International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.1 (2025) pp. 816-826
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Deep Learning Based COVID-19 Detection Using Computed Tomography Images

Abdullah Asım Yılmaz[1]*, Ömer Sevinç[2]

[1]Atılım University, Engineering Faculty, Computer Engineering Department, 06830, Ankara- Türkiye
* **Corresponding Author Email:** abdullah.yilmaz@atilim.edu.tr - **ORCID:** 0000-0002-3014-609X

[2]Ondokuz Mayıs University, Vezirköprü Vocational School, Computer Programming Department, 55900, Samsun-Türkiye
Email: osevinc@omu.edu.tr - **ORCID:** 0000-0003-0006-1682

Article Info:

DOI: 10.22399/ijcesen.963
Received : 04 December 2024
Accepted : 05 February 2025

Keywords :

COVID-19,
Deep neural network,
CT images,
Deep learning.

Abstract:

The infectious coronavirus disease (COVID-19), seen in Wuhan city of China in December 2019, led to a global pandemic, resulting in countless deaths. The healthcare sector has become extensively use of deep learning (DL), a method that is currently quite popular. The aim of this study is to identify the best and most successful deep learning model and optimizer approach combination for COVID-19 diagnosis. For this reason, several DL methods and optimizer techniques are tested on two comprehensive public data set to select the best DL model with optimizer technique. A variety of performance evaluation metrics, including f-score, precision, specificity, and accuracy, were used to assess the models' effectiveness. The experimental results show that the most suitable and effective architecture is DenseNet-201 in the network comparison, which achieved a 98% accuracy rate using the AdaGrad optimizer and 200 iterations.

1. Introduction

In recent years, the application of high technology in the health industry has increased significantly, especially in the field of diagnosis and detection of diseases. These advances have generated considerable interest, especially diagnose and detect respiratory disease COVID-19 caused by new coronavirus SARS-CoV-2. Changes such as turning into a global epidemic of COVID-19, the rapid spread of the virus and the increase or decrease of various clinical symptoms and mortality emphasize the need for timely medical care to avoid overloading the health system. Therefore, the development of urgent and effective methods for accurate and rapid detection of COVID-19 has become an important need. The diagnosis and detection of COVID-19 requires identification of a human disease or condition based on its signs and symptoms., as well as the identification of various biomarkers or patterns from a viral infection. This is usually accomplished by analyzing data from medical imaging such as chest X-rays, computed tomography (CT), clinical symptoms, and laboratory tests [1-5].

Artificial Intelligence (AI) has seen rapid and significant advancements in tackling complex issues across various fields, including economics [6], medicine [7], and engineering [8-10] in recent years. Deep Learning (DL), a key area within AI, has gained considerable popularity, particularly in medical applications. Traditionally, many tasks were performed manually by doctors, but with the advent of DL, these time-intensive processes are becoming more efficient [11, 12]. This progress has spurred significant interest in developing DL-based methods for diagnosing COVID-19 utilizing both CT images [13] and X-ray [14].

This study aims to determine optimal model for diagnosis of COVID-19 by utilizing DL architectures with varying optimizers on COVID-CT [15] and SARS-CoV-2 CT-scan [16] datasets. The architectures examined include ResNet-50 [17], UNet [18], SqueezeNet [19], AlexNet [20], MobileNetV2 [21], and DenseNet-201 [22]. The optimizers tested were Adaptive Moment Estimation (ADAM) [23], Adaptive Gradient Algorithm (AdaGrad) [24], Stochastic Gradient Descent (SGD) [25], and Root Mean Square Propagation (RMSProp) [26].

The remainder of this paper is structured as follows: A brief summary of relevant work on COVID-19 detection is given in Section 2. The COVID-19 datasets used, the DL model frameworks that were employed, and the optimizer techniques are all given in detail in Section 3. Section 4 presents experimental results and discusses the success of the models. The paper is finally brought to a close in Section 5 with a discussion of future work and a summary.

2. Related Work

Wang et al. [27] suggested a model named DeCoVNet to classify COVID-19 cases and localize lesions using chest CT scans. The DeCoVNet model integrates a progressive classifier with adaptive, stridden, and global maximum pooling layers alongside convolution and fully connected layers. The DeCoVNet also incorporates a weakly supervised lesion localization algorithm, combining 3d connected component [28] and class activation mapping [29] activation methods for improved lesion detection. In addition, Wang et al. employed a pre-trained UNet [18] to segment lung area. After being trained on 499 CT volumes and tested on 131 CT volumes, this model produced an accuracy of 0.901 at a probability threshold of 0.5, a negative predictive value of 0.982, and a positive predictive value of 0.840. The model's receiver operating characteristic (ROC) curve (ROC AUC) was 0.959 and its recision-recall (PR) curve (so-called area under curve (AUC)) was 0.975. DeCoVNet showed significantly better performance than traditional classifiers, exceeding their accuracy by at least 10% according to the study's findings. While it achieved impressive results, it still fell short of matching the performance of human experts. The model also provided a high lesion localization hit rate of 68.5% without using lesion annotations, highlighting its potential effectiveness in clinical settings.

Jaiswal et al. [30] suggested a deep transfer learning model based on DenseNet201 for classifying COVID-19 infected patients utilizing chest CT images. The model leverages DenseNet201's architecture, which includes dense connections and feature reuse, making it highly efficient and effective even with smaller datasets. The DenseNet201 is pretrained on ImageNet dataset, and its weights are fine-tuned for the COVID-19 classification task. The proposed model's architecture includes dense blocks separated by transition layers that include batch normalization, 2x2 average pooling, and 1x1 convolution layers, enhancing network's connectivity and performance. Experimental results indicate that model achieves high classification accuracy of 96%, with a notable reduction in false negative and false positive rates, making it a reliable alternative to rapid COVID-19 testing kits.

A DL-based CT diagnosis method was created by Song et al. [31] in order to locate the primary lesions and identify the pneumonia-causing COVID-19. The proposed model, named details relation extraction neural network (DRENet), utilizes a pre-trained ResNet50 [17] architecture enhanced with a Feature Pyramid Network and an attention module to extract and analyze important features from CT images. The model was validated and trained on a dataset comprising CT images from 88 COVID-19 patients, 100 bacterial pneumonia patients and 86 healthy individuals. The results demonstrated that DRENet achieved high accuracy, with a precision of 79%, a recall of 96%, and an AUC of 95% in differentiating COVID-19 from bacterial pneumonia. When discriminating COVID-19 from other conditions, the model achieved a precision of 86% and a recall of 93%. Additionally, the model was able to effectively highlight main lesion features, such as ground-glass opacity, aiding visual diagnosis by medical professionals.

The goal of He et al. [32] was to create deep learning techniques that might yield excellent diagnosis accuracy rates even in the case of sparse training CT samples. They suggested a self-supervised transfer learning (Self-Trans) method. To acquire robust and unbiased feature representations, contrastive self-supervised learning was paired with transfer learning to lower the danger of over-fitting. Additionally, they made a public data set with hundreds of CT scans that tested positive for COVID-19 available. According to the results obtained from the study, the Self-Trans method showed superior performance, achieving an AUC of 94% and an f-score of 85% in diagnosing COVID-19 from CT scans, despite being trained on only a few hundred CT scans.

Wua et al. [33] presented a multi-view fusion model based on DL to improve screening of COVID-19 disease. The proposed model incorporates ResNet50 architecture, processing sagittal, axial, and coronal views of the lung regions segmented from CT scans. The datasets used in the study were split into test, training, and validation sets with a distribution of 50, 395, and 50 cases, respectively. Here, RMSprop optimizer was used to train the model with a batch size of 4 and a learning rate of $1\times10-5$. The study demonstrated that the multi-view model significantly outperforms the single-view model, achieving higher AUC, accuracy, and sensitivity across training, test, and validation sets. The multi-view model achieved sensitivity of 0.823, accuracy of 0.833, and an AUC of 0.905 in the training set, with comparably strong results in the validation and test sets, highlighting its potential for improving COVID-19 diagnosis accuracy.

3. Material and Method

Lately, deep learning methods have shown exceptional proficiency with optimal performance in the domains of image processing and computer vision. This success of deep learning methods later found a place in the health sector and began to be widely used.

The objective of this study is to classify CT scans into two groups: normal and COVID-19 positive. To accomplish this classification, it is proposed six DL algorithms with different optimizers. These are ResNet-50 [17] , UNet [18], SqueezeNet [19], AlexNet [20], MobileNetV2 [21], and DenseNet-201 [22]. The details of proposed methodology utilizing DL methods for COVID-19 diagnosis using CT images is provided in Section 3.2 . The suggested method is shown in Figure 1.

This section's remaining content is organized as follows: While the COVID-19 datasets are described in full in Section 3.1. The suggested methodology using DL methods with different optimizers for COVID-19 detection is given in Section 3.2. Sections 3.3 and 3.4 thoroughly explain the deep learning methods and optimizer techniques utilized, respectively.

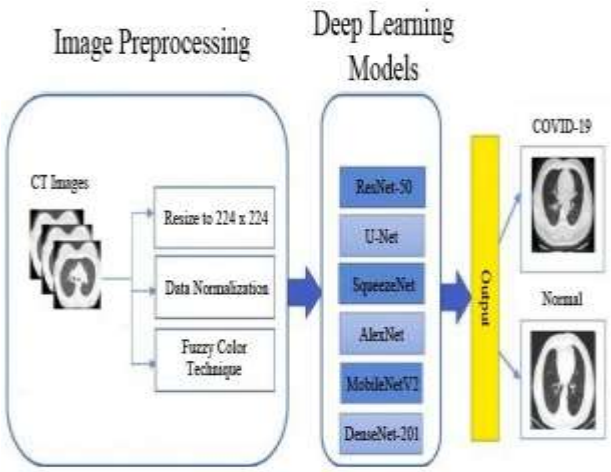

Figure 1. *The overview of the proposed model*

3.1. Datasets

COVID-CT [15] and SARS-CoV-2 CT-scan [16] datasets were used in this study. Examples of CT scans from this data collection are shown in Figure 2. 658 of these CT images were utilized for testing, while the remaining 2636 were employed for training. The COVID-CT dataset [15] comprise 349 COVID-19 positive CT images obtained from 216 patients. The dataset also includes metadata for 169 patients with age information and 137 with gender information. Here, the female and male patient numbers are 51 and 86 respectively. For negative training examples, non-COVID-19 CT images were sourced from multiple databases, including MedPix, LUNA, Radiopaedia, and PubMed Central, resulting in a total of 463 images from 55 patients. Additionally, the dataset for validation and testing includes original CT images not extracted from papers, with the test set comprising 168 non-COVID-19 images from 20 patients and 173 COVID-19 images from 4 patients, and the validation set containing 64 non-COVID-19 images from 39 patients and 88 COVID-19 images from 4 patients. The 2,482 CT scans in the SARS-CoV-2 CT-scan dataset [16] include 1,252 scans from COVID-19-infected patients and 1,230 images from non-infected individuals with other pulmonary conditions. These scans were taken from actual patients in Brazilian hospitals in Sao Paulo. The dataset was produced to aid in the improvement of AI techniques that use CT scan analysis to detect COVID-19 infections. Promisingly high accuracy and f-scores were obtained with the eXplainable Deep Learning approach (xDNN) applied to this dataset. The dataset and xDNN code are publicly available to encourage further research and development in this area.

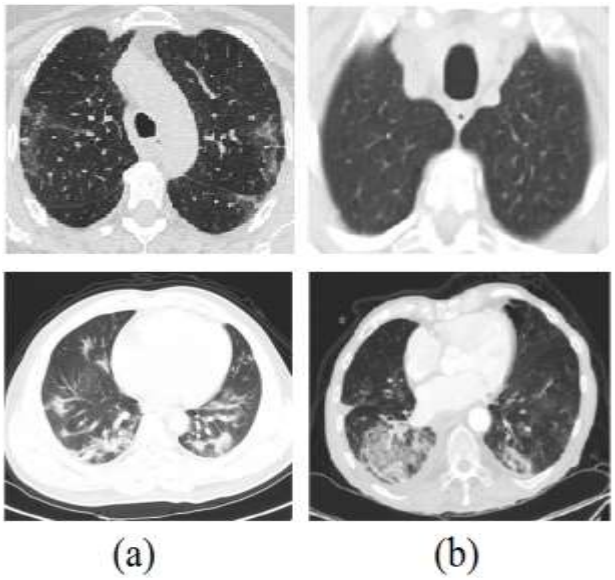

Figure 2. *Sample CT images from the COVID-CT and SARS-CoV-2 CT-scan datasets for normal cases (a) and COVID-19 patients (b).*

3.2. Methodology

The suggested methodology, which is illustrated in Figure 1, is designed to classify CT images. This method distinguishes COVID-19 virus-infected CT scans from healthy CT scans. First, COVID-19 data are collected from two extensive datasets. Second, the original datasets are subjected to the pre-processing stage. In order to eliminate the noise in the original photos, the original datasets were rebuilt

using Python code [34] with Fuzzy Color technique. The goal was to produce images with higher data quality. In addition, the datasets' images were standardized by changing their size to use 224-pixel-wide, 224-pixel-high images with three channels. Thirdly, the selected six deep learning models (ResNet-50 [17], UNet [18], SqueezeNet [19], AlexNet [20], MobileNetV2 [21], and DenseNet-201 [22]) were used as pre-trained network, and high- and low-level COVID-19 features are extracted using pre-trained networks. Finally, SVM method, which combines effective features to provide successful outcomes in numerous classes, was used to carry out the classification process. Here, the CT scan images are classified as COVID-19 positive and normal.

3.3. Optimizer Methods

The purpose of an optimizer is to minimize loss, which is the discrepancy between predicted and actual values. Therefore, the method of optimizer that is selected is crucial for study. This study employs four distinct optimizer methods: Root Mean Square Propagation (RMSProp), Stochastic Gradient Descent (SGD), Adaptive Gradient Algorithm (AdaGrad) and Adaptive Moment Estimation (ADAM).

The (ADAM) optimizer [23] is a widely used adaptive learning rate optimization algorithm in deep learning, designed to speed up convergence and handle noisy gradients. It combines the advantages of two methods: AdaGrad, which adapts learning rates for individual parameters, and RMSprop, which scales gradients to maintain stability. ADAM calculates running averages of both the squared gradient and the gradient, allowing for adaptive learning rates that adjust during training. This results in more efficient training, especially for complex models and large datasets, with minimal hyperparameter tuning needed for effective performance. The ADAM updating rule is provided in the equation 1 [35-38]:

$$Q_{t+1} = Q_t - \frac{\eta}{\sqrt{\hat{\vartheta}_t} + \epsilon} \hat{m}_t \qquad (1)$$

The AdaGrad optimizer [24] is designed to adjust the learning rate for each parameter individually, improving optimization efficiency, especially for sparse data. Its key feature is scaling the learning rate based on the historical accumulation of squared gradients. This causes frequently updated parameters to have smaller learning rates, while infrequent updates retain larger learning rates. AdaGrad's per-parameter learning rate adaptation helps models converge faster for tasks like natural language processing or image recognition with sparse features. However, it can result in overly small learning rates over time, limiting its effectiveness for longer training periods.

The SGD optimizer [25] is a simple and widely used optimization method in machine learning, particularly for deep learning. Its purpose is to minimize the loss function by updating model parameters in the direction of the negative gradient of the loss, one data sample at a time. Unlike traditional gradient descent, which computes the gradient for the entire dataset, SGD updates the parameters more frequently by using smaller, random batches of data, which speeds up the training process. While this introduces more noise, leading to faster but less stable convergence, it often helps escape local minima and improves generalization. Equation 1 is used for parameter updating in the SGD:

$$\theta = \theta - \eta . \nabla \theta \; J(\theta; x, y) \qquad (2)$$

The RMSProp optimizer [26] is generated to improve the efficiency of gradient-based optimization by addressing issues related to learning rate decay in AdaGrad. Its purpose is to maintain a moving average of the squared gradients for each parameter, which helps normalize the updates and prevent the learning rate from becoming too small. RMSProp adjusts the learning rate dynamically based on recent gradient information, making it well-suited for non-stationary and online learning tasks. By keeping the learning rate steady, it helps models converge faster and more reliably, especially in scenarios where gradients vary significantly during training, such as in deep learning with complex data. The update rule is as follows for RMSProp [39]:

$$E[g^2] = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2 \qquad (3)$$
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

3.4. Optimizer Methods

DenseNet-201 [22] is a deep convolutional neural network (CNN) designed to improve efficiency and accuracy by connecting each layer to every other layer in a densely connected architecture. Unlike traditional models where each layer receives inputs from the previous layer and passes its output to the next, DenseNet-201 introduces direct connections between all layers, ensuring that feature maps are reused and gradients are more effectively propagated during backpropagation. This model consists of 201

layers organized into dense blocks, where each layer takes all preceding feature maps as input. DenseNet-201 uses a bottleneck and transition layers to reduce the number of parameters, making it computationally efficient despite its depth. The model demonstrated superior performance on benchmark datasets such as ImageNet, achieving a top-5 error rate of 5.44%, while requiring fewer parameters and less computation compared to other state-of-the-art networks. UNet [18] is a deep learning model designed specifically for biomedical image segmentation tasks. The UNet architecture is built upon a fully convolutional network and is composed of a contracting path that captures context and a symmetric expanding path that allows for precise localization. This structure enables the model to be trained end-to-end with a relatively small amount of annotated data, thanks to an effective data augmentation strategy that includes elastic deformations. The UNet's implementation, based on the Caffe framework, is optimized for speed, segmenting a 512x512 image in less than a second on a recent GPU. The model demonstrated superior performance on the ISBI cell tracking challenge 2015, achieving an average intersection over union of 77.5% and 92% values on DIC-HeLa and PhC-U373 datasets respectively, significantly outperforming previous best methods by large margins. AlexNet [20] was designed to classify high-resolution images from the ImageNet dataset, which includes over 1000 different classes and 1.2 million images. This proposed model comprises five convolutional layers followed by three fully connected layers, with a final 10000 way softmax layer to produce classification outputs. The model incorporates several innovations, such as the use of ReLU (Rectified Linear Units) for faster training, local response normalization, overlapping pooling, and dropout to reduce overfitting. This model was trained on two GPUs for improved computational efficiency and achieved a top-5 error rate of 17.0% and a top-1 error rate of 37.5% on ILSVRC-2010 dataset, significantly outperforming previous state-of-the-art methods. In ILSVRC-2012 competition, AlexNet achieved a winning top-5 error rate of 15.3%, demonstrating effectiveness of deep convolutional networks for large-scale image classification. SqueezeNet [19], a deep convolutional neural network architecture, presents conceived to obtain AlexNet-level accuracy on the ImageNet dataset with 50 times fewer parameters. SqueezeNet employs three primary design methods: delaying downsampling to maximize the spatial resolution in the convolution layers, decreasing the number of input channels to 3x3 filters using "squeeze" layers, and replacing 3x3 filters with 1x1 filters to reduce the number of parameters. The architecture consists of "Fire modules," which combine 1x1 and 3x3 filters to balance model size and accuracy. SqueezeNet demonstrates the potential of small CNNs by achieving similar accuracy to AlexNet but with a significantly reduced model size of 4.8 MB. Further compression techniques, such as Deep Compression, reduce SqueezeNet to as little as 0.47 MB, making it highly suitable for deployment on devices with limited memory, like FPGAs and autonomous vehicles, without loss of accuracy. This approach shows that compact models are feasible and opens up new possibilities for efficient CNN design and deployment.The ResNet-50 [17] model, a deep convolutional neural network, is designed to enable the training of very deep networks by addressing the vanishing gradient problem, which often hampers the performance of traditional deep neural networks. ResNet-50 consists 50 layers, including average pooling layer, maximum pooling layer, and 48 convolutional layers, built around concept of residual learning. In order to enable deeper networks without deterioration, the model incorporates "shortcut connections" that bypass one or more layers and enable the network to learn residual functions with reference to the inputs. This architecture utilizes residual blocks with a bottleneck design that reduces computational complexity and ensures efficient training. ResNet-50 achieved state-of-the-art performance in image classification tasks, particularly on ImageNet dataset, with a top-5 error rate of 5.25%, significantly outperforming previous models and proving the effectiveness of deeper architectures for large-scale visual recognition tasks. MobileNetV2 [21] is a deep learning architecture optimized for resource-constrained and mobile environments that advances state-of-the-art in performance across benchmarks and multiple tasks. MobileNetV2 features an "inverted residual with linear bottleneck" architecture, where shortcut connections are made between thin bottleneck layers while intermediate expansion layers use lightweight depthwise convolutions to introduce non-linearity. This design minimizes memory footprint by not fully materializing large intermediate tensors, which is crucial for mobile applications. The model is evaluated on tasks like VOC image segmentation, COCO object detection, and ImageNet classification, showing superior performance-to-computation ratios. MobileNetV2 achieves competitive results, such as a top-1 accuracy of 72.0% on ImageNet with only 3.4 million parameters and 300 million multiply-adds, and demonstrates its efficiency through the SSDLite variant for object detection, which is 20x more computationally efficient than previous models

while maintaining accuracy. The results highlight the architecture's effectiveness for mobile and real-time applications.

4. Results and Discussions

This section give the experiments carried out in this work and the results of these experiments. We first provide a detailed description of the experimental setup used for the classification tasks in Section 3.1. The results of the experiments conducted on the two datasets are then provided in Section 3.2, along with a discussion of the results.

Table 1. *The hyperparameters of suggested model*

Hyperparameter	**Value(s)**
Epoch Count	100, 200
Loss Type	Categorical cross entropy
Optimizer	ADAM, AdaGrad, SGD, RMSProp
Learning rate	0.01 - 0.0001
Dropout	0.5
Batch size	128
Validation, test, train split ratios	0.1:0.2:0.7

Table 2. *The evaluation metrics formulas*

Evaluation Metric	Equation
F-score	2*TP / (2*TP+FP+FN)
Sensitivity	TP / (TP+FN)
Accuracy	(TP+TN) / (TP+TN+FP+FN)
Specificity	TN / (TN+FP)

4.1. Experimental Setup

Our studied model was implemented utilizing Python scripting language, and experiments were conducted on a personal computer running Linux platform on a 64 GB RAM and 5.0 GHz Intel Core i9 13700H CPU. From the datasets, the test, training, and validation data were selected at random, and evaluation were completed separately. A total of 10% of the data was used for validation, 20% for testing, and 70% for training. The training procedure ran without GPU support for around fifty hours, stopping at three hundred epochs. In addition, while softmax is used in the output layer, Rectified Linear Unit (ReLU) [39] is used in the hidden layer. Table 1 displays the hyperparameters chosen for the experiments.

4.2. Experimental Results

The models were individually trained on the training dataset using each iteration value and optimizer. The models performance was tested after each training operation is carried out. Evaluation metrics including f-score, precision, specificity and accuracy were utilized to assess the performance of the architectures. The formulae in Table 2 were used to calculate these evaluation metrics, where FN and TN represent false negative and true negative, while FP and TP denote false positive and true positive. The models' performances are displayed in Table 3. The table's abbreviations for the f-score, precision, specificity, and sensitivity are fsc, pre, spe, and acc, respectively. Upon reviewing Table 3 and Table 4, DenseNet-201
achieved the highest accuracy rate at 98%, using the AdaGrad optimizer and 200 iterations. When analyzing the best results from other architectures, SGD and RMSprop were each used in 5 experiments, while the ADAM optimizer was used in 6. Among these, 8 results were obtained with 100 iterations and 12 with 200 iterations. AlexNet produced lowest accuracy of 31% with the SGD optimizer at 100 iterations. Moreover, AlexNet's highest accuracy of 67% was notably lower compared to other architectures. DenseNet model performed exceptionally well, with accuracy rates of 91% or higher.

The f-scores show that the best result is 0.97, while the lowest is 0.64. DenseNet-201 achieved an f-score of 0.97 using the AdaGrad optimizer with 200 iterations, whereas the AlexNet network scored 0.35 with the SGD optimizer and 100 iterations. DenseNet models consistently outperform others in f-scores.

For sensitivity values, the top architectures are DenseNet-201, UNet, and SqueezeNet, all achieving a sensitivity value of 0.99. DenseNet-201 reached this using the SGD and AdaGrad optimizers with 100 and 200 iterations, respectively. SqueezeNet produced the same result with SGD at 100 iterations and AdaGrad at 200 iterations. UNet achieved 0.99 using AdaGrad at 100 and 200 iterations, as well as RMSprop and ADAM at 100 iterations. In terms of specificity values, DenseNet-201 stands out as the best architecture, while AlexNet performed the worst. DenseNet-201 achieved a specificity of 0.97 with RMSprop at 200 iterations. On the lower end, AlexNet produced the lowest specificity value using the SGD optimizer with 200 iterations. For precision values, the highest result was achieved by DenseNet-201 with a score of 0.96, using 200 iterations and the RMSprop optimizer. The lowest precision score was 0.15, produced by ResNet-50 with the SGD optimizer and 200 iterations. Analyzing all precision values, ADAM was the optimizer used in most of the top-performing results, with 200 iterations. Once again, DenseNet models demonstrated superior performance in these results.

Table 3. *Performances of networks on COVID-CT dataset*

Networks	Optimizer	Iter	Acc	Sen	Spe	Pre	Fsc
AlexNet	AdaGrad	100	0.59	0.90	0.54	0.31	0.65
AlexNet	RMSprop	100	0.68	0.91	0.61	0.42	0.74
AlexNet	SGD	100	0.49	0.94	0.57	0.34	0.64
AlexNet	ADAM	100	0.66	0.94	0.58	0.35	0.73
AlexNet	AdaGrad	200	0.60	0.94	0.54	0.32	0.71
AlexNet	RMSprop	200	0.63	0.94	0.56	0.24	0.72
AlexNet	SGD	200	0.51	0.93	0.47	0.33	0.66
AlexNet	ADAM	200	0.67	0.92	0.60	0.43	0.76
UNet	AdaGrad	100	0.86	0.99	0.62	0.34	0.78
UNet	RMSprop	100	0.87	0.99	0.77	0.68	0.88
UNet	SGD	100	0.68	0.98	0.61	0.29	0.77
UNet	ADAM	100	0.84	0.99	0.75	0.64	0.87
UNet	AdaGrad	200	0.84	0.99	0.74	0.62	0.86
UNet	RMSprop	200	0.79	0.96	0.70	0.53	0.83
UNet	SGD	200	0.72	0.98	0.64	0.39	0.79
UNet	ADAM	200	0.88	0.97	0.73	0.59	0.85
SqueezeNet	AdaGrad	100	0.89	0.92	0.88	0.65	0.92
SqueezeNet	RMSprop	100	0.83	0.86	0.73	0.62	0.86
SqueezeNet	SGD	100	0.93	0.99	0.85	0.81	0.93
SqueezeNet	ADAM	100	0.91	0.97	0.93	0.92	0.96
SqueezeNet	AdaGrad	200	0.94	0.99	0.87	0.89	0.96
SqueezeNet	RMSprop	200	0.96	0.94	0.92	0.92	0.95
SqueezeNet	SGD	200	0.89	0.93	0.81	0.75	0.91
SqueezeNet	ADAM	200	0.94	0.89	0.94	0.95	0.93
MobileNetV2	AdaGrad	100	0.78	0.90	0.71	0.62	0.81
MobileNetV2	RMSprop	100	0.91	0.94	0.84	0.81	0.91
MobileNetV2	SGD	100	0.76	0.98	0.67	0.45	0.83
MobileNetV2	ADAM	100	0.84	0.98	0.71	0.55	0.84
MobileNetV2	AdaGrad	200	0.76	0.72	0.76	0.74	0.75
MobileNetV2	RMSprop	200	0.91	0.95	0.83	0.84	0.89
MobileNetV2	SGD	200	0.94	0.98	0.77	0.67	0.87
MobileNetV2	ADAM	200	0.94	0.97	0.88	0.87	0.93
ResNet-50	AdaGrad	100	0.64	0.95	0.59	0.36	0.70
ResNet-50	RMSprop	100	0.73	0.96	0.66	0.45	0.79
ResNet-50	SGD	100	0.54	0.98	0.62	0.36	0.69
ResNet-50	ADAM	100	0.71	0.97	0.63	0.37	0.78
ResNet-50	AdaGrad	200	0.65	0.96	0.59	0.35	0.75
ResNet-50	RMSprop	200	0.68	0.97	0.61	0.29	0.77
ResNet-50	SGD	200	0.56	0.98	0.54	0.38	0.71
ResNet-50	ADAM	200	0.72	0.97	0.65	0.48	0.81
DenseNet-201	AdaGrad	100	0.91	0.98	0.83	0.77	0.92
DenseNet-201	RMSprop	100	0.93	0.98	0.85	0.81	0.93
DenseNet-201	SGD	100	0.95	0.99	0.87	0.84	0.94
DenseNet-201	ADAM	100	0.95	0.97	0.92	0.92	0.96
DenseNet-201	AdaGrad	200	0.98	0.99	0.88	0.86	0.97
DenseNet-201	RMSprop	200	0.94	0.96	0.97	0.96	0.95
DenseNet-201	SGD	200	0.97	0.98	0.91	0.92	0.94
DenseNet-201	ADAM	200	0.95	0.96	0.94	0.93	0.96

Table 4. *Performances of networks on SARS-CoV-2 CT-scan dataset*

Networks	Optimizer	Iter	Acc	Sen	Spe	Pre	Fsc
AlexNet	AdaGrad	100	0.39	0.64	0.31	0.28	0.36
AlexNet	RMSprop	100	0.45	0.62	0.36	0.32	0.44
AlexNet	SGD	100	0.31	0.71	0.38	0.24	0.35
AlexNet	ADAM	100	0.45	0.66	0.35	0.27	0.46
AlexNet	AdaGrad	200	0.42	0.65	0.37	0.28	0.42
AlexNet	RMSprop	200	0.43	0.68	0.33	0.26	0.43

AlexNet	SGD	200	0.32	0.67	0.28	0.22	0.38
AlexNet	ADAM	200	0.47	0.66	0.37	0.28	0.48
U-Net	AdaGrad	100	0.66	0.69	0.42	0.25	0.49
U-Net	RMSprop	100	0.67	0.75	0.54	0.5	0.59
U-Net	SGD	100	0.48	0.68	0.42	0.21	0.48
U-Net	ADAM	100	0.66	0.75	0.54	0.48	0.57
U-Net	AdaGrad	200	0.64	0.73	0.51	0.44	0.56
U-Net	RMSprop	200	0.59	0.72	0.47	0.35	0.54
U-Net	SGD	200	0.53	0.73	0.42	0.28	0.52
U-Net	ADAM	200	0.68	0.71	0.50	0.41	0.56
SqueezeNet	AdaGrad	100	0.69	0.66	0.65	0.79	0.63
SqueezeNet	RMSprop	100	0.65	0.62	0.52	0.44	0.59
SqueezeNet	SGD	100	0.72	0.71	0.6	0.61	0.66
SqueezeNet	ADAM	100	0.73	0.72	0.74	0.69	0.71
SqueezeNet	AdaGrad	200	0.75	0.73	0.64	0.72	0.68
SqueezeNet	RMSprop	200	0.78	0.68	0.69	0.73	0.64
SqueezeNet	SGD	200	0.69	0.64	0.59	0.58	0.58
SqueezeNet	ADAM	200	0.74	0.65	0.72	0.76	0.65
MobileNetV2	AdaGrad	100	0.59	0.63	0.48	0.45	0.53
MobileNetV2	RMSprop	100	0.72	0.69	0.62	0.63	0.64
MobileNetV2	SGD	100	0.58	0.74	0.46	0.32	0.57
MobileNetV2	ADAM	100	0.64	0.72	0.48	0.37	0.55
MobileNetV2	AdaGrad	200	0.55	0.45	0.53	0.56	0.45
MobileNetV2	RMSprop	200	0.71	0.69	0.61	0.66	0.62
MobileNetV2	SGD	200	0.75	0.73	0.55	0.49	0.59
MobileNetV2	ADAM	200	0.74	0.71	0.65	0.69	0.64
ResNet-50	AdaGrad	100	0.45	0.66	0.39	0.28	0.44
ResNet-50	RMSprop	100	0.53	0.70	0.43	0.29	0.5
ResNet-50	SGD	100	0.41	0.65	0.39	0.25	0.45
ResNet-50	ADAM	100	0.51	0.71	0.42	0.21	0.49
ResNet-50	AdaGrad	200	0.45	0.68	0.36	0.22	0.44
ResNet-50	RMSprop	200	0.48	0.71	0.38	0.21	0.46
ResNet-50	SGD	200	0.36	0.62	0.31	0.15	0.42
ResNet-50	ADAM	200	0.52	0.72	0.42	0.34	0.52
DenseNet-201	AdaGrad	100	0.71	0.70	0.61	0.59	0.63
DenseNet-201	RMSprop	100	0.74	0.73	0.63	0.62	0.64
DenseNet-201	SGD	100	0.73	0.72	0.62	0.64	0.61
DenseNet-201	ADAM	100	0.75	0.71	0.69	0.74	0.67
DenseNet-201	AdaGrad	200	0.79	0.76	0.65	0.67	0.72
DenseNet-201	RMSprop	200	0.74	0.74	0.74	0.78	0.66
DenseNet-201	SGD	200	0.78	0.75	0.68	0.75	0.70
DenseNet-201	ADAM	200	0.75	0.69	0.71	0.76	0.67

When reviewing all the results in the table, DenseNet model emerge as the most successful architecture, while AlexNet proves to be less effective. Despite AdaGrad being the optimizer responsible for most of the top-performing results, the ADAM optimizer also showed strong performance overall. Figure 3 illustrates the normalized confusion matrixes for DenseNet-201, using AdaGrad as the optimizer, with 100 and 200 iterations.

5. Conclusion

People who have COVID-19 are likely to experience long-term lung damage, which may eventually lead to death. The purpose of this study was to differentiate between healthy patients and those who had COVID-19-related lung injury. This study examined how well DL models with various optimizers performed in identifying COVID-19 disease from CT images using various optimizers in various DL architectures. Furthermore, iteration number was configured at two distinct values, 100 and 200. While ADAM, AdaGrad, SGD and RMSProp were used as optimizers, ResNet-50, UNet, SqueezeNet, AlexNet, MobileNetV2, and DenseNet-201 were utilized as DL models in this study. The efficiency of DL models with different optimizers was observed. F-score, precision, specificity and accuracy were used as evaluation metrics. The results showed that AlexNet was the less successful architecture, whereas DenseNet-201 was fairly successful.

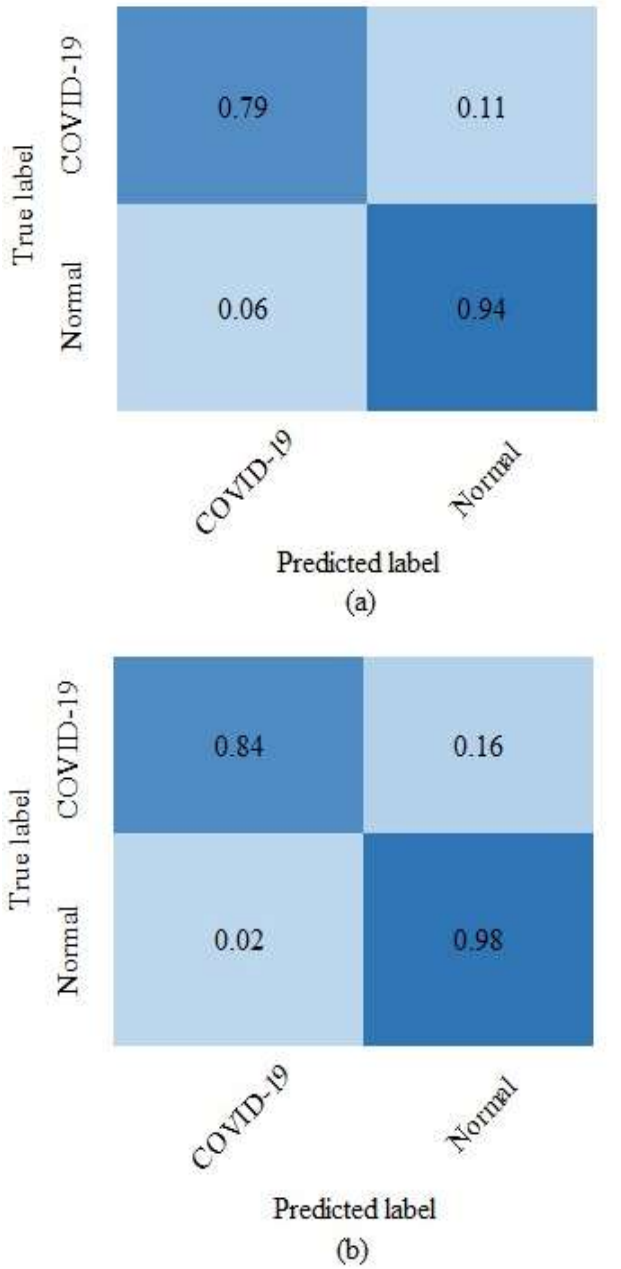

Figure 3. *Confusion matrixes obtained by the DenseNet-201 model and AdaGrad optimizer after 100 (a) and 200 epochs (b).*

The bulk of the top values in evaluation metrics contain the AdaGrad optimizer, even if ADAM is the better optimizer for the majority of architectures' own best results. In future work, we plan to compare the performances of more DL models on different databases and employ metaheuristic techniques for hyperparameter optimisation. Additionally, it is aimed to perform deep learning-based analyses utilizing data images of other organs affected by the virus. COVID-19 is studied and reported [40-43].

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] IS. Minaee, R. Kafieh, M. Sonka, S. Yazdani, and G. J. Soufi. (2020). Deep-covid: Predicting covid-19 from chest X-ray images using deep transfer learning. *Med. Image Anal*., 65; 101794. DOI: 10.1016/j.media.2020.101794.

[2] A. Amyar, R. Modzelewski, H. Li, and S. Ruan. (2020). Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: Classification and segmentation. *Comput. Biol. Med*., 126; 104037. DOI: 10.1016/j.compbiomed.2020.104037.

[3] M. M. Islam, F. Karray, R. Alhajj and J. Zeng. (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.

[4] S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao, J. Guo, M. Cai, J. Yang, Y. Li, X. Meng and B. Xu. (2021). A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). *Eur. Radiol.*, 31(8); 6096-6104. DOI: 10.1007/s00330-021-07715-1.

[5] G.D. Rubin, C.J. Ryerson, L.B. Haramati, N. Sverzellati, J.P. Kanne, S. Raoof, N.W. Schluger, A. Volpi, J.J. Yim, I.B.K. Martin, D.J. Anderson, C. Kong, T. Altes, A. Bush, S.R. Desai, J. Goldin, J.M. Goo, M. Humbert, Y. Inoue, H U. Kauczor, F. Luo, P.J. Mazzone, M. Prokop, M.Remy-Jardin, L. Richeldi, C.M. Schaefer-Prokop, N. Tomiyama, A.U. Wells and A.N. Leung. (2020). The role of chest imaging in patient management during the COVID-19 pandemic: a multinational consensus statement from the Fleischner Society. *Radiology*, 296 (11); 172–180,. DOI: 10.1016/j.chest.2020.04.003.

[6] F. Akba, I. T. Medeni, M. S. Guzel, and I. Askerzade. (2021). Manipulator Detection in Cryptocurrency Markets Based on Forecasting Anomalies. *IEEE Access*, 9; 108819–108831,. DOI: 10.1109/ACCESS. 2021.3101528

[7] R. Sille, T. Choudhury, P. Chauhan, and D. Sharma. (2021). Dense hierarchical CNN – A unified approach

for brain tumor segmentation. *Revue d'Intelligence Artificielle*, 35(3); 223–233,. DOI: 10.18280/ria. 350306.

[8] A. A. Yilmaz. (2022). A novel hyperparameter optimization aided hand gesture recognition framework based on deep learning algorithms. *Traitement du Signal*, 39(3); 823–833. DOI: 10.18280/ts.390307

[9] A. A. Yilmaz. (2022). Intrusion detection in computer networks using optimized machine learning algorithms. *3rd Int. Informatics and Software Eng. Conf. (IISEC)*, Ankara, Turkey.

[10] A. A. Yılmaz. (2024). A machine learning-based framework using the particle swarm optimization algorithm for credit card fraud detection. *Commun. Fac. Sci. Univ. Ankara Ser. A2-A3 Phys. Sci. Eng.*, 66, (1); 82–94. DOI: 10.33769/aupse.1361266.

[11] S. Bhattacharya, P. K. R. Maddikunta, Q. Pham, T. R. Gadekallu, S. R. Krishnan, C. L. Chowdhary, M. Alazab And Md. J. Piran. (2021). Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey. "*Sustain. Cities Soc.*, 65; 102589. DOI: 10.1016/j.scs.2020.102589.

[12] Ş. Özsarı, F. Z. Ortak, M. S. Güzel, M. B. Başkır and G. E. Bostanci. (2023). ML based prediction of COVID-19 diagnosis using statistical tests. *Commun. Fac. Sci. Univ. Ankara Ser. A2-A3 Phys. Sci. Eng.*, 65, (2); 79–99. DOI: 10.33769/aupse.1227857.

[13] F. Shan et al., "Lung infection quantification of COVID-19 in CT images with deep learning," arXiv preprint arXiv:2003.04655, 2020. DOI: 10.1002/mp.14609.

[14] A. I. Khan, J. L. Shah, and M. M. Bhat. (2020). CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest X-ray images. *Comput. Meth. Prog. Biomed.*, 196; 105581. DOI: 10.1016/j.cmpb.2020.105581.

[15] J. Zhao, Y. Zhang, X. He, and P. Xie. (2020). COVID-CT-dataset: a ct scan dataset about covid-19. *arXiv preprint* arXiv:2003.13865

[16] E. Soares, P. Angelov, S. Biaso, M. H. Froes, and D. K. Abe. (2020). SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification. *medRxiv*. DOI: 10.1101/2020.04.24.20078584.

[17] K. He, X. Zhang, S. Ren, and J. Sun. (2015). Deep residual learning for image recognition. *arXiv preprint* arXiv: 1512.03385

[18] O. Ronneberger, P. Fischer, and T. Brox. (2015). U-Net: Convolutional networks for biomedical image segmentation," Int. Conf. on Medical Image Computing and Computer-Assisted Intervention, Springer, Cham, pp. 234–241.

[19] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv preprint* arXiv:1602.07360.

[20]A. Krizhevsky, I. Sutskever, and G. Hinton, (2012). Imagenet classification with deep convolutional neural networks. *Int. Conf. Neural Inf. Process. Syst*., pp. 1097–1105.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *In Proc. of CVPR*, pp. 4510–4520.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. (2020) Densely connected convolutional networks. *In Proc. of CVPR*, pp. 1–8.

[23] D. P. Kingma and J. Ba. (2014). Adam: A method for stochastic optimization," *arXiv preprint* arXiv:1412.6980.

[24] J. Duchi, E. Hazan, and Y. Singer. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7); 2121–2159.

[25] R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik. (2019). SGD: general analysis and improved rates. *Int. Conf. Mach. Learn.*, pp. 5200–5209.

[26] T. Tijmen. Lecture slides on neural networks. (2012). http://www.cs.toronto.edu/tijmen/csc321/slides/lecture slides lec6.pdf (Accessed Jan. 1, 2025)

[27] X. Wang, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, W. Liu, C. Zheng. (2020). A weakly-supervised framework for COVID-19 classification and lesion localization from chest CT. *IEEE Trans. Med. Imaging*, 39(8); 2615–2625. DOI: 10.1109/TMI.2020. 2995965.

[28] F. Liao, M. Liang, Z. Li, X. Hu, and S. Song. (2019). Evaluate the malignancy of pulmonary nodules using the 3-D deep leaky noisy-OR network. *IEEE Trans. Neural Netw. Learn. Syst.*, 30(11), 3484–3495. DOI: 10.1109/TNNLS.2019.2892409.

[29] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. (2016). Learning deep features for discriminative localization. *In Proc. of CVPR*, pp. 2921–2929.

[30] A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, and M. Kaur. (2020). Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning. *J. Biomol. Struct. Dyn.*, 39(15); 4700–4708. DOI: 10.1080/07391102.2020.1788642.

[31] Y. Song, S. Zheng, L. Li, X. Zhang, X. Zhang, Z. Huang, J. Chen, R. Wang, H. Zhao, Y. Chong, J. Shen, Y. Zha and Y. Yang. (2021). Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. *IEEE/ACM Trans. Comput. Biol. Bioinf.* DOI: 10.1109/TCBB.2021.3065361.

[32] X. He, X. Yang, S. Zhang, J. Zhao, Y. Zhang, E. Xing and P. Xie. (2020). Sample efficient deep learning for COVID-19 diagnosis based on CT scans. *medRxiv*. DOI: 10.1101/2020.04.13.20063941.

[33] X. Wu, H. Hui, M. Niu, L. Li, L. Wang, B. He, X. Yang, L. Li, H. Li, J. Tian and Y. Zha. (2020). Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: a multicentre study. *Eur. J. Radiol*., 128; 109041. DOI: 10.1016/j.ejrad.2020.109041.

[34]W. Kntar. (2019). Fuzzy color image enhancement algorithm.https://github.com/WaseemKn/FuzzyColor ImageEnhancement-FuzzyLogicCourse-ITE5thYear?tab=readme-ov-file (Accessed Jan. 1, 2025)

[35] T. Bardak and S. Bardak. (2017). Prediction of wood density by using red-green-blue (RGB) color and fuzzy logic techniques. *J. Polytech*., 20; 979–985. DOI: 10.2339/politeknik.369132.

[36] J. Arnal and L. Súcar. (2020). Hybrid filter based on fuzzy techniques for mixed noise reduction in color images. *Appl. Sci*., 10(1); 243. DOI: 10.3390/app10010243.

[37] J. M. Soto-Hidalgo, D. Sanchez, J. Chamorro-Martinez, and P. M. Martínez-Jimenez. (2019). Color comparison in fuzzy color spaces, *Fuzzy Sets Syst*., 390; 160–182 DOI: 10.1016/j.fss.2019.09.013.

[38] S. Ruder. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint* arXiv: 1609.04747.

[39]V. Nair and G. E. Hinton. (2010). Rectified linear units improve restricted Boltzmann machines. *In Proc. of* ICML, pp. 807–814

[40]Naresh Babu KOSURI, & Suneetha MANNE. (2024). Revolutionizing Facial Recognition: A Dolphin Glowworm Hybrid Approach for Masked and Unmasked Scenarios. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.560

[41]Agnihotri, A., & Kohli, N. (2024). A novel lightweight deep learning model based on SqueezeNet architecture for viral lung disease classification in X-ray and CT images. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.425

[42]Anakal, S., K. Krishna Prasad, Chandrashekhar Uppin, & M. Dileep Kumar. (2025). Diagnosis, visualisation and analysis of COVID-19 using Machine learning . *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.826

[43]Say, A., Çakır, D., AVRAMESCU, T., USTUN, G., NEAGOE, D., KAHVECİ, M., … KOMOREK, J. (2024). Examining the Prevalence of Long-Covid Symptoms: A Cross-Sectional Study. *International Journal of Computational and Experimental Science and Engineering*, 10(1). https://doi.org/10.22399/ijcesen.243