Deep learning-based automated COVID-19 classification from computed tomography images

Kenan Morani and D. Unay

Electrical and Electronics Engineering Department, Izmir Democracy University, Izmir, Turkey

ABSTRACT
This paper introduces a lightweight Convolutional Neural Networks (CNN) method for image classification in COVID-19 diagnosis. The proposed approach emphasizes simplicity while achieving high performance, and it leverages a meticulously annotated database. The CNN model consists of four convolutional layers, followed by flattening and two dense layers. The methodology focuses on classifying 2D slices of Computed Tomography (CT) scans. To enhance accuracy, the slices undergo anatomy-relevant masking and the removal of non-representative slices from the CT volume. This is achieved by cropping a fixed-sized rectangular area to capture the relevant region of interest and using a threshold based on bright pixels in binarized slices. The proposed methodology demonstrates improved quantitative results in slice classification by employing slice processing techniques. Additionally, augmentation techniques such as class weight balancing, slice flipping, and a learning rate scheduler are applied to diagnose at the slice level. For patient-level diagnosis, a majority voting method is employed by considering the slices of each CT scan. The proposed method surpasses the baseline approach and other alternatives in terms of macro F1 score, both on the validation set and a test partition containing previously unseen images from the rigorously annotated dataset.

1. Introduction
The COVID-19 virus, or the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), is believed to have initially originated from a species of bats and was transmitted to human beings in December 2019. The virus spread rapidly all around the world, affecting lots of people and claiming lives (Hassanin et al 2021). COVID-19-infected individuals have experienced fever at the onset, generalised fatigue, dry coughing, and diarrhoea among other possible symptoms (Mizrahi et al 2020). Early detection and isolation are vital to successfully handle the COVID-19 pandemic.

Studies have shown the importance of lung imaging for that cause as in (Rosenthal 2020). The traditional method of detecting COVID-19 is through a Polymerase Chain Reaction (PCR) test. This test involves taking a sample from the respiratory tract (usually a nasal or throat swab) and amplifying any viral genetic material present in the sample. PCR is a highly sensitive and specific test and is considered the gold standard for diagnosing COVID-19. On the other hand, a CT (Computed Tomography) scan is a type of medical imaging that uses X-rays to produce detailed images of the body. While a CT scan can help diagnose certain conditions, it is not typically used as a primary diagnostic tool for COVID-19. This is because CT scans are less sensitive and less specific than PCR tests for detecting the virus. Additionally, CT scans involve the use of ionising radiation, which can potentially be harmful to the body, especially with repeated exposure. Therefore, CT scans are considered to be more invasive than PCR tests, which only require a simple swab of the respiratory tract. Nevertheless, a CT scan is needed when the infection is so severe that it spreads to the lungs, and detection by nose or oral swabs may not be possible. As reported by the Republic of Turkey, Ministry of Health (2020), a CT scan is advised in either or both of the following:

(1) History of fever in child patient or body temperature at or above 38 degrees Celsius.
(2) Existence of any findings during auscultation.

Furthermore, a CT scan may also be conducted depending on the status of the patient, whose respiratory findings cannot be examined with X-ray image modality, or who deteriorates clinically. The reasons for using CT image modalities for COVID-19 diagnosis are further explained in the following paragraph.

CT scans can provide additional information about the severity and progression of COVID-19 that traditional PCR testing may not always be sufficient for. CT scans can identify lung damage, inflammation, blood clots, and pneumonia associated with COVID-19, which can help doctors assess the severity of the disease and allocate resources efficiently (Xiong et al 2020). Moreover, CT can be used to triage patients, particularly those with negative initial RT-PCR but clinically suspected of COVID-19 (He et al 2020). Therefore, the use of CT is justified as a complementary tool for diagnosing and managing COVID-19. In certain cases, such as when checking the severity of the illness and monitoring the progress of the disease, CT imaging is vital. All in all, we can see the importance and necessity of acquiring CT images for accurate diagnosis of COVID-19 in certain cases mentioned above. In our work, we chose to develop a method using CT scans.
Automated solutions were proposed by (Rahmani et al. 2022) for COVID-19 detection from images of different modalities using various algorithmic methods. The proposed methods would report classification performances using different measures including accuracy, precision, recall, specificity, and F1 scores as in an example reported by Islam et al. (2021). However, in case of an unequal number of observations in the classes (unbalanced data), the accuracy of the solutions is important, but it might be misleading. If this is the case, then the model can be assessed in terms of its ‘Precision’ and ‘Recall’. If the former is high, then that means the model gives more relevant results than irrelevant ones. On the other hand, if the latter is high then that means the model gives most of the relevant results (whether irrelevant ones are also returned). Therefore, for unbalanced classification problems, the weighted average of the two scores or the macro F1 score can be used to evaluate the classification performance of a model in a more reliable manner as discussed in different papers, such as the one proposed by Waleed Salehi et al. (2020).

If a model has a high precision score, it means that the model is making very few false positive predictions. On the other hand, if the recall score is high, it means that the model is correctly identifying a high proportion of the actual positive instances in the dataset. However, optimising for precision or recall alone may not be enough in unbalanced classification problems. For example, a model that always predicts the majority class may achieve a high precision score but a low recall score. In this case, the model is only detecting a small proportion of the actual positive instances in the dataset. Assuming we have 100 instances, out of which 95 belong to class A and 5 belong to class B, and the model predicts two instances for class B correctly, then the macro F1 score for class B would be 0.5, indicating the model needs improvement in correctly identifying the minority class instances. Therefore, to evaluate the performance of a model in unbalanced classification problems, both precision and recall metrics are used together. The F1 score is a metric that combines precision and recall into a single score. A high F1 score indicates that the model has a good balance between precision and recall. In addition, for unbalanced classification problems, the weighted average of the F1 scores or the macro F1 score can be used to evaluate the classification performance. The weighted average F1 score gives more weight to the performance of the minority class, which can be useful in cases where the minority class is of particular interest or concern. The macro F1 score gives equal weight to the performance of all classes, which can be useful in cases where all classes are equally important.

In our work, the macro F1 score was used to compare the performances of different deep learning models and methods validated on the same dataset. The comparison was made at two levels: slice level and patient level. Our deep learning model resulted in a state-of-the-art macro F1 score at the slice level. Results at the patient level were obtained by combining the deep learning model with two essential processing techniques: slice processing and hyperparameter tuning. The final method achieved a sufficiently high macro F1 score at the patient level such that it well exceeds the baseline score and many other alternatives on the COV19-CT-DB database.

The design of those models/methods is aimed at finding an automated solution for COVID-19 diagnosis via CT images. The proposed classification solution in this paper progresses from a deep learning model consisting of four similar 2D convolutional layers, followed by a flattening layer and two dense layers, to a method that is then used to make diagnosis predictions at the patient’s level using different thresholds via class probabilities and voting from the slices.

The main contributions of this work can be listed as follows:

- We propose a less complex methodology to achieve COVID-19 diagnosis from CT images.
- We show that processing CT images with a Region of Interest (ROI) dedicated to the lung region improves diagnostic performance.
- We propose a mechanism to take patient-level diagnosis from slice-level in each CT scan.
- We evaluate the performance of the proposed solution on a relatively large, and rigorously annotated dataset designed solely for COVID-19 diagnosis.

2. Related work

Recently, deep learning-based decision support systems are proposed for COVID-19 diagnosis using either CT or X-ray modalities (Wynnats et al. 2020; Huang et al. 2020; Fang et al. 2022). Some of these systems are developed based on pre-trained models with transfer learning, such as in (Panwar et al. 2020; El Asnaoui and Chawki 2021), while a few others are introduced using customised networks trained from scratch, such as in (Fan et al. 2020; Nayak et al. 2021; Mary Shyni and Chitra 2022). In these solutions and many others, high performances were mainly reported, rather than method complexity.

One approach by Fan et al. (2020) proposed a novel COVID-19 lung CT infection segmentation network, named Inf-Net. The work utilised implicit reverse attention and explicit edge attention aiming at the identification of infected regions in CT images. They also introduced a semi-supervised solution, Semi-Inf-Net, aiming at alleviating the shortage of high-quality labelled data. The proposed method was designed to be effective in case of low contrast regions between infections and normal tissues.

Another approach, by Nayak et al. (2021), presented a deep learning-based automated method validated using chest X-ray images collected from different sources. Different pre-trained CNN models were compared, and the impact of several hyperparameters was analysed to eventually obtain the best-performing model. In this work, the ResNet-34 model outperformed other competitive networks, and thus the development of effective deep CNN models (using residual connections) proved to give more accurate diagnoses of COVID-19 infection.

More recent work, by Ali Ahmed et al. (2022), introduced an ensemble deep neural network (IST-CovNet), providing an evaluation of different 2D and 3D approaches on two different datasets and discussing the effects of pre-processing, segmentation, and classifier combination steps on the performance of the approach. The final model combined the use of a novel attention mechanism with a slice-level combination using LSTMs (Long Short-Term Memory) and an extended architecture for 3D data. This approach has proven to increase accuracies in both 2D and 3D models validated on the public dataset.
‘MosMedData’, introduced by Morozov et al. (2020), achieving state-of-the-art performance. Furthermore, the authors introduced a large, collective dataset referred to as ‘IST-C’, which was made public to contribute to the literature. Their approach also achieved high performance on their introduced dataset.

Another recent work, by Wang et al. (2022), used a chest CT dataset with four categories/class labels and incorporated a 12-layer attention-based VGG-style network with a convolutional block attention module and improved multiple-way data augmentation to resist overfitting. The proposed AVNC achieved high sensitivity/precision/F1 scores per class and a micro-averaged F1 score of 96.87%, outperforming 11 state-of-the-art approaches. Overall, the AVNC model is effective in recognising COVID-19 diseases.

Yet another work by Zhang et al. (2022) introduced a novel seven-layer convolutional neural network-based smart diagnosis model for COVID-19 diagnosis (7-L-CNN-CD) using chest CT images. The model used 14-way data augmentation and introduced stochastic pooling to replace traditional pooling methods. Results from 10 runs of a 10-fold cross-validation experiment showed that the proposed 7-L-CNN-CD approach achieved a sensitivity of 94.44 ± 0.73, a specificity of 93.63 ± 1.60, and an accuracy of 94.03 ± 0.80, outperforming several state-of-the-art algorithms.

While there is a multitude of studies aiming at COVID-19 diagnosis using different datasets or combinations of these, we focused on a recent, heterogeneous database of CT scan images, called ‘COV19-CT-DB’. Our work was employed on this particular database for the advantages that come from its large size, challenging nature, and its rigorous and accurate annotation process. The reasons and the advantages are further explained in the ‘DATASET’ section.

The database was shared via an international competition about mid of the year 2021 and was used by several international teams for COVID-19 diagnosis. At the time of conducting this research, such a big number of annotated CT scans was not possible to come by to the best of our knowledge. Furthermore, we believe that reaching high performance on the COV19-CT-DB database will confirm the robustness of our methodology for other similar CT scan problems, given the challenging nature of the database.

Using the COV19-CT-DB series of images, introduced by Kollias et al. (2021a), a baseline approach introduced a deep neural network based on CNN-Recurrent Neural Network (RNN) architecture. The CNN part of the model extracts features from the images while the following RNN part takes the final diagnostic decision as published in Kollias et al. (2018, 2020, 2021b).

Another study by Anwar (2021), which used the same database (COV19-CT-DB) for validation, introduced a different method. In this study, 2D deep CNN models were trained on individual slices of the database. Performances of the following pre-trained models were compared: VGG, ResNet, MobileNet, and DenseNet. Evaluation of the models was reported both at slice level (2D) as well as at patient/volumetric level (3D) using different thresholding values for voting at the patient level for the latter. The best results were achieved using the ResNet14 architecture (referred to as the AutoML model) via 2D images.

In another study, conducted by Tan and Liu (2021), a 3D CNN-based network with BERT was used to classify slices of CT scans. Only part of the images from the COV19-CT-DB database was used in this work. Firstly, the training and validation sets of images were passed through a lung segmentation process to filter out images of closed lungs and to remove the background. Secondly, a resampling method was used to select a set of a fixed number of slices for training and validation. Finally, a 3D CNN-based model was used followed by a second-level MLP classifier to capture all the slices’ information from 3D-volumetric images. The final model architecture achieved improved accuracy and macro F1 score on the validation set.

Hsu et al. (2021) introduced 2D and 3D deep learning models to predict COVID-19 cases. The 2D model, named Deep Wilcoxon signed-rank test (DWCC), adopted non-parametric statistics for deep learning, making the predicted result more stable and explainable, finding a series of slices with the most significant symptoms in a CT scan. On the other hand, the 3D model was based on pixel- and slice-level context mining. The model was termed a Convolutional CT scan Aware Transformer (CCAT) and used to further explore the intrinsic features in temporal and spatial dimensions.

More work on the same dataset involved deploying a hybrid deep learning framework named CTNet which combines a CNN and a transformer network for the detection of COVID-19. The method proposed by Liang et al. (2021) deployed a CNN feature extractor module with Squeeze-and-Excitation (SE) attention to extract features from the CT scans, together with a transformer model to model the discriminative features of the 3D CT scans. The CTNet provides an effective and efficient method to perform COVID-19 diagnosis via 3D CT scans with a data resampling strategy. The method’s macro F1 score exceeded the baseline on the test partition of the COV19-CT-DB database.

Additionally, on the COV19-CT-DB, two experimental methods that customised and combined Deep Neural Networks to classify the series of 3D CT scans chest images were deployed. The proposed methods included experimenting with two backbones: DenseNet 121 and ResNet-101. The experiments were separated into two tasks, one was for the combination of ResNet and DenseNet backbones while the other one was for the DenseNet backbones combination. Introduced by Trinh and Van Nguyen (2021), the method’s macro F1 score on the test partition of COV19-CT-DB exceeded the baseline score as can be seen on the leader board of ICCV 2021 Workshop: MIA-COV19D, introducing AI-enabled Medical Image Analysis and COVID-19 Diagnosis Competition’.

The proposed deep learning approaches in the literature summarised above achieved high macro F1 scores on the COV19-CT-DB database and other data. However, these methodologies can be quite complex; implementing a full pipeline of segmentation, slice removal, and then classification, or using transfer learning architectures with borrowed model weights to obtain high performance. For example, the baseline methodology (Kollias et al. 2020, 2021b), focuses on implementing a rather complex CNN-RNN architecture. In our paper, however, we present a computationally less expensive and less complex solution for COVID-19 diagnosis and detection. Despite the simplicity of our solution, the performance of our methodology is comparable to the state-of-the-art methods on the same dataset and thus provides a noteworthy alternative.
3. Methodology

The methodology used in this work includes slice processing before inputting images into a CNN model, which is tuned to predict the probability of CT slices being COVID. Then we use a majority voting method in each CT scan to take the final diagnosis decision of COVID-19 existence in patients. Figure 1 shows a flow diagram of the proposed methodology. Section 3 discusses the model architecture and hyperparameters tuning, slice processing techniques for slice removal as well as taking a diagnostic decision from slice-level to patient-level in detail.

3.1 The model architecture

The proposed model’s architecture consists of four similar sequential 2D convolutional layers followed by a flattened layer, and two dense layers. The input images are designed to have one channel, greyscale. The flattened layer is used to transform the 2D feature maps into a 1D vector that can be fed into the dense layers. The first dense layer is responsible for mapping the input features to a higher-dimensional space, while the second dense layer maps the features to the output classes, using a softmax activation function. The input images are originally one-channelled and therefore the model architecture takes one-channel input. This makes the process less complex and matches the original image shapes at the same time. These tie well with the aim of our work.

The numbers of filters in the convolutional layers are 16, 32, 64, and 128, in order. All filters are 2D, and \(3 \times 3\) in size. Padding was also applied in all four convolutional layers, to match input and output image sizes \(\text{Padding}=\text{same}\). Padding allows for more space for the kernel to cover the image. Adding padding to an image processed by a CNN allows for a more accurate analysis of images (Tang et al. 2019). The four layers had batch normalisation and max pooling \((2,2)\), and ReLu (Rectified Linear unit) activation functions with a binary output for the final diagnosis. The batch Norm is a normalisation technique applied between the layers of a Neural Network instead of in the raw data. It is done along mini-batches instead of the full data set and serves to speed up training and use higher learning rates, making learning easier (Chen et al. 2017). Figure 2 shows the proposed CNN model’s architecture.

Following the four convolutional layers was a flattening layer, followed by a dense layer with a dimensionality of 256, batch normalisation, ReLu activation function, and a dropout of 0.1. The model then ends with a dense layer using a sigmoid activation function.

The model could be replicated in the following sequence:

- Convolutional layer \((512, 512, 16)\) with padding, Batch Norm, Relu activation
- Max pooling 2D \((256, 256, 16)\).

![Figure 1. Block diagram of the proposed approach.](image)

![Figure 2. Visualization of the layers of the proposed CNN model.](image)
• Convolutional layer (256, 256, 32) with padding, Batch Norm, ReLu activation
• Max pooling 2D (128, 128, 32)
• Convolutional layer (128, 128, 64) with padding, Batch Norm, ReLu activation
• Max pooling 2D (64, 64, 64)
• Convolutional layer (64, 64, 128) with padding, Batch Norm, ReLu activation
• Max pooling 2D (32, 32, 128)
• Flatten (131072)
• Dense (256)
• Batch Norm (256)
• ReLu activation (256)
• Dropout (256)
• Dense (1)

The output of the final dense layer ‘Dense (1)’ is class 1 probability, i.e. the probability of the CNN model predicting class 1 corresponding to the Non-COVID class.

The motivation behind the model architecture is to adopt symmetric and less complex deep learning network architecture with standard components. For example, using a $3 \times 3$ filter was preferred since the processing time of $5 \times 5$ or larger filters would be almost three times longer or more. Another example is the use of the ReLu activation function. Apart from helping with the gradient vanishing problem, ReLu Function is very simple to calculate, as it involves only a comparison between its input and the value ‘0’. Consequently, its usage helps to prevent the exponential growth in the computation required to operate the neural network compared to non-linear functions such as sigmoid as explained by Kiliçaslan et al. (2021)’s work. What is more, using a $2 \times 2$ 2D max pooling with 'same' padding is a common choice as it can help improve the performance of a neural network by reducing the computational complexity and memory requirements of subsequent layers, increasing robustness to small translations, promoting non-linear feature extraction, and acting as a form of regularisation to prevent overfitting.

To build an efficient and less hand-engineered CNN model, different numbers of hidden layers were tested against the validation accuracy at the slice level. Using a three-layer depth model has substantially reduced validation accuracy as compared to the 4-layer depth usage. Furthermore, using more complex model architectures only trivially increased the validation accuracy. Therefore, a consensus was on using a four-layer depth CNN model in our methodology.

3.2 Slice processing

The activation visualisation results of classification on the database show room for improvement in terms of accuracy. Following the Grad-Cam visualisation in Figure 9, one can theorise that masking the images with the lung area should improve the performance as the model can better learn to discriminate COVID from Non-COVID. To prove the theory and improve the performance, a fixed-sized rectangular Region of Interest (ROI) was applied to localise the anatomy of interest (lung regions) in the slices. The rectangular area was empirically set to contain both left and right lungs over all slices of every scan in the database. Figure 3 shows this ROI overlaid on an original slice.

The above-mentioned fixed-sized masking may be affected by variations in the sizes and shapes of the lung seen at different slices. For example, in the upper slices, the lungs appear smaller (relative to the mid-slices), while the lower slices display the lungs in the shape of a banana. Nevertheless, the fixed-size cropping proposed is an attempt to localise the lung region from the slices in a simple and less complex way. It was conducted by choosing one fully representative slice from a CT scan in the training set and a manually drawn rectangle aimed to capture the lung area. The resulting size of the rectangle is $227 \times 300$, which is the size of the input image to the CNN model. In an attempt to account for varying sizes of the lung region in the slices, we work on removing non-representative slices as follows.

After cropping, thresholding was applied to identify and remove the uppermost and lowermost slices of the CT scans, corresponding to non-representative slices of the lung volume, aiming to achieve better performance at the patient-level diagnosis. Identification of the non-representative slices was realised based on the number of bright pixels in a binarised slice. This procedure is explained below.

Firstly, the cropped images were blurred by using a Gaussian filter to suppress noise and thus enhance large structures in the image. A Gaussian function with a standard deviation of one was convolved with the cropped image's pixel intensity values. The Gaussian function can be expressed in two dimensions as in Equation 1:

$$G(x,y) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

Where $x$ and $y$ are the distances from the origin in the horizontal and vertical directions, respectively, and $\sigma$ is the standard deviation ($\sigma = 1$).

The Gaussian filtering was chosen over other filters because it is less computationally expensive to implement thanks to its filter separability property. The 2D Gaussian filter is separable into two 1D filters and can be expressed as the outer product of the two, which in turn means that the filter can be split into two

Figure 3. Static rectangular cropping of images.
passes, horizontal and vertical (Talbi et al. 2015). With a square image \( x[k,l] \) of size \( N \times N \) and a square filter kernel \( h[n,m] \) such as the Laplacian filter of size \( M \times M \), the raw 2D convolution to produce the cropped output image of size \( N \times N \) requires about \( N^2 \times M^2 \). Multiply – Accumulates (MACs). The raw 2D convolution between \( x[k,l] \) and the filter \( h[n,m] \) is implemented by using two ‘for loops’ to range through each output pixel and two additional for loops to perform the 2D convolution at that pixel location. Hence a total of four nested for-loops are required resulting in a complexity of \( O(N^2 \times M^2) \). With a separable filter \( h[n,m] \), such as the Gaussian filter, we can have \( h[n,m] = f[n] \times g[m] \), where \( f[n] \) and \( g[m] \) are the one-dimensional filters. In this case, the convolution between the image \( x[k,l] \) and the filter \( h[n,m] \) can be performed without a raw 2D convolution sum, by the following approach:

- First, perform a 1D convolution between columns of \( x[k,:) \) and the 1D filter \( f[n] \), which requires about \( N \times M \) MACs to complete. This operation should be performed for each column of \( x[k,l] \) by proceeding along its horizontal, \( N \) many, columns. Hence a total of \( N \times M \times N \) MACs will be required to complete the first step to produce the intermediate image \( w[o,p] \).
- Then, apply a similar algorithm, with the filter \( g[m] \) and rows of the intermediate image \( w[,:p] \), which will require a similar number of MACs as \( N \times M \times N \).

Hence in total, only about \( 2 \times N^2 \times M \) MACs will be needed for the separable implementation of the 2D convolution. The actual number depends on the cropping type applied. Thus, a complexity of only about \( O(N^2 \times M) \) is attained. In conclusion, for the implementation of the separable convolution algorithm, only three nested ‘for loops’ are required. This filtering method helps to keep the approach drastically less time and memory-consuming when training or testing our method.

Secondly, a histogram-based binarization was applied to the resulting blurred images. By looking at the slice’s histograms, an estimated threshold for histogram-based image binarization was empirically chosen to be 0.45. This fixed threshold was chosen after applying scale ([0,1]) normalisation to the voxel intensities of each scan. Figure 4 illustrates an exemplary histogram of one of the Gaussian blurred images in the database and the corresponding resulting binarized image.

Finally, the binarized image’s pixels were used to find a threshold to remove non-representative slices of the CT volume. To choose the threshold, four candidate CT scans were arbitrarily selected from the training set (CT scans 5, 6, 7, and 8) and random slices from them were processed as explained above. To indicate the importance of the slices in the CT volume, labels from one to three were visually assigned by careful examination; three being the most representative slice and one being the least representative; a representative slice means a slice that shows a large area of the lung. Similarly,

**Figure 4.** Histogram of cropped and blurred image (top) and the resulting binary image (bottom).
less representative slices are those that display little to no lung area. Figure 5 shows the results of four candidate CT scan slices.

The chosen filtering threshold ratio for the number of bright pixels was 0.066 (corresponding to 4500 out of a total of 68,100 pixels in a 227 × 300 sized slice). Consequently, if the resulting binarized image has a bigger number of bright pixels than the threshold, this indicates that the slice corresponding to the binarized image will be kept in the CT scan, otherwise, it will be removed. The threshold was chosen carefully so that at least one slice in each CT scan volume is not removed which will allow us to take the final diagnosis decision for all CT scans or all patients. Table 1 shows the error percentage, where errors indicate CT scan images left fully empty (zero slices) after our selected filtering threshold and slightly higher one on all three partitions. The error is then calculated by dividing the number of CT scans left empty after slice removal divided by all the CT scan in the partition. The results indicate that a filtering threshold in the range of [3400,3500] is optimum for the given data partitions to capture all the CT scans.

The slice processing methodology reduces the number of slices in the dataset by including only the representative slices. Accordingly, the numbers of training and validation slices were reduced to 280,462 (corresponding to a 16% reduction) and 63,559 (15.3% reduction), respectively. Please note that the original number of the slices is as shown in the ‘DATASET’ section.

The motivation behind slice processing is to localise the lung region of the images simply and efficiently. It also aims at removing non-representative slices from each CT scan in a way that will keep at least one slice for each scanned image to take the final diagnosis for each patient in all sets of the COV19-CT-DB.

### 3.3 Hyperparameters tuning

Hyperparameters were tuned while training the CNN model by changing the learning rate, and class weight balance, and using augmentation techniques.

Adam optimiser was used with an initial learning rate of 0.1. The learning rate decreased exponentially via a learning rate scheduler. The decay of the learning is calculated as in Equation 2:

\[
\text{LearningRate} = \text{Initial Learning Rate} \times \text{Decay Rate}^\frac{\text{Optimiser Step}}{\text{Optimiser Decay Step}}
\]  

(2)

The decay rate was set to 0.96. The value of steps divided by decay steps is an integer division, i.e. the decayed learning rate follows a staircase function.

The optimiser’s steps were defined using floor divisions as in Equation 3:

\[
\text{train step} = \left\lfloor \frac{\text{number of training slices}}{\text{batch size}} \right\rfloor
\]

\[
\text{validation step} = \left\lfloor \frac{\text{number of validation slices}}{\text{batch size}} \right\rfloor
\]

(3)

Decay steps were set every 100,000 steps.

Furthermore, class weights were used to balance out varying numbers of input images in the classes. The number of training samples after using non-representative slices removal is reported in Section 3.2. The class weight is calculated using the formula in Equation 4:

\[
\text{class weight} = \frac{\text{number of slices for a class in the training set}}{\text{number of all slices in the training set}}
\]

(4)

**Table 1.** Percentage errors of different filtering thresholds on all dataset partitions (no error is observed on the Validation COVID [165 CTs], Validation Non-Covid [209 CTs], or Training Covid [690 CTs] sets).

| Data Partition/Threshold | 3400 (Selected Threshold) | 3500 |
|--------------------------|---------------------------|------|
| Training Non-Covid [870 CTs] | 0% (No Empty CT) | 0.090% (1 Empty CT) |
| Test [3455 CTs] | 0% (No Empty CT) | 0.005% (5 Empty CTs) |
Finally, horizontal and vertical flipping were used as augmentation techniques. These image-flipping techniques aimed at improving the accuracy by smoothing the effects of the content variations present in the slice as explained by Hussain et al. (2017).

3.4 Patient-level decision

At the patient level, different class probability thresholds were tried and compared using class prediction probability to achieve the highest diagnosis accuracy. The class probability thresholds were based on the probability of predicting class 1 (Non-COVID); if the output probability for class 1 is greater than the chosen threshold, then the slice would be predicted as Non-COVID. Otherwise, the slice would be predicted as COVID. In that, if the number of COVID slices is equal to the number of Non-COVID slices in any one of a CT volume, then the decision is that the patient is a Non-COVID. This slice-level decision can be expressed as follows:

if Class1 probability > class probability threshold:
    Predict slice as Non-COVID
else:
    Predict slice as COVID

After slice-level predictions are obtained, a patient is diagnosed based on the presence/absence of COVID slices in his/her CT: if the patient’s CT data contains more Non-COVID predicted slices than COVID predicted slices, the patient is diagnosed as Non-COVID else the patient is diagnosed as COVID (majority voting method).

The clinical relevance of the patient-level diagnosis approach we presented above can be explained as follows. Assuming that a patient has lung damage due to COVID seen in 30% of its slices. So, our network classifies around 30% of the slices as COVID and the rest as Non-COVID, and thus the final result will be Non-COVID (in line with majority voting).

While even a minor anomaly seen in a single slice may be attributed to a disease, we speculate that in the Covid case, a reasonable amount of involvement is necessary for the diagnostic decision to be taken, and our deep learning model is highly sensitive to even the smallest anomalies observed in the slices.

Please note that we also tried the ‘all-or-nothing approach’ where the COVID diagnosis decision is taken even though a single slice is predicted as COVID, but that approach yielded less accurate results – as elaborated in the results section – supporting our above observations.

3.5 Performance evaluation

At the slice level, validation accuracy was used to evaluate the method’s performance. The accuracy is calculated as in Equation 5:

\[
\text{Accuracy} = \frac{\text{number of correct predictions}}{\text{total number of predictions}}
\]  

(5)

Over the training, the maximum validation accuracy reached was used as a model performance measurement.

Furthermore, to report the confidence intervals of the results obtained, the Binomial proportion confidence intervals for the macro F1 score are used. The confidence intervals were calculated as discussed in (28 May 2018 posting Jason Brownlee in Statistics on machinelearningmastery; unreferenced, see ‘confidence-intervals-for-machine-learning’), where the radius of the interval is defined as in Equation 6:

\[
\text{radius of interval} = z \times \sqrt{\frac{\text{accuracy} \times (1 - \text{accuracy})}{n}}
\]  

(6)

n is the number of samples used.

In the above formulation, z is the number of standard deviations from the Gaussian distribution, which is taken as \(z = 1.96\) for a significance level of 95%.

At the patient level, the proposed model was evaluated via the COV19-CT-DB database. As explained in Takahashi et al. (2022) and Opitz and Burst (2019), the macro F1 score was calculated after averaging precision and recall matrices (the arithmetic mean) at patient-level as in Equation 7:

\[
\text{macro F1} = \frac{2 \times \text{average precision} \times \text{average recall}}{\text{average precision} + \text{average recall}}
\]  

(7)

Performance evaluation of our method is conducted at the slice level and at the patient level, where the former corresponds to considering 2D slices individually in any quantitative or qualitative analysis. Whereas, in patient-level results, CT volumes are considered as a whole, and thus the prediction is emphasising 3D-CT prediction value or patient-level rather than each 2D slice’s predicted value.

4. Dataset

COV19-CT-DB is the dataset used for validating the methodology proposed in this paper. The CT images in the database were manually annotated by experts and distributed for academic research purposes via the ‘AI-enabled Medical Image Analysis Workshop and COVID-19 Diagnosis Competition’

The database consists of about 5000 3D chest CT scans acquired from more than 1000 patients. The training set contains 1560 scans in total with 690 of the cases – being COVID while the rest (870) belong to the Non-COVID class. The validation set contains, in total, 374, where 165 are COVID cases and 209 are Non-COVID cases. Figure 6 shows the distribution of the CT images concerning the classes in the training and validation sets. The CT scans in the database contain largely varying slice numbers, ranging from 50 to 700. Please note that the COV19-CT-DB database includes three different sets/partitions: a training set, a validation set, and a test set.

The data is unbalanced in terms of the number of 2D slices for both COVID and Non-COVID classes. The images, which are input to the model, were received mainly in Joint Photographic Experts Group (JPEG) format, greyscale images, with 8-bit depth. The images were all resized to an original size of 512 × 512 and processed as such.

The dataset used in our paper is preferred for its rigorous and accurate annotation process and its large size. The COV19-CT-DB database was annotated with the help of two radiologists and two pulmonologists with more than 20 years of medical experience in
the field. Annotation of the dataset had been realised via the results of a PCR test and a consensus agreement among all medical experts involved, where the agreement reached 98%. Furthermore, COV19-CT-DB is a large dataset, and we believe its use will ensure the robustness of our methodology and will make it generalisable on other smaller datasets.

The dataset was also preferred for its challenging nature. In the provided COV19-CT-DB, the slice thickness, the number of slices, and the observed anatomy in the 3D field-of-view in each CT scan vary – as elaborated in the dataset section. This variability further increases the challenging nature of the problem and potentially makes an accurate predictive method pervasive and robust. The model reaching high performance on the COV19-CT-DB is expected to perform well on more consistent datasets.

5. Results

5.1 Slice-level results

Training the CNN model using a batch size of 128 took about two and half days over a workstation using GNU/Linux operating system on 62GB System memory with Intel X1(R) W-2223 CPU @ 3.60 GHz processor.

Our method achieved an accuracy of 84.5% on the validation set at the slice level. Figure 7 shows the evolution of validation and training accuracies when using slice processing techniques. Early stopping was used during training to halt the training if the validation accuracy does not show improvement after 7 epochs.

The interval of the validation accuracy score with a 95% significance level is calculated as in Equation 8:

$$ interval = 1.96 \times \sqrt{\frac{0.845(1 - 0.845)}{7532}} \approx 0.0013 \quad (8) $$

The results show a narrow deviation from our reported validation accuracy.

To evaluate the improved results achieved by using the slice processing technique before training, we use validation accuracy at the slice level as a performance metric. The proposed CNN model reaches a maximum of 80.8% accuracy on the validation set when the original-sized images are used as input, with similar hyperparameters tuning. That concludes that processing the slices as
described above is a major factor to improve the validation accuracy at the slice level. Figure 8 shows the evolution of training and validation accuracies when slice processing is not employed.

In terms of the model over/underfitting, we realise a trend where the validation accuracy keeps increasing but the validation loss starts to increase instead of decreasing during training. We understand that accuracy and loss are not necessarily exactly (inversely) correlated, as loss measures a difference between the raw prediction (float) and the class label (0 or 1), while accuracy measures the difference between the threshold prediction (0 or 1) and the class label. So, if raw predictions change, loss changes but accuracy is more ‘resilient’ as predictions need to go over/under a threshold to affect the accuracy. Here, we present our analysis of this issue and our CNN model. Two phenomena can be discussed here:

1. Some images with borderline predictions get predicted better and so their output class changes (for example, a Covid image whose prediction was 0.4 becomes 0.6). This explains the regular ‘loss decreases while accuracy increases’ behaviour that we expect.

2. Some images with very bad predictions keep getting worse (for example, a Covid image whose prediction was 0.2 becomes 0.1). This leads to a less expected ‘loss increases while accuracy stays the same’ behaviour. Note that cross-entropy loss measures the calibration of a model and when it is used for classification as we did in our methodology, bad predictions are penalised much more strongly than good predictions are rewarded. For a Covid image, the loss is \(\log(1 - \text{prediction})\), so even if many Covid images are correctly predicted (low loss), a single misclassified Covid image will have a high loss, hence ‘blowing up’ the average loss.

![Figure 8. Evolution of training and validation accuracy without slice processing.](image)

![Figure 9. Evolution of training and validation losses with slice processing techniques.](image)
The network is starting to learn the patterns that are only relevant for the training set, which hinders its generalisation capability, leading to phenomenon 2. Some images from the validation set get predicted to increase incorrectly. However, the model is at the same time still learning some patterns, which are useful for generalisation (phenomenon 1, ‘good learning’) as more and more images are being correctly classified. All that being said, our resulting model was chosen so long as it was learning and thus the model was trained for the full 40 epochs. Figure 9 shows the validation loss relative to the training loss, proving that the model is doing not as well on validation, but eventually, it is still learning.

The effect of increasing and decreasing the batch size on diagnostic performance has been explored as well. In that, using a batch size of 64 allowed the model to reach a slice-level validation accuracy of 83.3% with slice processing techniques. A batch size of 128 increased the resulting validation accuracy to the number reported above. With that, we can consent to use a 128-batch size.

The effect of varying the number of hidden layers was also explored. The results show slice-level validation accuracy of 81.7% when a three-layer deep CNN model was used. To achieve that, the last hidden layer with 128 filters was eliminated. The validation accuracy is sufficiently less than the number reported above as compared to a four-layer deep model when slice processing techniques are used. On the other hand, using more layers in the CNN model was tried by starting by adding one layer. As we move forward in adding the layers, the detected patterns get more complex; hence there are larger combinations of patterns to capture. That is why we increase the filter size in subsequent layers to capture as many combinations as possible as explained by (31 December 2018 posting Adrian Rosebrock on pyimage-search; unreferenced). Therefore, we multiply the number of filters in the fifth layer to have 256 filters, leaving all other convolutional layer settings the same. Using a five-layer deep CNN model increased the validation accuracy at the slice level only trivially to 84.8%.

To assess the model’s complexity, the number of trainable parameters was used as a measuring factor for different CNN models with varying depths. The number of parameters was compared by eliminating the fully connected layer and the output layer. As observed in Table 2 validation accuracy improves as complexity increases.

The complexity seems to increase about four times as we add one more convolutional layer to the CNN model. However, the accuracy only increases by a factor of 0.038 by adding a fifth hidden layer. Looking at the complexity results of different layer depths and considering the improvement achieved in validation accuracy at the slice level, a consensus was on using a four-layer deep CNN model as our standard model architecture.

It is well established that improved quantitative results often come at the cost of increased complexity and computational resources required for a model. However, this should not be seen as a limitation or a drawback, but rather as a necessary trade-off to achieve better performance. In many cases, the benefits of improved accuracy or efficiency far outweigh the added cost, especially in high-stakes applications such as medical diagnosis or financial forecasting.

Despite the increase in cost associated with achieving better performance, it remains well within acceptable limits when compared to other available alternatives on the dataset as can be seen in our results. Moreover, it serves the essential purpose of developing a lightweight solution for COVID-19 diagnosis. As such, this model represents a highly competitive option for researchers seeking to implement an efficient and effective solution for diagnosing COVID-19, while simultaneously minimising computational requirements and costs.

On the other hand, to understand how our proposed model performs the classification, Guided Grad-cam class activation visualisation was used at the last convolutional layer of the model – the layer followed by a (256) flattening layer (Zhang et al. 2021). Figure 10 shows the Grad-cam visualisation for a slice in the validation set. The slice belongs to a COVID case and was correctly classified by the model. The outputs for the correct and incorrect classifications are adapted to the input image. They show that the model pays attention to:

- the lung area, and
- the posterior and anterior walls (with the anterior walls getting very strong attention values).

The results, however, show that the model also pays attention to areas outside of the lung, mainly to the patient’s sitting table. For further understanding of the model’s prediction mechanism, correctly and incorrectly classified slices were checked, and we can observe a similar attention distribution on the COVID cases incorrectly classified and Non-COVID cases (correctly and incorrectly classified slices) as well.

As for the slice-level decision, the proposed model can sometimes incorrectly predict the uppermost and the lowermost slices as Non-COVID (specifically, 20 out of 24 extreme slices in the validation partition are misclassified). These extreme slices correspond to the anatomical regions where COVID involvement is not seen and therefore can be considered the least representative slices for the diagnosis.

| Network’s Depth       | No. of Model Parameters | Increase Rate in Complexity | Validation Accuracy | Improvement Rate in Accuracy |
|-----------------------|-------------------------|-----------------------------|---------------------|-------------------------------|
| 3 Hidden Layers       | 23744                   | -                           | 81.7%               | -                             |
| 4 Hidden Layers (standard) | 98122         | x4.132                      | 84.5%               | x0.034                        |
| 5 Hidden Layers       | 394304                 | x16.606                     | 84.8%               | x0.038                        |
of the illness. Figure 11 shows exemplary slices that are correctly classified by our proposed model, while Figure 12 depicts exemplary slices that are incorrectly classified where the extreme slices can be observed.

### 5.2 Patient-level results

To obtain a patient-level diagnosis from slice-level decisions different class probability thresholds (for slice prediction) varying in the range of [0, 1] were tried as explained in Section 3.5, and the corresponding macro F1 scores were compared. The majority voting was used at the patient level (for CT prediction). As observed in Figure 13, the model achieves the highest macro F1 score with a class probability threshold of 0.40, followed by a class probability threshold of 0.15. The validation accuracies at the patient level when using the mentioned thresholds are 88.5% and 87.7%, respectively. With that, the results demonstrate that at patient-level a class probability threshold of 0.40 gives the best performance when used with a majority voting in terms of macro F1 score. The patient-level macro F1 score achieved using that class probability threshold with the proposed methodology reaches 0.882 on the validation set. The resulting macro F1 score of the proposed model comfortably exceeds that of the baseline model on the validation set. The baseline score on the COV19-CT-DB validation set is 0.70, as reported by Kollias et al. (2021b).
Figure 12. Examples of incorrectly classified slices from COVID (right) and Non-COVID (left) cases.

Figure 13. Different class probability thresholds (horizontal axis) for making predictions at the patient level and the corresponding macro F1 scores.

Table 3. Confusion matrix on patient-level decision using a voting threshold of 0.40. Columns refer to actual cases while rows display predictions of the proposed model.

| Actual  | Predicted | COVID | Non-COVID |
|---------|-----------|-------|-----------|
| COVID   | 135       | 13    |           |
| Non-COVID | 30       | 196   |           |
In general, the model misclassifies 13 Non-COVID cases out of 209, and 30 COVID ones out of 165. Class-specific macro F1 scores of the proposed method are 0.86 for the COVID class and 0.90 for the Non-COVID. Table 3 shows the confusion matrix of the proposed method at the patient level for the best threshold value.

Although the misclassification of 13 Non-COVID cases out of 209 is less problematic than a misclassification of 30 COVID cases out of 165, the model aimed mainly at increasing the quantitative results.

Additionally, to validate the results the method was tested on the test partition of the COV19-CT-DB database (unseen images). On an unseen dataset of images, the method’s performance exceeded those of the baseline and other works. Within the context of the MIA-COVID19 competition, the teams were provided with a test partition of images. The model achieved a 0.82 macro F1 score, with a 0.96 F1 score for Non-COVID and a 0.68 F1 score for COVID. This score is above the baseline, which is a 0.67 macro F1 score.

Our proposed method did not only exceed the baseline macro F1 score (0.67) but also outperformed other alternatives that entered the competition and reported accuracies on COV19-CT-DB’s test partition as reported on the MIA-COVID19 competition’s leaderboard. Table 4 compares our model to other alternatives on the test partition (unseen images) of COV19-CT-DB. Our team is named ‘IDU-CVLab’ and the code was developed using Python.3 Our proposed solution outperforms almost all state-of-the-art alternatives with an exception of the CCAT and DWCC by Hsu et al. (2021). The solution by Hsu et al. (2021) is based on ensembled deep learning models; firstly, a proposed Deep Wilcoxon signed-rank test for COVID-19 classification adopts nonparametric statistics for deep learning to find a series of slices with the most significant symptoms in CT scan. Secondly, a Convolutional CT scan Aware model is used to further explore the intrinsic features in temporal and spatial dimensions. Finally, a three-layer perceptron is used as a classifier to take final diagnosis decisions. While our method uses one simple deep neural network, the ‘CCAT and DWCC’ method uses more. This makes our solution much less complex while achieving good performance.

Table 4. Comparison of the proposed method with the baseline and other state-of-the-art models on the COV19-CT-DB test partition (unseen images).

| The Method                                      | Macro F1 |
|-------------------------------------------------|----------|
| ResNet50-GRU (Baseline model) (Kollias et al. 2021b) | 0.67     |
| A hybrid deep learning framework (CTNet) (Liang et al. 2021) | 0.78     |
| Custom Deep Neural Network (Trinh and Van Nguyen 2021) | 0.78     |
| Our proposed methodology (IDU-CVLab)             | 0.82     |
| CCAT and DWCC (Hsu et al. 2021)                  | 0.88     |

6. Conclusion and discussion

This paper proposes a solution for COVID-19 diagnosis using deep learning and image processing techniques. The adopted CNN model architecture was trained, validated, and tested on the recently carefully annotated COV19-CT database. Proposing a simple and efficient solution was the main aim of this study. CT scan image process techniques aimed at non-representative slice removal. The resulting slices were used as input to train a simple and symmetric CNN model where the validation accuracy at the slice level was monitored for several architectural modifications. To achieve that, different numbers of model layers and different training batch sizes were tested. The proposed CNN method was tested with different class probability thresholds to make a diagnosis decision.

Next and to achieve predictions at the patient level, majority voting on each CT/for each patient scan was used. The method achieved a macro F1 score exceeding the baseline score and other alternatives on the test partition of the COV19-CT-DB database. More complex modelling techniques do not reach as high macro F1 scores as the CNN model trained. With that, we encourage researchers, programmers, and otherwise to consider simpler and from-scratch deep learning models with appropriate modifications to suit the task.

One gap in this study is that the non-representative slice removal threshold may require fine-tuning for other CT datasets. This threshold should guarantee that the CT scan is not fully void of any 2D slice to allow our proposed model to make diagnosis decisions at the patient level.

Another gap in this work is using majority voting after non-representative slice removal in each CT scan. Since the removal of the slices is subject to a threshold-biased, then the majority vote should not be the best method to make patient-level diagnosis decisions. This gap could be closed by performing further tuning between the slice removal approach and the patient-level voting methods. Some examples of closing such a gap can be to find a synchronised dynamic between the number and the type (uppermost, central, or lowermost) of slices left in the CT and the voting method at the patient level. Finding such a synchronisation should maximise the performance of the proposed method for each diagnostic decision on a CT scan/patient.

Even though the rectangular region selection for slice processing along with other slice processing and hyperparameters tuning techniques improved the method’s performance or accuracy, using a manually fixed-size rectangular shape to crop the slices can still be considered another gap of this presented methodology. The reason is that this slice cropping approach will perform poorly in localising the lung region in the lowermost and uppermost slices in the CT image. Further improvement in limiting the region of interest with the lung volumes instead of processing the whole CT scan will be a promising approach that can be realised through segmenting lung parenchyma prior to classification. This approach could ultimately improve the diagnostic performance of the proposed method.

One more gap in this research is in focusing on the noise suppression of slices. In an attempt to reach an appropriate
threshold for non-representative slice removal, the Gaussian filter is used to suppress the noise in the slices. However, other filters could have achieved better results in reaching more appropriate image enhancement results and therefore finding a better threshold for non-representative slices removing. Further improvement could include comparing different noise-compressing filters or techniques to reach better performance.

Finally, we should mention utilising Service Oriented Networks (SONs) for reliable and effective COVID-19 diagnosis from CT images. The use of automated tools for COVID-19 diagnosis from CT images has the potential to significantly improve personalised medicine in clinical environments. However, for these tools to be effective, they must be reliable and consistent. A SON architecture is a suitable solution to ensure the continuity and reliability of these tools. An example is presented by Conti et al. (2021). Their work is an architecture proposed for SON using a bio-inspired management approach based on the functioning of cell metabolism. The nodes in the network communicate with each other by stimulation or suppression chains, similar to the processes in metabolic networks. The proposed methodology aims to improve network reliability and robustness for maintaining service continuity. The paper also outlines a case study for a healthcare imaging application using the proposed framework, which achieved a 4x speed-up factor compared to the software counterpart. The results show the effectiveness of the proposed computational framework for SON analysis. With that in mind, the proposed architecture for COVID-19 image classification, based on the SON architecture, could be a valuable tool for clinicians in the diagnosis and treatment of COVID-19. Further research can be done to optimise and improve the proposed architecture to meet the demands of clinical environments.

Notes
1. 512x512 was the size of the original images in COVID-19-CT-DB database. Cropped images are of size 227x300.
2. https://mlearn.lincoln.ac.uk/mia-cov19d/
3. https://github.com/IDU-CVLab/COV19D

Acknowledgements
The authors acknowledge the work of all the medical staff and others who manually annotated the images in the COV19-CT-DB database and shared them in a relatively big dataset.

Disclosure statement
No potential conflict of interest was reported by the authors.

Funding
This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

ORCID
Kenan Morani http://orcid.org/0000-0002-4383-5732
D. Unay http://orcid.org/0000-0003-3478-7318

References
Ali Ahmed SA, Yavuz MC, Şen MU, GÜlsen F, Tutar O, Korkmazer B, Samanci C, Şirloş S, Hamid R, Eryürekli AE, et al. 2022. Comparison and ensemble of 2D and 3D approaches for COVID-19 detection in CT images. Neurocomputing. 488:457–469. doi:10.1016/j.neucom.2022.02.018.
Anwar T. 2021. COVID19 Diagnosis using AutoML from 3D CT scans. [accessed 2023 Feb 2]. doi:10.36227/techxvr.14914851.v1.
Chen L, Fei H, Xiao Y, He J, Li H 2017. Why batch normalization works? a buckling perspective. In: 2017 IEEE International Conference on Information and Automation (ICIA); Macao, China. IEEE. p. 1184–1189.
Conti V, Milletello C, Rundo L, Vitabile S. 2021. A novel bio-inspired approach for high-performance management in service-oriented networks. IEEE Trans Emerg Top Comput. 9(4):1709–1722. doi:10.1109/TETC.2020.3018312.
El Asnaoui K, Chawki Y. 2021. Using X-ray images and deep learning for automated detection of coronavirus disease. J Biomol Struct Dyn. 39 (10):3615–3626. doi:10.1080/07391102.2020.1767212.
Fang R, Guo J, Xian B. 2022. How machine learning applied in COVID-19 prevention & control. J Phys Conf Ser. 2386(1):012033. doi:10.1082/1742-6596/2386/1/012033.
Feng-D P, Zhou T, Ji G-P, Zhou Y, Chen G, Fu H, Shen J, Shao L 2020. InfNet: automatic COVID-19 lung infection segmentation from CT images. IEEE Trans Med Imaging. 39(8):2626–2637. doi:10.1109/TMI.2020.2996645.
Hassanin A, Tu VT, Curaudue M, Csorba G. 2021. Inferring the ecological niche of bat viruses closely related to SARS-CoV-2 using phyleogeographic analyses of Rhinolophus species. Sci Rep. 11(1):14276. doi:10.1038/s41598-021-9738-z.
He J-L, Luo L, Luo Z-D, Lyu J-X, Ng M-Y, Shen X-P, Wen Z. 2020. Diagnostic performance between CT and initial real-time RT-PCR for clinically suspected 2019 coronavirus disease (COVID-19) patients outside Wuhan, China. Respir Med. 168(105980):105980. doi:10.1016/j.rmed.2020.105980.
Hsu C-C, Chen G-L, Wu M-H. 2021. Visual Transformer with statistical test for COVID-19 classification. arXiv (eessvl).
Huang L, Han R, Ai T, Yu P, Kang H, Tao Q, Xia L. 2020. Serial quantitative chest CT assessment of COVID-19: a deep learning approach. Radiol Cardiothoracic Imaging. 2(2):e200075. doi:10.1082/rci.20200075.
Hussain Z, Gimenez F, Yi D, Rubin D. 2017. Differential data augmentation techniques for medical image classification tasks. AMIA Annu Symp Proc. 2017:979–984.
Islam MM, Karray F, Alhajj R, Zeng J. 2021. A review on deep learning techniques for the diagnosis of novel Coronavirus (COVID-19). IEEE Access. 9:30551–30572. doi:10.1109/ACCESS.2021.3058537.
Kılıçarslan S, Adem K, Çelik M. 2021. An overview of the activation functions used in deep learning algorithms. J New Results In Sci. 10(3):75–88. doi:10.5418/jrnls.1011739.
Kollia D, Arsenos A, Soukissian L, Kollia S. 2021a. MIA-COV19D: COVID-19 detection through 3-D chest CT image analysis. In: 2021 IEEE/CFV International Conference on Computer Vision Workshops (ICCVW); Montreal, Canada. IEEE. p. 537–544.
Kollia D, Bouas N, Vlaxos Y, Brilakis V, Seferas M, Kollia I, Sukissian L, Wingate J, Kollia S. 2020. Deep transparent prediction through latent representation analysis. arXiv [cs.LG].
Kollia D, Tagaris A, Stafylapatis A, Kollia S, Tagaris G. 2018. Deep neural architectures for prediction in healthcare. Complex Intell Syst. 4 (2):119–131. doi:10.1007/s40747-017-0064-6.
Kollia D, Vlaxos Y, Seferas M, Kollia I, Sukissian L, Wingate J, Kollia S. 2021b. Transparent adaptation in deep medical image diagnosis. In: Trustworthy AI - Integrating learning, optimization and reasoning. Cham: Springer International Publishing; pp. 251–267. 10.1007/978-3-030-73959-1_22.
Liang S, Zhang W, Gu Y. 2021. A hybrid and fast deep learning framework for Covid-19 detection via 3D chest CT images. In: 2021 IEEE/CFV International Conference on Computer Vision Workshops (ICCVW); Montreal, Canada. IEEE.
Mary Shyny H, Chitra E. 2022. A comparative study of x-ray and ct images in COVID-19 detection using image processing and deep learning techniques. Comput Methods Programs Biomed. Update. 2 (100054):100054. doi:10.1016/j.cmpbup.2022.100054.
Mizrahi B, Shilo S, Rossman H, Kalkstein N, Marcus K, Barer Y, Keshet A, Shamir-Stein N, Shalev V, Zohar AE, et al. 2020. Longitudinal symptom dynamics of COVID-19 infection. Nat Commun. 11(1):6208. doi:10.1038/s41467-020-20053-y.

Morozov SP, Andreychenko AE, Blokhin IA, Gelezhe PB, Gonchar AV, Pavlov NA, Chernina VY, Nikolaev AE. 2020. MosMedData: data set of 1110 chest CT scans performed during the COVID-19 epidemic. Digital Diagnostics. 1(1): 49–59. doi:10.17816/dd46826.

Nayak SR, Nayak DR, Sinha U, Arora V, Pachori RB. 2021. Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: a comprehensive study. Biomed Signal Process Control. 64 (102365):102365. doi:10.1016/j.bspc.2020.102365.

Opitz J, Burst S. 2019. Macro F1 and Macro F1. arXiv [cs.Lg].

Panwar H, Gupta PK, Siddiqui MK, Morales-Menendez R, Singh V. 2020. Application of deep learning for fast detection of COVID-19 in X-Rays using nCovnet. Chaos Solitons Fractals. 138(109944):109944. doi:10.1016/j.chaos.2020.109944.

Rahman AM, Azhir E, Naserbakht M, Mohammadi M, Aldalwie AHM, Majeed MK, Taher Karim SH, Hosseinzadeh M. 2022. Automatic COVID-19 detection mechanisms and approaches from medical images: a systematic review. Multimed Tools Appl. 81(20):28779–28798. doi:10.1007/s11042-022-12952-7.

Rosenthal PJ. 2020. The importance of diagnostic testing during a viral pandemic: early lessons from novel Coronavirus disease (COVID-19). Am J Trop Med Hyg. 102(5):915–916. doi:10.4269/ajtmh.20-0216.

Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D 2017. Grad-CAM: visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV); Venice, Italy. IEEE. p. 618–626.

Takahashi K, Yamamoto K, Kuchiba A, Koyama T. 2022. Confidence interval for micro-averaged F1 and macro-averaged F1 scores. Appl Intell. 52(5):4961–4972. doi:10.1007/s10489-021-02635-5.

Talbi F, Alim F, Seddiki S, Mezzah I, Hachemi B 2015. Separable convolution gaussian smoothing filters on a xilinx FPGA platform. In: Fifth International Conference on the Innovative Computing Technology (INTECH 2015); Galicia, Spain. IEEE. p. 112–117.

Tang H, Ortis A, Battista S. 2019. The impact of padding on image classification by using pre-trained convolutional neural networks. In: Lecture notes in computer science. Cham: Springer International Publishing; pp. 337–344. 10.1007/978-3-030-30645-8_31

Tan W, Liu J. 2021. A 3D CNN network with BERT for automatic COVID-19 diagnosis from CT-scan images. arXiv [eess.l]. [accessed 2023 Feb 2]. http://arxiv.org/abs/2106.14403.

Trinh QH, Van Nguyen M. 2021. Custom deep neural network for 3D covid chest CT-scan classification. arXiv [eess.l].

Waleed Salehi A, Baglat P, Gupta G. 2020. Review on machine and deep learning models for the detection and prediction of Coronavirus. Mater Today. 33:3896–3901. doi:10.1016/j.matpr.2020.06.245.

Wang S-H, Fernandes SL, Zhu Z, Zhang Y-D. 2022. AVNC: attention-based VGG-style network for COVID-19 diagnosis by CBAM. IEEE Sens J. 22(18):17431–17438. doi:10.1109/JSEN.2021.3062442.

Wynants L, Van Calster B, Collins GS, Riley RD, Heinze G, Schuit E, Bonten MMJ, Dahly DL, Damen JAA, Debray TPA, et al. 2020. Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal. BMJ. 369:m1328. doi:10.1136/bmj.m1328.

Xiong Y, Sun D, Liu Y, Fan Y, Zhao L, Li X, Zhu W. 2020. Clinical and high-resolution CT features of the COVID-19 infection: comparison of the initial and follow-up changes: comparison of the initial and follow-up changes. Invest Radiol. 55(6):332–339. doi:10.1097/RLI.0000000000000674.

Zhang Y, Hong D, McClement D, Oladosu O, Pridham G, Slaney G. 2021. Grad-CAM helps interpret the deep learning models trained to classify multiple sclerosis types using clinical brain magnetic resonance imaging. J Neurosci Methods. 353(109098):109098. doi:10.1016/j.jneumeth.2021.109098.

Zhang Y, Satapathy SC, Zhu L-Y, Gorriz JM, Wang S. 2022. A seven-layer convolutional neural network for chest CT-based COVID-19 diagnosis using stochastic pooling. IEEE Sens J. 22(18):17573–17582. doi:10.1109/JSEN.2020.3025855.