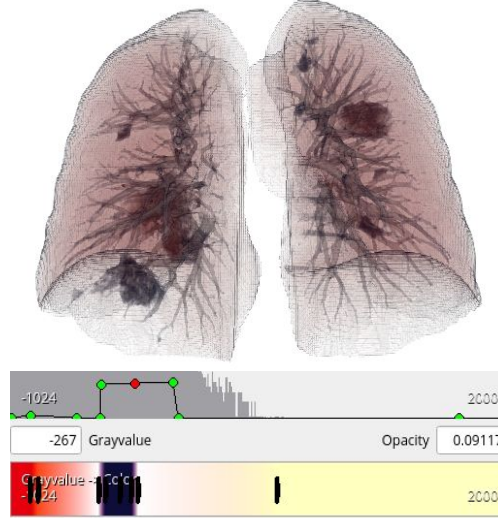


Figure 4: MITK interface: on top, the rendering window. Below, widgets to manually adjust opacity and color values.

We used this interface to select the most effective transfer function (TF) for generating volume-rendered images, compared using the ternary classification model on the public dataset. We explored over 60 TFs that were tested on segmented lung volumes. Six representative TFs (TF1–TF6), shown in Figure 5, illustrate how different mappings affect image outcomes.

Although they were all applied to the same CT exam, the rendered image differs significantly. TF1 highlights the outer layer of the lungs but misses internal structures. TF2 to TF6 reveal internal lesions in different ways. TF2 is defined within the lower range of values in the lung and ground-glass opacity (*i.e.*, [-700, -300]), mapping only HU values in this interval. This function conserves a particular spatial distribution of the ground-glass opacity present in the lung but misses details on the lung texture and regions without any lesions. TF3 aims to replicate the behavior of 2D CTs that vary in a single color scale (usually gray-scale), mapping original values in the range of [-750, -200] with variations applied to the brightness. TF3 manages to preserve the lung as a three-dimensional image without losing critical information on the internal lesions caused by COVID-19, such as ground-glass opacity. TF4 to TF6 are obtained by including more colors to help differentiate the lung textures and features of the different classes. TF4 shows a thin delineation of the lung inner layer and a prominent highlighting of the bronchi, which in the presence of COVID-19 may show changes such as bronchial wall thickening [20]. For TF5 and TF6, we observe a more precise differentiation of ground-glass opacities from other characteristics, which may be interesting for the classification model as this feature is the most common imaging finding in COVID-19 patients [20]. Thus, by exploring a wide range of mapping combinations for color and opacity, each TF

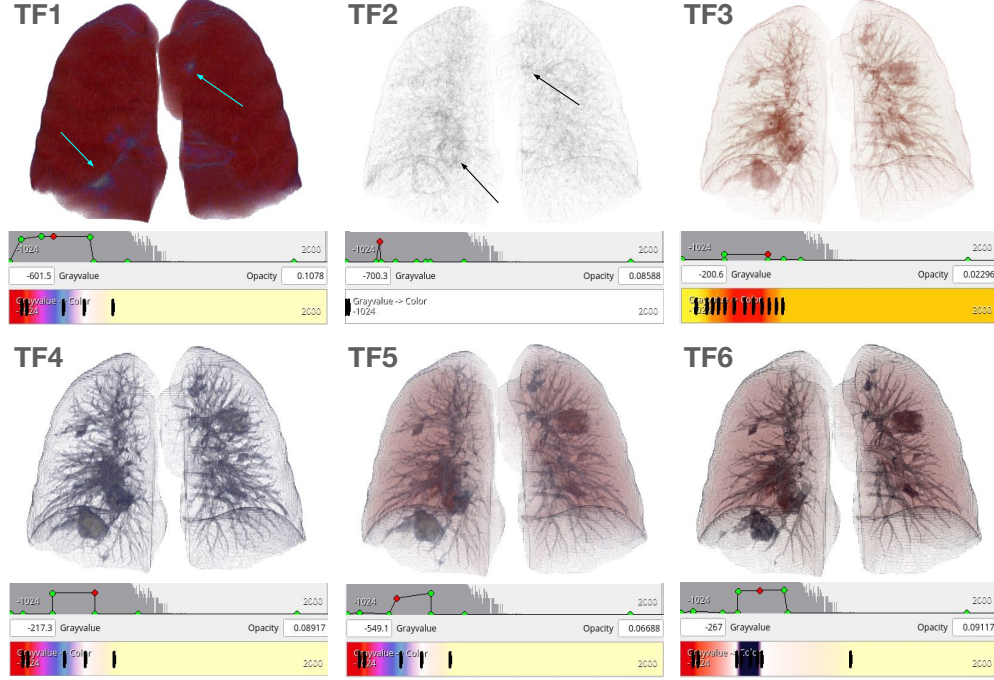


Figure 5: Volume rendering of the lung using customized transfer functions. TF1 shows the boundary of the lungs but misses internal structures. TF2-TF4 shows internal structures but has similar colors for both GGOs (lesions) and the lung. TF5-TF6 better separates lungs and lesions.

can highlight different regions or features of interest when applied to CT scans, such as external surfaces of the lungs (TF1), arteries (TF3), and ground-glass opacity (TF5 and TF6).

Figure 6 compares the ROC curves obtained for distinct TFs. Although there is a small difference between the AUC score for TF6 and TF2, the ROC curve for TF6 achieves the highest sensitivity for a 10% false-positive rate. Therefore, we chose TF6 as the standard transfer function. Besides the visual detection of different features highlighted by each TF, this test proved relevant, as switching TF while keeping the model architecture results in considerable accuracy variation. Comparative model training and validation metrics are given in Table 1.

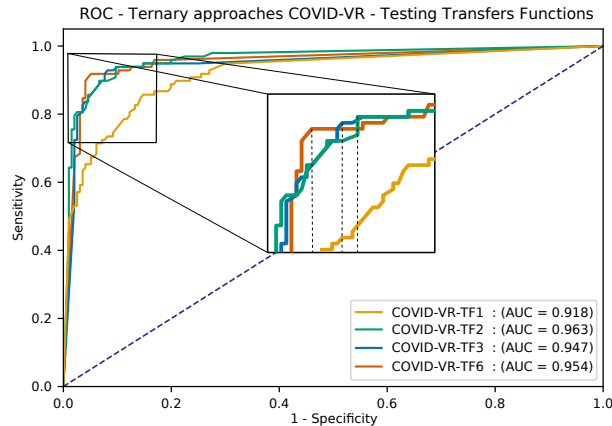


Figure 6: ROC curves for transfer functions TF1, TF2, TF3, and TF6

Table 1: Comparison of distinct transfer functions in CT-VR. Model training and validation use the train and validation sets from the COVID-CT-MD public dataset for the ternary classification task (COVID-19 vs. CAP vs. Normal).

	Metrics	TF1	TF2	TF3	TF6
Overall	Acc	79.6%	85.7%	87.8%	90.8%
	F1	78.7%	84.5%	87.6%	90.8%
	AUC	91.8	96.3	94.7	95.4
COVID-19	Sens	89.1%	96.4%	89.1%	89.1%
	Spec	72.1%	72.1%	88.4%	93.0%
	Prec	80.3%	81.5%	90.7%	94.2%
	F1	84.5%	88.3%	89.9%	91.6%

Another decision involved the choice of image background. We tested two colors, white and black (Figure 7). By changing the background while applying the same transfer function, different lung regions could be emphasized. In experiments with the COVID-CT-MD dataset, changing the background from black to white using transfer function TF6 improved ternary classification accuracy by 3% compared to models trained on black-background images.

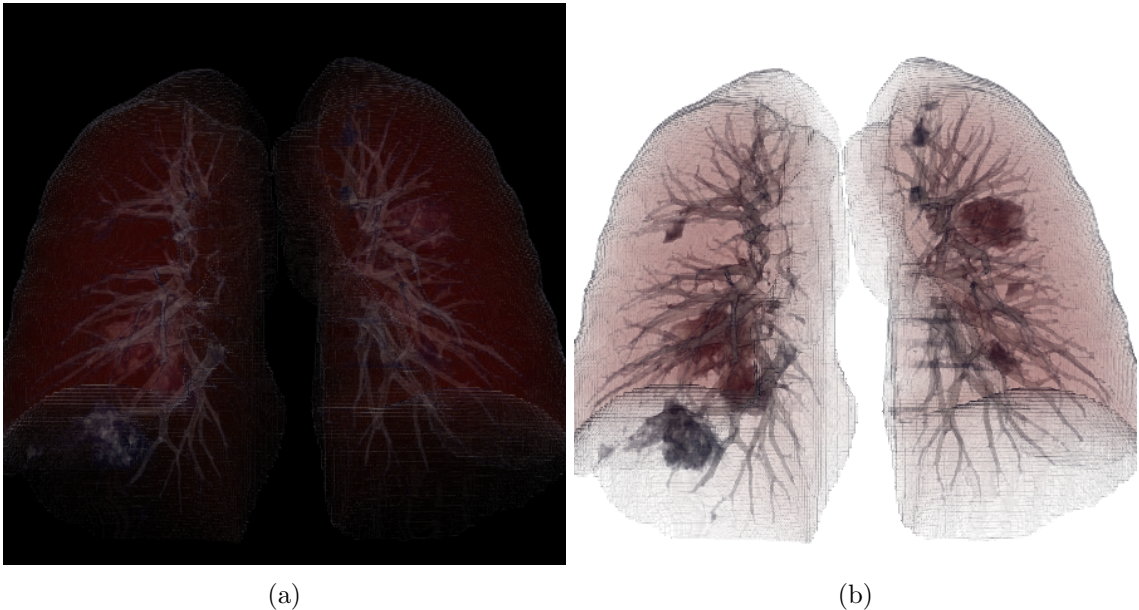


Figure 7: Same transfer function with different background colors. It should be noted how the lesions in the white background are easily seen in contrast with a black background.

Finally, similar to the step-by-step procedure reported by TheSaviours, where only the central 80 slices are used for classification, we applied an analogous strategy in our approach. This resulted in images excluding the top and bottom of the lungs, as illustrated in Figure 8, altering the axial and top views. The intention was to evaluate whether removing these extremities could provide additional information. For instance, excluding borders in the coronal view prevents the network from focusing on peripheral features, while removing the top of the lungs in the axial view shifts attention from outer to inner regions. However, this additional step did not improve the performance of the trained networks and was therefore omitted from the final model.

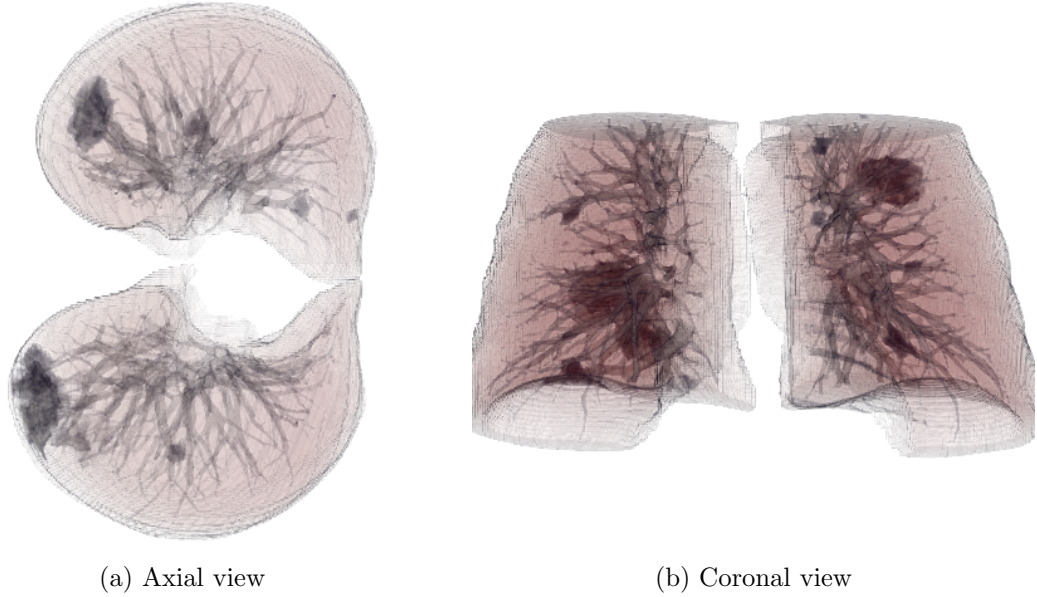


Figure 8: Volume rendering from 80 center slices for COVID-CT-MD.

2.3 Selection of the Backbone Network

We tested several CNN architectures as the backbone network to perform CT scan classification with the models described in the paper, including ResNet [21], DenseNet [22], VGG [23], and EfficientNet [24] families. We first experimented with VGG16 and VGG19, where VGG served as the backbone followed by additional layers. From these added layers, we included a 20% Dropout (having also tested 10%, 15%, and 25%, with 20% yielding the best results). We also considered whether to use Batch Normalization together with Dropout. Experiments without Batch Normalization failed to reach baseline performance, confirming the need to retain both Dropout and Batch Normalization layers. For the Dense layers following the first Dropout, we tested configurations of 2048×2048 and 2048×1024 , but results showed no meaningful improvement. Therefore, we adopted the 1024×1024 configuration. Optimization was performed using Adam with learning rates in the set 1×10^{-3} , 1×10^{-4} , 1×10^{-5} , 2×10^{-5} , 5×10^{-4} , 5×10^{-5} , with the best results obtained at 2×10^{-5} , which we used in subsequent experiments.

We also tested an architecture with multiple inputs (one per selected view) that produced a single unified output. In this setup, the network received three images per view, each processed through a subnetwork (one per view) similar to that in Figure 1, and the outputs were combined via a smaller dense network. Although implemented with VGG16 as the backbone, this approach did not yield significant performance gains. Given the larger model size and increased training time, we concluded that this architecture was not advantageous.

We evaluated different input image sizes: 512×512 , 448×448 , 384×384 , 320×320 , 256×256 , 224×224 pixels. The best results were achieved with 512px and 448px inputs. Since 448px reduced the model size and training time while maintaining performance comparable to 512px, we adopted 448×448 as the standard input resolution.

We explored different depths for each architecture (*i.e.*, ResNet50, ResNet101, DenseNet121, DenseNet201, EfficientNet-B0, EfficientNet-B1, EfficientNet-B6, VGG16, and VGG19) with fixed training and validation sets contained in the COVID-CT-MD dataset. A comparison of the classification performance of the better models for each network family is given in Table 2.

Table 2: Comparison of distinct backbone network architectures in CT-VR. Model training and validation use the train and validation sets from the COVID-CT-MD public dataset for the ternary classification task (COVID-19 vs. CAP vs. Normal).

	Metrics	VGG16	DenseNet121	EfficientNet-B2	ResNet101
Overall	Acc	88.8%	87.8%	86.7%	90.8%
	F1	88.6%	87.7%	86.7%	90.8%
	AUC	95.6	96.5	95.1	95.4
COVID-19	Sens	89.1%	85.5%	83.6%	89.1%
	Spec	88.4%	93.0%	90.7%	93.0%
	Prec	90.7%	94.0%	92.0%	94.2%
	F1	89.9%	89.5%	87.6%	91.6%

2.4 Classification Performance for the Public Dataset

We conducted experiments to evaluate CT-VR for the ternary classification model (COVID-19 vs. CAP vs. Normal) in the validation set and the three test sets from the COVID-CT-MD dataset (here unified in a single test set). All models use the ResNet101 backbone network and TF6 for volume rendering. We compared CT-VR against the winner of the competition (TheSaviours) [11], and two state-of-the-art methods, DeCoVNet [2] and COVNet [3].

Instead of directly averaging softmax probabilities, our approach sums the class-wise votes from each submodel (one trained on axial views and one on coronal views) to form a patient-level consensus distribution. This voting-based aggregation is illustrated in Figure Figure 1.F–G. While simple, this method preserves interpretability in the decision process, avoids overfitting and unnecessary complexity in the aggregation step, and aligns with established ensemble learning principles, where majority voting or similar schemes are widely adopted for robust decision-making.

We use a fixed transfer function (TF6) to generate volume rendering images in all experiments. The performance results correspond to the ternary classification models using the train and validation sets from the public dataset (*i.e.*, COVID-CT-MD dataset). We show the performance for the COVID-19 class and the overall performance for the three classes. F1 and AUC scores are obtained with the micro-average. Additional results per class can be found in Table 3.

Table 3: Ternary classification results by class in the Public COVID-CT-MD. Performance metrics are in percentages.

		General		Covid-19				By Classes				CAP				Mean				
		Accu.	Kappa	Preci.	Sensi.	Speci	F-score	Preci.	Sensi.	Speci.	F-score	Preci.	Sensi.	Speci.	F-score	Weighted (AUC Micro)				
Transfer Function Comparison	TF 1	79.6	63.7	80.3	89.1	72.1	84.5	71.4	83.3	89.2	76.9	100.0	47.4	100.0	64.3	81.9	79.6	81.7	78.7	91.8
	TF 2	85.7	74.0	81.5	96.4	72.1	88.3	95.7	91.7	98.6	93.6	90.0	47.4	98.7	62.1	86.6	85.7	83.7	84.5	96.3
	TF 3	87.8	79.2	90.7	89.1	88.4	89.9	85.2	95.8	94.6	90.2	82.4	73.7	96.2	77.8	87.7	87.8	91.4	87.6	94.7
	TF 4	88.8	80.8	90.9	90.9	88.4	90.9	85.2	95.8	94.6	90.2	87.5	73.7	97.5	80.0	88.8	88.8	91.7	88.6	95.2
	TF 5	85.7	75.4	87.5	89.1	83.7	88.3	82.1	95.8	93.2	88.5	85.7	63.2	97.5	72.7	85.8	85.7	88.7	85.3	95.0
	TF 6	90.8	84.6	94.2	89.1	93.0	91.6	85.7	100.0	94.6	92.3	88.9	84.2	97.5	86.5	91.1	90.8	94.3	90.8	95.4
Architecture Comparison	VGG16	88.8	81.0	90.7	89.1	88.4	89.9	88.9	100.0	95.9	94.1	82.4	73.7	96.2	77.8	88.6	88.8	91.7	88.6	95.6
	DenseNet121	87.8	79.7	94.0	85.5	93.0	89.5	82.8	100.0	93.2	90.6	78.9	78.9	94.9	78.9	88.3	87.8	93.4	87.6	96.5
	EfficientNetB2	86.7	78.0	92.0	83.6	90.7	87.6	85.7	100.0	94.6	92.3	75.0	78.9	93.7	76.9	87.2	86.7	92.2	86.7	95.1
	ResNet101	90.8	84.6	94.2	89.1	93.0	91.6	85.7	100.0	94.6	92.3	88.9	84.2	97.5	86.5	91.1	90.8	94.3	90.8	95.4
Method Comparison (Train / Validation)	DeCoVNet	67.3	44.9	74.5	74.5	67.4	74.5	71.4	41.7	94.6	52.6	51.7	78.9	82.3	62.5	69.3	67.3	76.9	66.8	78.2
	COVNet	77.6	63.2	85.4	74.5	83.7	79.6	66.7	83.3	86.5	74.1	75.0	78.9	93.7	76.9	78.8	77.5	86.3	77.7	89.8
	TheSaviours	74.5	58.8	86.7	70.9	86.0	78.0	51.4	75.0	77.0	61.0	88.9	84.2	97.5	86.5	78.5	74.5	86.0	75.5	88.4
	CT-VR	90.8	84.6	94.2	89.1	93.0	91.6	85.7	100.0	94.6	92.3	88.9	84.2	97.5	86.5	91.1	90.8	94.3	90.8	95.4
Method Comparison (Test)	DeCoVNet	52.2	30.3	48.8	57.1	61.8	52.6	77.8	20.0	96.4	31.8	50.0	100.0	71.4	66.7	60.3	52.2	77.4	47.6	74.4
	COVNet	67.8	50.0	60.0	77.1	67.3	67.5	74.1	57.1	87.3	64.5	77.8	70.0	94.3	73.7	69.4	67.7	81.1	67.7	86.1
	TheSaviours	90.0	84.6	90.9	85.7	94.5	88.2	89.2	94.3	97.1	90.0	90.0	90.0	97.1	90.0	90.0	90.0	94.4	90.0	97.4
	CT-VR	86.7	79.7	89.2	94.3	92.7	91.7	96.4	77.1	98.2	85.7	72.0	90.0	90.0	80.0	88.2	86.7	94.2	86.8	95.7

CT-VR achieved the highest overall performance in the validation set. CT-VR accuracy was

90.8% in contrast to 77.6% obtained with COVNet, which ranked second. In terms of AUC score, CT-VR achieved 95.4 while COVNet achieved 89.8. CT-VR also had the best predictive power for COVID-19 cases, reaching high and balanced sensitivity and specificity values. When comparing the CAP and Normal cases, CT-VR had a 100% sensitivity and 94% specificity for the Normal class and 84.2% sensitivity and 97.5% specificity for the CAP class - in both cases surpassing the competing strategies.

CT-VR achieved an accuracy of 86.7% in the test set, while TheSaviours correctly classified 90.0% of the test instances. CT-VR was the best method for detecting COVID-19 cases (94.3% sensitivity), keeping high specificity (92.7%), and F1-score (91.7%). In contrast, TheSaviours achieved the highest F1-score for Normal and CAP cases. The CT-VR model is trained with labels at the patient level, while the approach presented by TheSaviours [11] trains a model by exploring labels at the slice level. We reach similar results despite using a coarser-grained annotation in CT scans.

Figure 9 compares the micro-average ROC curves for the ternary classification models using the validation test (top) and the test set (bottom). CT-VR has the best performance in the validation set, notably improving the true positive rate (*i.e.*, sensitivity) for false-positive rates ranging from 0 to 0.3. Considering the test set, the performance of the TheSaviours model improves in experiments with the validation set and surpasses CT-VR (*i.e.*, 97.4 vs. 95.7) in the AUC score. Nonetheless, we highlight that CT-VR had the most stable performance between validation and test sets, despite the clinical and technical differences introduced in the CT images from the SPGC-COVID Test Set [25]. Finally, we note that CT-VR had an accuracy close to that reported by the first place of the competition like IITDelhi [12] with 88.9%, LLSCP [13] with 87.8%, and UniSheff_EEE [14] with 85.56%. The results for these approaches were not included in the table due to the lack of public code to reproduce the experiments.

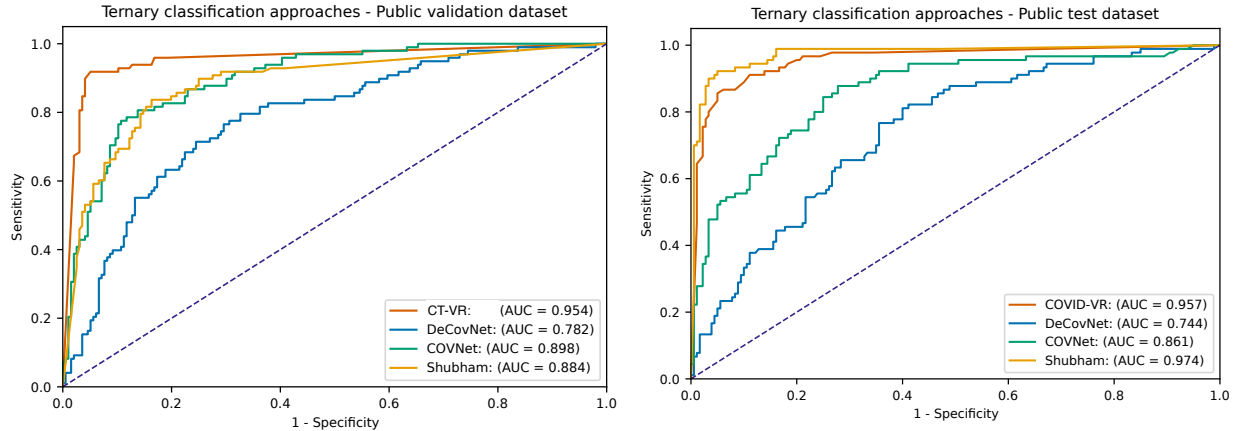


Figure 9: Micro-average ROC curves for the ternary classification task using the public dataset: validation set (left) and test set (right).

2.5 Classification Performance for the Private Dataset

We analyzed the performance of the methods for a binary classification task, COVID-19 vs. non-COVID-19, using the private datasets. The experiments considered two definitions for the negative class: in the first, we merged the Negative, Indeterminate, and Atypical classes into a unique non-COVID-19 class, and in the second, we only considered the original negative class (*i.e.*, Negative for pneumonia) as the classifiers non-COVID-19 class. In both cases, the Typical classification

was considered the positive class (*i.e.*, COVID-19). Performance assessment was based on a 5-fold CV, with the configuration of the same folds for CT-VR, DeCoVNet, and COVNet. CT-VR obtained the best results in all metrics, with 92.2% of accuracy and 95.6% of AUC for this binary classification task (Table 4).

In comparing COVID-19 vs. Normal classification, CT-VR obtained accuracy and F1-score similar to COVNet, but with higher Sensitivity and AUC scores (Table 5). The ROC curves are given in Figure 10, that shows that in both cases CT-VR has the best AUC scores.

Table 4: COVID-19 vs. Others task results. Training and validation in the private dataset

Metrics	CT-VR	DeCovNet	COVNet
Acc	92.2%	87.8%	89.4%
Sens	83.6%	78.7%	83.6%
Spec	96.2%	92.0%	92.0%
F1	87.2%	80.3%	83.3%
AUC	95.6	89.2	93.1

Table 5: COVID-19 vs. Normal task results. Training and validation in the private dataset

Metrics	CT-VR	DeCovNet	COVNet
Acc	96.1%	92.5%	96.1%
Sens	96.7%	91.0%	95.9%
Spec	95.3%	94.3%	96.2%
F1	96.3%	92.9%	96.3%
AUC	98.6	96.3	97.7

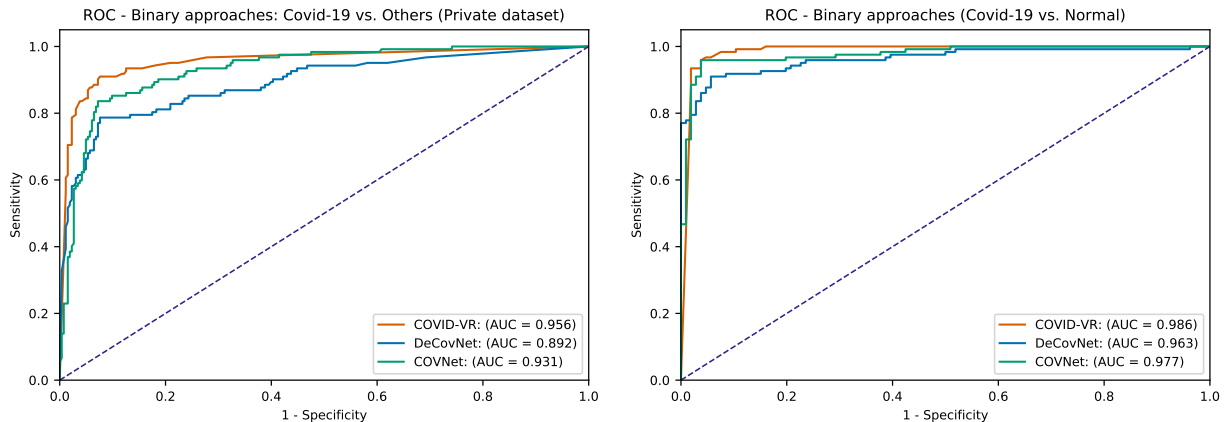


Figure 10: ROC curves for the COVID-19 against the class others (left) and normal (right) in the private dataset.

3 Public Repository

The repository for the source code and additional supplementary information are publicly available at GitHub: <https://github.com/joacomba/ct-vr>.

References

- [1] C. Jin, W. Chen, Y. Cao, Z. Xu, Z. Tan, X. Zhang, L. Deng, C. Zheng, J. Zhou, H. Shi, and J. Feng, “Development and evaluation of an artificial intelligence system for COVID-19 diagnosis,” *Nature Communications*, vol. 11, no. 1, 2020.
- [2] X. Wang, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, W. Liu, and C. Zheng, “A Weakly-Supervised Framework for COVID-19 Classification and Lesion Localization From Chest CT,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2615–2625, 2020.
- [3] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang, Q. Song, K. Cao, D. Liu, G. Wang, Q. Xu, X. Fang, S. Zhang, J. Xia, and J. Xia, “Using Artificial Intelligence to Detect COVID-19 and Community-acquired Pneumonia Based on Pulmonary CT: Evaluation of the Diagnostic Accuracy,” *Radiology*, vol. 296, pp. E65–E71, 2020.
- [4] 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.
- [5] L. Wang, Z. Q. Lin, and A. Wong, “COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images,” *Scientific Reports*, vol. 10, 12 2020.
- [6] O. Gozes, M. Frid-Adar, H. Greenspan, P. D. Browning, H. Zhang, W. Ji, A. Bernheim, and E. Siegel, “Rapid AI development cycle for the coronavirus (COVID-19) pandemic: Initial results for automated detection & patient monitoring using deep learning CT image analysis,” 2020.
- [7] A. Amyar, R. Modzelewski, H. Li, and S. Ruan, “Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: Classification and segmentation,” *Computers in Biology and Medicine*, vol. 126, p. 104037, 2020.
- [8] D. M. Ibrahim, N. M. Elshennawy, and A. M. Sarhan, “Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases,” *Computers in Biology and Medicine*, vol. 132, p. 104348, 2021.
- [9] S. Serte and H. Demirel, “Deep learning for diagnosis of COVID-19 using 3D CT scans,” *Computers in Biology and Medicine*, vol. 132, p. 104306, 2021.
- [10] P. Afshar, S. Heidarian, N. Enshaei, F. Naderkhani, M. J. Rafiee, A. Oikonomou, F. B. Fard, K. Samimi, K. N. Plataniotis, and A. Mohammadi, “COVID-CT-MD, COVID-19 computed tomography scan dataset applicable in machine learning and deep learning,” *Scientific Data*, vol. 8, 12 2021.
- [11] S. Chaudhary, S. Sadbhawna, V. Jakhetiya, B. N. Subudhi, U. Baid, and S. C. Guntuku, “Detecting COVID-19 and Community Acquired Pneumonia Using Chest CT Scan Images With Deep Learning,” in *ICASSP 2021 - IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 8583–8587.
- [12] P. Garg, R. Ranjan, K. Upadhyay, M. Agrawal, and D. Deepak, “Multi-Scale Residual Network for COVID-19 Diagnosis Using CT-Scans,” in *ICASSP 2021 - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8558–8562.

- [13] Z. Yang, Y. Hou, Z. Chen, L. Zhang, and J. Chen, "A Multi-Stage Progressive Learning Strategy for Covid-19 Diagnosis Using Chest Computed Tomography with Imbalanced Data," in *ICASSP 2021 - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8578–8582.
- [14] S. Xue and C. Abhayaratne, "Covid-19 Diagnostic Using 3d Deep Transfer Learning for Classification of Volumetric Computerised Tomography Chest Scans," in *ICASSP 2021 - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8573–8577.
- [15] F. Bougourzi, R. Contino, C. Distanto, and A. Taleb-Ahmed, "CNR-IEMN: A Deep Learning Based Approach to Recognise Covid-19 from CT-Scan," in *ICASSP 2021 - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8568–8572.
- [16] B. Li, Q. Zhang, Y. Song, Z. Zhao, Z. Meng, and F. Su, "Diagnosing COVID-19 from CT Images Based on an Ensemble Learning Framework," in *ICASSP 2021 - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8563–8567.
- [17] Z. Tang, W. Zhao, X. Xie, Z. Zhong, F. Shi, J. Liu, and D. Shen, "Severity assessment of coronavirus disease 2019 (covid-19) using quantitative features from chest ct images," *arXiv preprint arXiv:2003.11988*, 2020.
- [18] W. Lu, J. Wei, T. Xu, M. Ding, X. Li, M. He, K. Chen, X. Yang, H. She, and B. Huang, "Quantitative CT for detecting COVID-19 pneumonia in suspected cases," *BMC infectious diseases*, vol. 21, no. 1, pp. 1–8, 2021.
- [19] I. Wolf, M. Vetter, I. Wegner, M. Nolden, T. Bottger, M. Hastenteufel, M. Schobinger, T. Kunert, and H.-P. Meinzer, "The medical imaging interaction toolkit (MITK)," in *Medical Imaging: Visualization, Image-Guided Procedures, and Display*, vol. 5367, 2004, pp. 16–27.
- [20] Z. Ye, Y. Zhang, Y. Wang, Z. Huang, and B. Song, "Chest CT manifestations of new coronavirus disease 2019 (COVID-19): a pictorial review," *European Radiology*, vol. 30, no. 8, pp. 4381–4389, 2020.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [22] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jul 2017, pp. 2261–2269.
- [23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.
- [24] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020.
- [25] S. Heidarian, P. Afshar, N. Enshaei, F. Naderkhani, M. J. Rafiee, A. Oikonomou, A. Shafiee, F. B. Fard, K. N. Plataniotis, and A. Mohammadi, "Robust automated framework for covid-19 disease identification from a multicenter dataset of chest ct scans," *arXiv preprint arXiv:2109.09241*, 2021.