Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 12, No. 3, 2024, 423-434.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Designing a Visual Geometry Group-based Triad-Channel Convolutional Neural Network for COVID-19 Prediction

Seyed Alireza Bashiri Mosavi[1*], Omid Khalaf Beigi[2] and Arash Mahjoubifard[3]

1. Department of Electrical and Computer Engineering, Buein Zahra Technical University, Buein Zahra, Qazvin, Iran.
2. Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran.
3. Department of Computer Engineering and Information Technology, University of Qom, Qom, Iran.

Article Info

Article History:
Received 18 October 2024
Revised 23 November 2024
Accepted 04 December 2024

DOI:10.22044/jadm.2024.15205.2626

Keywords:
COVID-19 Prediction, Convolutional Neural Network, Transfer Learning, Computer Vision, Image Processing.

**Corresponding author: abashirimosavi@bzte.ac.ir (S.A. Bashiri Mosavi).*

Abstract

Using intelligent approaches in diagnosing the COVID-19 disease based on machine learning algorithms (MLAs), as a joint work, has attracted the attention of pattern recognition and medicine experts. Before applying MLAs to the data extracted from infectious diseases, techniques such as RAT and RT-qPCR were used by data mining engineers to diagnose the contagious disease, whose weaknesses include the lack of test kits, the placement of the specialist and the patient pointed at a place and low accuracy. This study introduces a three-stage learning framework including a feature extractor by visual geometry group 16 (VGG16) model to solve the problems caused by the lack of samples, a three-channel convolution layer, and a classifier based on a three-layer neural network. The results showed that the Covid VGG16 (CoVGG16) has an accuracy of 96.37% and 100%, precision of 96.52% and 100%, and recall of 96.30% and 100% for COVID-19 prediction on the test sets of the two datasets (one type of CT-scan-based images and one type of X-ray-oriented ones gathered from Kaggle repositories).

1. Introduction

After the outbreak of the COVID-19 pandemic at the end of 2019, people all over the world faced a new challenge. As a result of the virus's high spreading characteristic, the challenge of maintaining a social distance (accepted as 2 meters) arose. In addition to disrupting daily life, it caused issues for medical staff and jobs that require concentrated working environments in fixed locations. This led to the emergence of new concepts such as teleworking and performing job tasks through internet platforms. Despite all the problems caused by the COVID-19 disease, the pandemic has also created opportunities to accelerate the diagnosis process. As more people moved online, the virus outbreak generated a large volume of structured and unstructured data, creating numerous business and research opportunities in various fields. Among these opportunities, utilizing artificial intelligence algorithms to address issues such as prediction, diagnosis, and vaccine development for COVID-19 patients is notable [1].

Before medical experts focused on applying machine learning techniques to medical diagnosis, methods such as RAT (Rapid Antigen Test) and RT-qPCR (Real-Time Reverse Transcription Quantitative Polymerase Chain Reaction) garnered significant attention from medical experts. When compared to other methods, the first technique is more prone to errors and may wrongly categorize patients with lower viral loads as healthy. In other words, the mentioned technique's false-negative criterion assigns a high value to these types of patients. The second technique has better automatic performance (about 90% accuracy), but it takes more time and it needs a special kit that some laboratories or clinics are not capable of obtaining or having the required specialists to work with. In addition to the mentioned deficiencies of sale, the patient and the specialist must be placed in a room

to perform the test, and the lack of observance of the medical protocols by both parties will cause the transmission of the virus. It is essential to utilize alternative methods for diagnosing the mentioned viruses in addition to laboratory tests. Thus, in addition to non-automated methods, experts diagnose COVID-19 using chest image analysis from X-ray or CT-Scan techniques [2-4].

Careful examination of recent research on COVID-19 diagnosis reveals compelling evidence supporting the use of features extracted by sophisticated pre-trained networks. From the point of using the type of algorithm, the use of support vector machine and neural network methods as the most used algorithms in the group of samples is noticeable. In this paper, we introduce a convolutional architecture based on visual geometry group 16 (VGG16), trained on the ImageNet dataset, as the classification method of X-ray and CT images. In the proposed structure, the feature matrices calculated by VGG16 are then mapped to a matrix with 3 channels and classified into two classes (infected and healthy) by a neural network. First, VGG16 extracts the comprehensive features. Next, the features related to the COVID-19 samples are extracted and classified by the final component. The proposed framework is termed the Covid visual geometry group 16 (CoVGG16). With the development of Internet-based services (such as GitHub and Kaggle), it is possible to share the characteristics of healthy people infected with COVID-19, and there is a suitable dataset for study. Moreover, advances in computer hardware and the development of machine learning techniques have enabled algorithms like neural networks and their derivatives to diagnose and treat various diseases, reducing the need for medical staff. For example, researchers have developed a multi-task model (Recursive Convolutional Neural Network) to determine the type of breast cancer [5]. One of the challenges of developing an intelligent system in the medical field is the lack of standard data. This weakens the performance of neural network-based architectures and the usefulness of the resulting system. In other words, the model generalization capacity decreases. To solve this problem, many solutions have been provided for image data collection systems, such as methods of dataset augmentation (rotation, shift, zoom, and so on) and using generative algorithms (such as GAN10) to produce synthetic data that are as close to real data as possible. To solve the problem of the lack of breast cancer images, researchers also have invented a method based on GAN (Generative Adversarial Network) to generate new samples according to existing samples in the opposite class. In other words, the designed algorithm is applied to produce infected samples with cancer or healthy samples and vice versa. The data used in the algorithm is composed of four different channels, the information stored in each as the determining matrix of the related cancerous part (specified by using random numbers based on the normal distribution), the matrix of the gray-level cancerous tumor (labeled as a mask), and two classes (benign and malignant) which are characterized by two values [6]. As mentioned earlier in this section, artificial intelligence-based systems can be used with basic methods for diagnosing and treating serious diseases. The other challenge in medical issues is the cost and complexity of the diagnosis process for many patients. To overcome these challenges, researchers have developed a system using Encoder architecture (introduced as the Clinical Decision Support System) for diagnosing disease and recognizing the characteristics of the patient [7]. In another research, the researchers introduced the voice of the patient as the criterion for the diagnosis of COVID-19. First, they collected audio data based on 5 tasks (sustained phonation /aaaa/, three times coughing, three times deep breathing, counting from 1 to 20, and reading certain/defined sentences), then used GeMaps (Geneva Minimalistic Acoustic Parameter Set), eGeMaps2 (extended Geneva Minimalistic Acoustic Parameter Set), ComParE, Wavelet, VGGish (128-dimensional feature extraction by pre-trained framework) and OpenL3 methods, to extract the existing features in the data. It's crucial to highlight that in the classification of extracted features, three combined methods (Bagging, Boosting, and Random Forest), MLP, VGGish, and OpenL3 used, and the best performance is related to the combination of Wavelet and Boosting with an accuracy of 88.52% [8]. Another solution to solve the problem of the lack of training data is to use pre-trained models. For example, in another study, researchers used End-to-End Training, Fine-Tuning, and Feature Extraction with the aid of pre-trained models (such as ResNet18). In addition to neural network-based methods, the Support Vector Machine (SVM) algorithm is used for image classification, with the most accurate approach being the combination of SVM (using the Linear Kernel function) and the ResNet50 feature extractor, achieving an accuracy of 94.7% [9]. In another similar research, the estimation of the probability of disease diagnosis based on the chest cavity, taken from the set of CT-Scan images and extracted features by DenseNet201 (including 201 layers) was discussed. The features extracted from the images using this architecture are fed into a

network with two hidden layers and an output layer, which then determines the class of the model [10]. One of the challenges in deep neural network training is determining hyper-parameters and choosing a suitable structure for architecture. To solve it, researchers used an algorithm called MADE (Memetic Adaptive Differential Evolution) to determine the best hyper-parameters of the architecture used in image classification. Unlike previous research that used convolutional neural networks to extract and classify chest images, this study employs Long-Short Term Memory (LSTM) and Mixture Density Network to classify CT-Scan results, achieving an accuracy rate of 96.19%. [11-13].

The approaches discussed in this section have limitations in addressing data scarcity and model generalizability. For example, using a framework based on GANs in [6] is associated with the challenge of network training, which will initially lead to significant overhead. Among their other challenges are the high costs of training the model and matching it with similar datasets. Also, in order to overcome the challenges mentioned, [21] and [22] have employed the strategies of combining similar datasets and utilizing feature selection. However, it's important to note that this approach presents challenges such as high computational requirements and manual feature selection. Moreover, in [8], researchers have applied the audio processing approach for voice recognition, which is associated with the challenge of collecting appropriate datasets (according to the mentioned tasks) and low performance (accuracy less than 90%). Given the highly transmissible nature of COVID-19 and its significant impact on individuals, the performance of frameworks based on predefined criteria, such as accuracy, is absolutely crucial. According to the explanations, an approach to deal with the lack of data is to use pre-trained models. However, it is important to choose a suitable and lightweight model (preventing memory challenges and low speed) to extract the basic and pre-trained features. Despite utilizing pre-trained models in researches [9] and [10], the features extracted from the model are still based on ImageNet data. The dataset contains images of 1000 different classes that have no connection to the samples of the diagnosis of COVID-19. Therefore, a framework based on one lightweight pre-trained model (VGG16) was introduced and compared to extract comprehensive features, and an additional feature extraction layer to obtain features with more details and a fully connected network for the sample classification. Due to the extraction of features from two levels by a pre-trained model and a sequential component, the proposed framework has brought a great promising performance.

The key contributions of this paper to handle the above-mentioned challenges are categorized as follows:

- Integrating a pre-trained feature extractor in CoVGG16 to reduce training time and computational resource requirements.
- Using the lightweight VGG model and a feature extractor at the partial level to meet the lack of connection between the applied dataset and the dataset used to achieve the pre-trained models.
- CoVGG16 benefits from a lightweight structure that does not need to determine the hyper-parameters, which is a significant challenge in training neural network-based frameworks in past works.
- Comparing the proposed method with other classification techniques offered in previous research based on the same experimental scenario.

The rest of the paper is organized as follows: The proposed framework is elaborated in Section 2. Experimental results related to applying the learning model on the utilized datasets for COVID-19 prediction are discussed in Section 3. Furthermore, comparing the performance of our model with other classifiers is detailed at the end of Section 3. Finally, the conclusion is noted in Section 4.

2. The Proposed Method (CoVGG16)

2.1. CoVGG16 Components

The CoVGG16 architecture consists of two parts: a) VGG16 feature extraction blocks (pre-trained on ImageNet) and an additional three-channel feature extraction layer. It is worth knowing that the ImageNet collection is made of classified samples in 1000 different groups [15], and b) classification, which consists of two dense layers (respectively with 512 and 256 nodes) and an output layer with a single node. The visual summary of CoVGG16 is depicted in figure 2. Also, the pseudocode of the proposed framework is shown in table 1. For more information about proposed learning model details refer to the following subsections.

2.1.1. Convolution Layer

The convolutional layer is one of the constituent elements of a convolutional neural network that performs the function of feature extraction from the values in numerical structures (vector matrices in Euclidean space). This process can be implemented using the convolution operator (1):

$$F[m,n] = a(X[m,n] * K[m,n]) = \quad (1)$$
$$a(\sum_{j=-\infty}^{+\infty} \sum_{i=-\infty}^{+\infty} X[i,j] \bullet K[m-i, n-j])$$

In (1), X is the input matrix, K is the filter matrix (Kernel), α is the activation function, * is the convolution operator, and • is point-to-point multiplication (Hadamard Product).

2.1.2. Dense Layer

Besides the noticeable capability of the neural network to model data, in some cases by the rise of complexity and disability to be separated linearly, this ability transforms into a considerable inefficiency. To solve this problem, layers are used in the extension template, called "dense" (Combination of linear equation with activation function) (2). This method provides the ability to solve non-linear separable problems for the neural network. Also, the above layers are known as “Hidden Layers”.

$$Y = a(W \circ X + b) \quad (2)$$

In (2), W is the needle matrix, X is the row matrix, b is the bias vector, α is the activation function, $\circ$ is the dot product, and Y is the layer output. In section 2.2, pre-trained models are elaborated.

2.2. Pre-Trained Models

Based on the research reviewed in the introduction section, various methods are used to solve the problem of model generalization capacity to predict and neutralize the impact of the lack of datasets in the training process. In this study, to accurately extract the features in each sample, one intensive pre-trained model on the ImageNet dataset is used in the proposed architecture: VGG16. The VGG16 consists of 13 convolution layers, 5 Max Pooling layers (choosing the largest values in a certain dimension), an Adaptive Average Pooling layer, two dense layers, and an output layer. The structure of this model can also be seen in figure 1. It is worth noting that to investigate the performance of the feature extraction phase of the proposed framework (CoVGG16), it has been compared with another pre-trained architecture called VGG19. This architecture has a larger number of parameters for feature extraction and is referred to as QUASI-CoVGG16 (CoVGG19) in the current study (Figure 4).

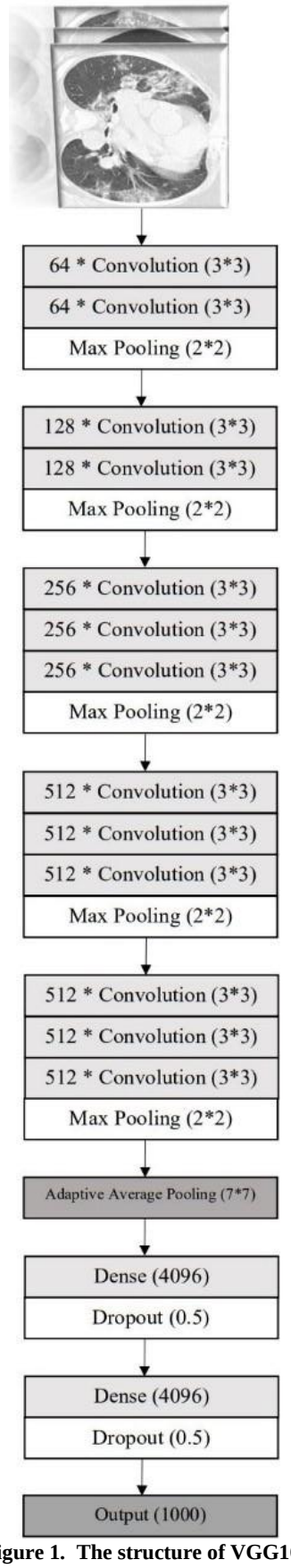

Figure 1. The structure of VGG16.

This model receives each image as a 3-channel matrix (RGB structure and 224*224 sizes). The feature matrices are extracted by applying filter matrices on the input by selecting the largest value

(2*2 dimensions). After detecting the existing features in each sample, their related class is predicted by the post-processing of the hidden layers with 4096 nodes and the Dropout regularization with a 0.5 probability rate.

2.3. COVID-19 Datasets

This study utilizes two datasets to evaluate the performance of the proposed architecture. The first dataset, available at (https://www.kaggle.com/datasets/plameneduardo/sarscov2-ctscan-dataset) contains 1230 non-infected cases and 1252 infected ones. These data have been collected from real patients in hospitals from Sao Paulo, Brazil. The second dataset, found at(https://www.kaggle.com/datasets/khoongweihao/covid19-xray-dataset-train-test-sets) consists of 94 infected samples and 94 healthy samples including X-ray scans of the chest of healthy individuals infected with COVID-19. The second dataset is collected from public sources as well as through indirect collection from hospitals and physicians. Before model design, the images were resized to 224x224 pixels. To ensure precise accuracy and validate results, we utilized the k-fold cross-validation method (with k=10) to partition the data into train and test sets. In this method, one part of the dataset serves as test samples while the remainder acts as training samples, with this process repeated k times. The readers of this article can access the corresponding repository page by clicking on each dataset's link. Samples of X-ray and CT-Scan datasets are shown in figure 3 [16-19]. In Figure 3, the right column shows infected samples while the left column displays healthy samples. The top row contains CT-Scan images and the bottom row is related to X-ray images.

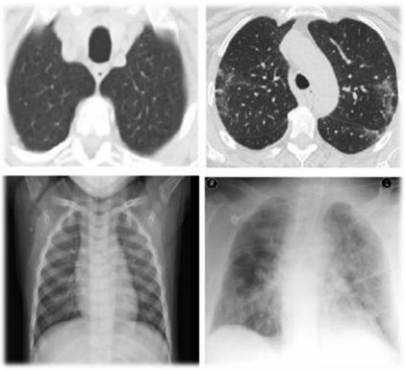

Figure 3. Samples of COVID-19 data extracted from two datasets.

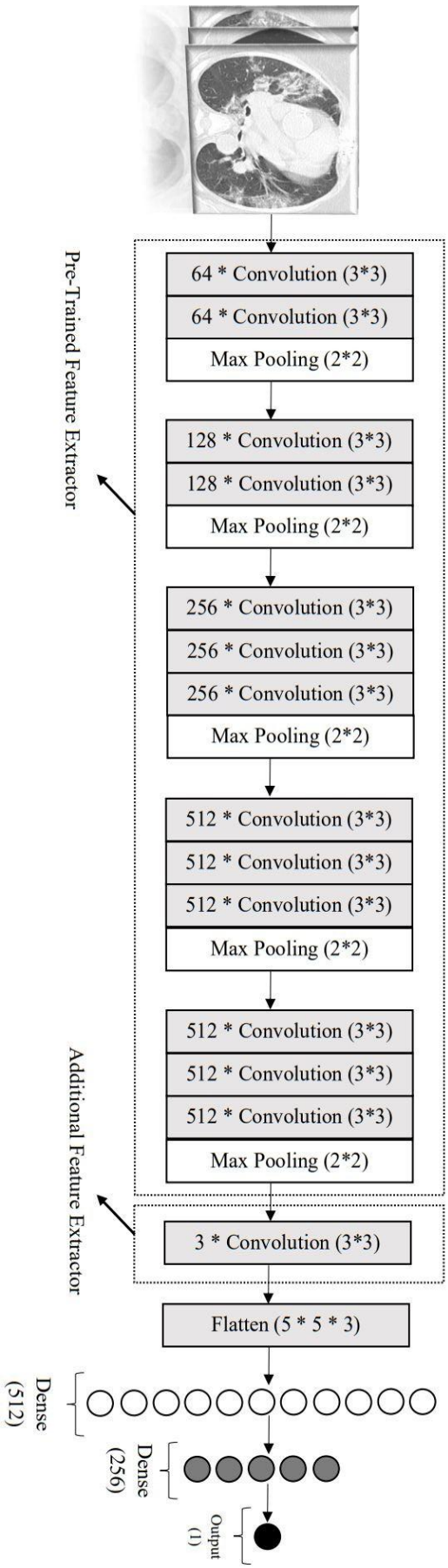

Figure 2. The overall summary of CoVGG16 model.

Table 1. The pseudo-code of CoVGG16.

Input (x): CT-scan and X-ray RGB images; **Output (y)**: Target class (healthy or infected with Covid-19)
Feature Extraction Blocks
Block1
Conv1 ← Convolution (channels=64, kernel_size=(3, 3), requires_grad=False)(x)
Conv2 ← Convolution (channels=64, kernel_size=(3, 3), requires_grad=False)(Conv1)
MaxPool ← MaxPooling (kernel_size=(2, 2))(Conv2)
Block2
Conv1 ← Convolution (channels=128, kernel_size=(3, 3), requires_grad=False)(MaxPool)
Conv2 ← Convolution (channels=128, kernel_size=(3, 3), requires_grad=False)(Conv1)
MaxPool ← MaxPooling (kernel_size=(2, 2))(Conv2)
Block3
Conv1 ← Convolution (channels=256, kernel_size=(3, 3), requires_grad=False)(MaxPool)
Conv2 ← Convolution (channels=256, kernel_size=(3, 3), requires_grad=False)(Conv1)
Conv3 ← Convolution (channels=256, kernel_size=(3, 3), requires_grad=False)(Conv2)
MaxPool ← MaxPooling (kernel_size=(2, 2))(Conv3)
Block4
Conv1 ← Convolution (channels=512, kernel_size=(3, 3), requires_grad=False)(MaxPool)
Conv2 ← Convolution (channels=512, kernel_size=(3, 3), requires_grad=False)(Conv1)
Conv3 ← Convolution (channels=512, kernel_size=(3, 3), requires_grad=False)(Conv2)
MaxPool ← MaxPooling (kernel_size=(2, 2))(Conv3)
Block5
Conv1 ← Convolution (channels=512, kernel_size=(3, 3), requires_grad=False)(MaxPool)
Conv2 ← Convolution (channels=512, kernel_size=(3, 3), requires_grad=False)(Conv1)
Conv3 ← Convolution (channels=512, kernel_size=(3, 3), requires_grad=False)(Conv2)
MaxPool ← MaxPooling (kernel_size=(2, 2))(Conv3)
Additional Feature Extraction Layer
AddConv ← Convolution (channels=3, kernel_size=(3, 3), requires_grad=True)(MaxPool)
*Flatten ← FlattenLayer (5 * 5 * 3)(AddConv)*
Fully-Connected Network (FCN)
Dense1 ← Dense (nodes=512, activation=ReLU)(Flatten); Dense2 ← Dense (nodes=256, activation=ReLU)(Dense1)
y ← Dense (nodes=1, activation=Linear)(Dense2); y ← Sigmoid (y)

3. Experimental Results

3.1. Training Details and Implementation Tools

The Python programming language (version 3.7) and the PyTorch library were used to implement the models. Google Colab cloud system (12.68 GB memory, Intel Xeon 2.2 GHZ central processor, and Tesla T4 15 GB graphics processor) and Adam optimization algorithm with a learning rate of 0.0001, eps1e-8, beta 0.9, and beta 0.999 were used to train and determine the parameters of the proposed structures. The variables are values to control the time steps or the speed of the learning process (learning rate), increasing the stability of learning and not dividing by zero (Epsilon), and determining the moving average (Beta1 and Beta2 values). The size of the training batch (batch size) is 32 and the number of training epochs is 30 [20].

3.2. COVID-19 Prediction Based-on Proposed Method

Measuring the efficacy of CoVGG16 in COVID-19 prediction is on the agenda in this section. The performance evaluation of the CoVGG16 per dataset (the first and second datasets) is done based on the 10-fold cross-validation technique, wherein the CoVGG16 classifier is used for the train-test procedure running for each fold. Also, to find the optimal values among training parameters (See Section 3.1), we conducted train-test procedures per fold based on different settings. The classification metrics for evaluating the performance of the proposed framework in predicting COVID-19 status per dataset is accuracy (Acc=(TP+TN)/(TP+TN+FP+FN)). Taking into cognizance points regarding requirements of train-test procedures for COVID-19 prediction, the values of the Acc index in COVID-19 prediction per dataset in each fold are shown in table 2. By setting the different values for training parameters, the maximum value of Acc among Acc variations is captured per fold. These values are recorded in table 2. The accuracy of the COVGG16 in different folds and datasets is reported in table 2 (CoVGG16 is highlighted by bold underline-face). For example, CoVGG16 in the 1st, 5th, and 8th fold of the first dataset has an accuracy of 96.78%, 97.17%, and 98.38% on test sets, respectively. In the case of the second dataset, the CoVGG16 has 100% accuracy in all test sets. According to Table 2, the accuracy (last column in Table 2) of CoVGG16 related to the test sets for the first and second datasets are 96.37% and 100%, respectively. It is worth noting that the values of precision and recall in Tables 4 and 5 are also significant for all learning models.

Table 2. The obtained accuracies based on applying CoVGG16, CoVGG19 and End2End models on two datasets.

Dataset	Learning model	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Average Accuracy
1	**CoVGG16**	96.78	96.37	95.56	95.16	97.17	95.96	96.37	98.38	96.77	95.16	**96.37**
2	**CoVGG16**	100	100	100	100	100	100	100	100	100	100	**100**
1	CoVGG19	95.58	95.56	96.37	94.75	93.95	94.35	96.77	97.17	96.37	94.75	95.56
2	CoVGG19	100	100	100	100	94.73	94.73	94.73	100	100	100	98.42
1	End2End	94.77	93.54	90.72	90.72	92.33	92.33	95.16	97.58	92.33	91.93	93.14
2	End2End	94.73	100	100	94.73	100	100	89.47	100	88.88	100	96.78

3.3. Comparison of Experimental Methods (CoVGG16 vs. QUASI-CoVGG16 AND NON-QUASI-CoVGG16)

For precise performance evaluation of the proposed learning model, we compare the CoVGG16 with quasi-CoVGG16 (CoVGG19) (Feature extraction phase comparison) and non-quasi-CoVGG16 (End2End) (Feature extraction and classification comparison) for COVID-19 prediction in this section. In the case of quasi-CoVGG16, we compare our proposed framework with the CoVGG19 model (See Figure 4) and the End2End model (See Figure 5). Compared to CoVGG16, the CoVGG19 model has more parameters (three convolution layers) and the ability to extract more features. For addressing the performance of CoVGG16, we consider the CoVGG19, End2End frameworks and two other architectures that have been proposed in similar studies. In the End2End architecture, unlike to CoVGG16 and CoVGG19 models, feature extraction discovery matrices are determined during the training process. End2End architecture consists of two convolution layers, two Max Pooling layers, and three dense layers. It is noticeable that the End2End architecture has been introduced to investigate the effects of using pre-trained models.

The results of training the CoVGG16 model on the two datasets and its comparison with CoVGG19 and End2End are available in table 2. According to the obtained results, the performance of the End2End model in different folds and datasets is equal to or less than CoVGG16. For example, the End2End model in the 8^{th} fold of the second dataset has the same performance as CoVGG16. However, this rule does not apply to other pairs of structures. For example, in the 7^{th} fold of the first dataset, CoVGG19 is more powerful to generalize the training set to the test set than CoVGG16. Nevertheless, CoVGG16 has better performance in the 5^{th} fold of the same dataset. As another example, in the 6^{th} fold of the second dataset, the End2End architecture has a higher performance than CoVGG19, but this is not the case for the 7^{th} fold. According to the results in Table 2, the title of the best performance (based on the accuracy) belongs to the CoVGG16 model and the weakest to the End2End model. It plays a vital role in problem modeling as a feature extractor. In light of the results in Table 2, the CoVGG16 architecture has performed better on the two datasets despite using fewer convolutional and MaxPool2D layers. In both the first and second datasets, this architecture demonstrated remarkable accuracy rates of 96.37% and 100% on test sets respectively, surpassing CoVGG19 by 0.81% and 1.58%, and outperforming End2End by 3.23% and 3.22%. Notably, the error rate and accuracy plot from each training epoch (on the first fold per dataset) is presented in figure 6 (produced with the Matplotlib library). In Figure 6, the first graph (top) represents the error rate and accuracy plot for the first dataset and the second graph (bottom) represents the error rate and accuracy plot for the second dataset. According to Figure 6, all three convolutional architectures used for the first fold modeling method from the training dataset have the same performance and the minimum learning error rate. Based on the first chart, CoVGG16 has more power in generalizing the training samples to the test, and the amount of its test error tends to zero more than other models, on the other hand, based on the second chart, all three frameworks have the same test error, which is obvious according to the results in table 2. Based on the accuracy, CoVGG16 has a higher ability to model CT images in the mentioned dataset. Therefore, the End2End architecture is less capable of solving problems than the other two models (According to Table 2). It is worth noting that the accuracy rate of CoVGG16 per epoch has less variation and its learning process is more stable.

In the case of other proposed methods, we recount the results of some previous works focused on the same datasets that we used in this study. [21] has used the second dataset and achieved a maximum accuracy of 98.29%. In this research, researchers have tried to solve the problem of lack of samples and fix the problem of generalizability by combining two similar datasets and using data augmentation techniques. However, their proposed model is more computationally intensive compared

to the CoVGG16 framework. However, due to the use of a pre-trained model and secondary feature extractor, our proposed framework has achieved better performance by using fewer samples and less computation. In the case of the first dataset, the proposed method in [22] achieved an accuracy value of 94.33%. This article presents a unique approach based on feature selection. Despite its innovative nature, the use of feature selection for decision-making has resulted in lower performance. Our proposed framework obtained accuracies of 96.37% (on test data from the first data set) and 100% (on test data from the second data set), respectively. The results show that CoVGG16 outperforms other frameworks on COVID-19 prediction. For more information refer to table 3. It is worth noting that, to carefully examine all three mentioned frameworks and compare the two frameworks with CoVGG16, the confusion matrices for each of the test sets in Fold 2 can be seen in figures 7 and 8. According to the confusion matrices (Figure 7), the number of TP (True Positive) and TN (True Negative) values of 108 and 131 were recorded for the first set and CoVGG16. Also, the number of FN (False Negative) and FP (False Positive) variables has a value of 4 and 5, which indicates the high performance of CoVGG16 in separating the samples from each other. On the other hand, the CoVGG19 and End2End frameworks have more FN and FP values, which indicates the high risk of using these models (instead of CoVGG16) to diagnose COVID-19. For this reason, according to the listed results, it is logical to use CoVGG16 with its impressive performance and very low error rate.

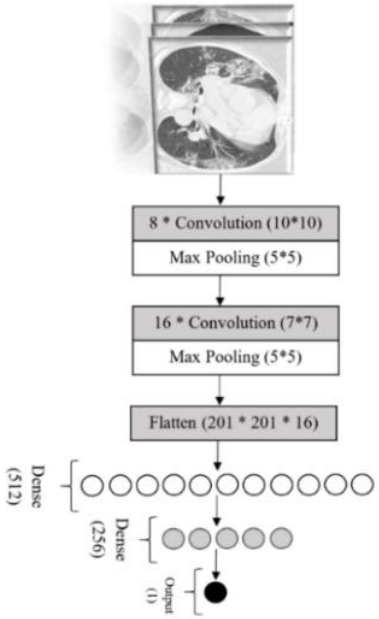

Figure 5. The overall summary of the End2End.

Table 3. Comparing CoVGG16 with other proposed methods.

Methods	Dataset	Accuracy (%
Bashiri Mosavi et al. [22]	1	94.33
Bashiri Mosavi et al. [22]	2	98.66
Khan et al. [21]	2	98.29
Our proposed framework	1	96.37
Our proposed framework	2	100

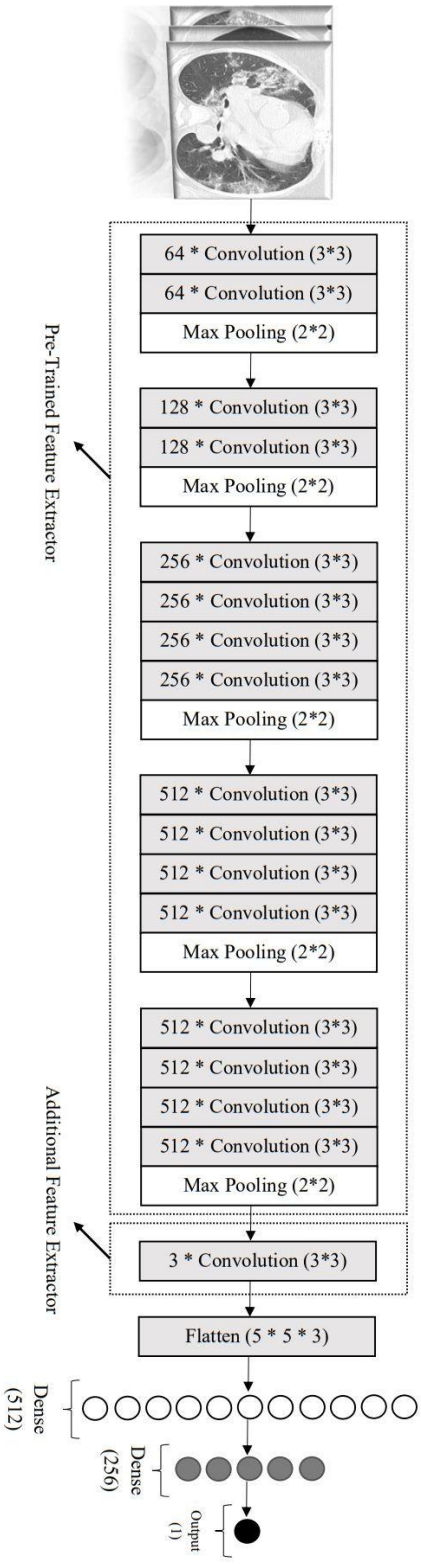

Figure 4. The overall summary of CoVGG19.

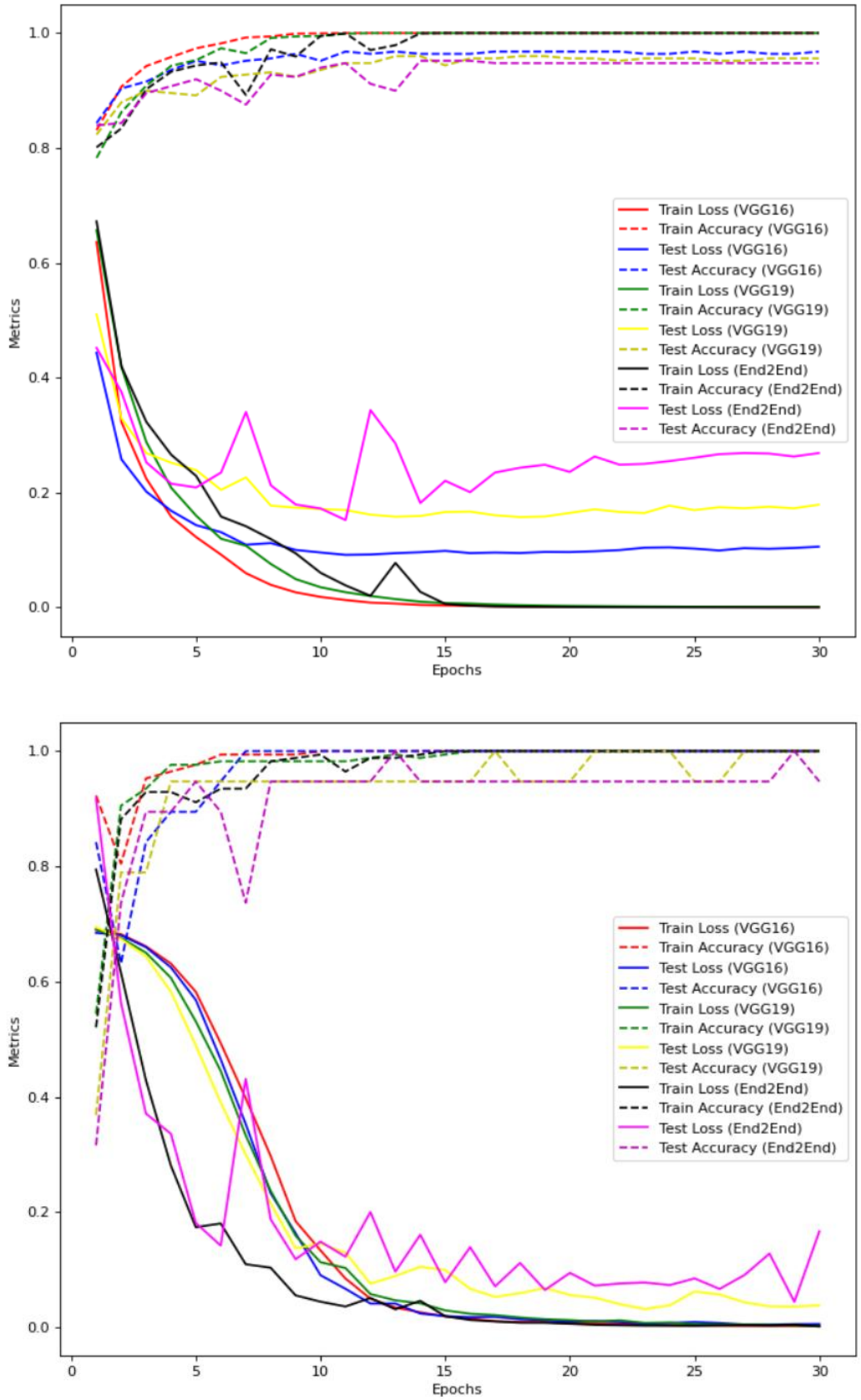

Figure 6. The loss and accuracy diagrams of CoVGG16, CoVGG19, and End2End on the train-test splits related to the first and second datasets (First fold per dataset).

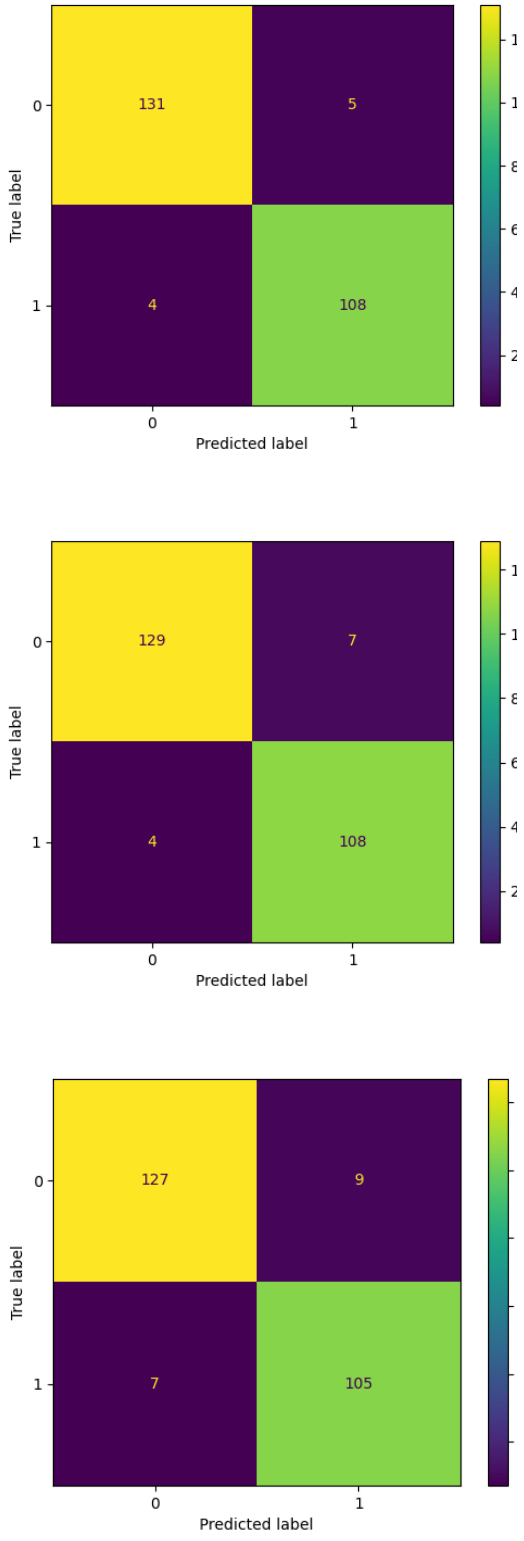

Figure 7. Confusion matrices for the first dataset (fold 2).

Top: CoVGG16, Middle: CoVGG19, Bottom: End2End

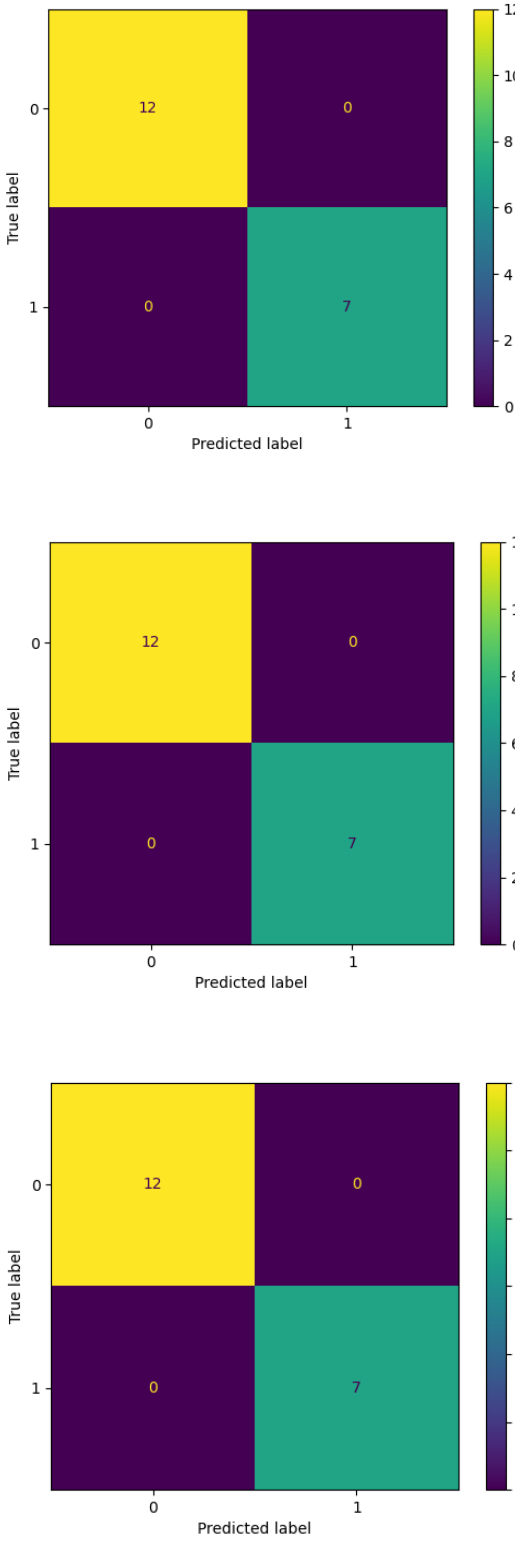

Figure 8. Confusion matrices for the second dataset (fold 2).

Top: CoVGG16, Middle: CoVGG19, Bottom: End2End

Table 4. The obtained precision scores based on applying CoVGG16, CoVGG19 and End2End models on two datasets.

Dataset	Learning model	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Average Precision
1	**CoVGG16**	98.46	95.57	96.85	94.57	98.40	93.22	95.31	97.61	97.61	97.61	**96.52**
2	**CoVGG16**	100	100	100	100	100	100	100	100	100	100	**100**
1	CoVGG19	94.24	93.91	95.48	95.96	95.20	90.83	97.56	96.06	97.60	94.07	95.09
2	CoVGG19	100	100	100	100	100	91.66	100	100	100	100	99.16
1	End2End	98.40	92.105	89.62	91.93	89.208	90.43	93.79	96.09	92.24	90.57	92.44
2	End2End	85.71	100	100	100	100	100	81.81	100	100	100	96.75

Table 5. The obtained recall scores based on applying CoVGG16, CoVGG19 and End2End models on two datasets.

Dataset	Learning model	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Average Recall
1	**CoVGG16**	95.52	96.42	94.61	96.06	96.09	98.21	97.60	99.19	96.09	93.18	**96.30**
2	**CoVGG16**	100	100	100	100	100	100	100	100	100	100	**100**
1	CoVGG19	97.76	96.42	97.69	93.70	92.96	97.32	96.0	98.38	95.31	96.21	96.17
2	CoVGG19	100	100	100	100	88.88	100	88.88	100	100	100	97.77
1	End2End	91.79	93.75	93.07	89.76	96.87	92.85	96.80	99.19	92.96	94.69	94.17
2	End2End	100	100	100	90.0	100	100	100	100	80.0	100	97.0

To more accurately examine the performance of the proposed framework and other models, the recall (Recall=(TP/(TP+FN)) and precision (Precision=(TP/(TP+FP)) metrics have also been calculated and examined on the test data. Precision measures the accuracy of the positive predictions made by the model. It's the ratio of true positive results to the total predicted positives. High precision indicates that the model is very selective in its positive predictions and makes fewer false-positive errors. Recall (also known as sensitivity) measures the ability of the model to identify all relevant instances (true positives). It's the ratio of true positive results to the actual total positives (true positives + false negatives). High recall means the model captures most of the actual positives but may also include more false positives. It is worth noting that the table related to the mentioned criteria can be seen in tables 4 and 5.

4. Conclusion and Future Work

In this paper, we aim to reduce complexity in dealing with the lack of many samples as an advantage of the proposed framework compared to the previous models in COVID-19 prediction. In earlier works, data mining engineers consider increasing the number of samples by data augmentation and adversarial neural networks. In such circumstances, the learning scenario will face problems such as GPU memory overflow or random memory. Also, designing two differentiating and productive networks will pose a significant challenge. To address the mentioned concerns, in this paper, we offer a three-stage learning framework including a feature extractor by VGG16 model to solve the problems caused by the lack of samples, a three-channel convolution layer, and a fully connected network mounted on a three-layer neural network. Regardless of solving the mentioned challenges, our model through utilizing the pre-trained model and combining it with two trainable components organizes the weakness of the lack of more samples. The obtained results manifested that the CoVGG16 has an accuracy of 96.37% and 100%, precision of 96.52% and 100%, and recall of 96.30% and 100% for COVID-19 prediction on COVID-19 test sets of two datasets (one type of CT-scan-based images and one type of X-ray-oriented ones gathered from Kaggle repositories). According to the data set segmentation criterion and the obtained results, it has been observed that the proposed framework does not suffer from the problem of overfitting. However, it is expected that the proposed framework will not show the same performance when faced with data and other issues such as breast cancer diagnosis. Given the unique nature of other diseases, it is evident that CoVGG16 requires either minor or comprehensive modifications to address the highlighted weakness.

In future research, we will consider adding an attention mechanism to the proposed training model presented in this paper to boost the speed of training and inference scenarios for healthcare prediction.

References

[1] N. Arora, A.K. Banerjee, and M.L. Narasu, "The role of artificial intelligence in tackling COVID-19," *Future Virology*, vol. 15, no. 11, pp.717-724, 2020.

[2] A. Scohy, A. Anantharajah, M. Bodéus, B. Kabamba-Mukadi, A. Verroken, and H. Rodriguez-Villalobos, "Low performance of rapid antigen

detection test as frontline testing for COVID-19 diagnosis," *Journal of Clinical Virology*, vol. 129, p.104455, 2020.

[3] T. E. Miller, W. F. G. Beltran, A. Z. Bard, T. Gogakos, M. N. Anahtar, M. G. Astudillo, D. Yang, J. Thierauf, A. S. Fisch, G. K. Mahowald, and M. J. Fitzpatrick, "Clinical sensitivity and interpretation of PCR and serological COVID-19 diagnostics for patients presenting to the hospital," *The FASEB Journal*, vol. 34, no. 10, p.13877, 2020.

[4] S. Salehi, A. Abedi, S. Balakrishnan, and A. Gholamrezanezhad, "Coronavirus disease 2019 (COVID-19): a systematic review of imaging findings in 919 patients," American Journal of Roentgenology, vol. 215, no. 1, pp.87-93, 2020.

[5] M. M. Keikha, and Y. Kord Tamandani, "Breast Cancer Detection Using Deep Multilayer Neural Networks," *Journal of Epigenetics*, vol. 3, no. 1, pp.27-34, 2022.

[6] E. Wu, K. Wu, D. Cox, and W. Lotter, "Conditional infilling GANs for data augmentation in mammogram classification," in *Image Analysis for Moving Organ, Breast, and Thoracic Images: Third International Workshop, RAMBO 2018, Fourth International Workshop, BIA 2018, and First International Workshop, TIA 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, Proceedings 3. Springer International Publishing*, 2018, pp. 98-106.

[7] Y. Yang, P. A. Fasching, M. Wallwiener, T. N. Fehm, S. Y. Brucker, and V. Tresp, "Predictive clinical decision support system with RNN encoding and tensor decoding," *arXiv preprint arXiv:1612.00611*, 2016.

[8] V. Despotovic, M. Ismael, M. Cornil, R. Mc Call, and G. Fagherazzi, "Detection of COVID-19 from voice, cough and breathing patterns: Dataset and preliminary results," *Computers in Biology and Medicine*, vol. 138, p.104944, 2021.

[9] A. M. Ismael, and A. Şengür, "Deep learning approaches for COVID-19 detection based on chest X-ray images," *Expert Systems with Applications*, vol. 164, p.114054, 2021.

[10] A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, and M. Kaur, "Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning," *Journal of Biomolecular Structure and Dynamics*, vol. 39, no. 15, pp.5682-5689, 2021.

[11] Y. Pathak, P. K. Shukla, and K. V. Arya, "Deep bidirectional classification model for COVID-19 disease infected patients," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, vol. 4, pp.1234-1241, 2020.

[12] J. Schmidhuber, and S. Hochreiter, "Long short-term memory," Neural Comput, vol. 9, vol. 8, pp.1735-1780, 1997.

[13] C. M. Bishop, "Mixture density networks", 1994.

[14] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE conference on computer vision and pattern recognition*, 2009, pp.248-255.

[16] E. Soares, P. Angelov, S. Biaso, M. H. Froes, and D. K. Abe, "SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification," *MedRxiv*, pp.2020-04, 2020.

[17] P. Angelov, and E. Soares, "Towards explainable deep neural networks (xDNN)," *Neural Networks*, vol. 130, pp.185-194, 2020.

[18] J. Micah Bennett, "SMART-CT-SCAN_BASED COVID19_VIRUS_DETECTOR," 2020. [Online]. Available:https://github.com/JordanMicahBennett/SMART-CT-SCAN_BASED COVID19_VIRUS_DETECTOR/

[19] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi, "Covid-19 image data collection: Prospective predictions are the future," *arXiv preprint arXiv:2006.11988*, 2020.

[20] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] S. H. Khan, A. Sohail, M. M. Zafar, and A. Khan, "Coronavirus disease analysis using chest X-ray images and a novel deep convolutional neural network," *Photodiagnosis and Photodynamic Therapy*, vol. 35, p.102473, 2021.

[22] S. A. Bashiri Mosavi, M. Javaherian, and O. Khalaf Beigi, "Selecting Optimal Moments of Chest Images by Partialized-Dual-Hybrid Feature Selection Scheme for Morphological-based COVID-19 Diagnosis," *Journal of AI and Data Mining*, 2024.

طراحی یک شبکه‌ی عصبی پیچشی سه کاناله‌ی مبتنی بر گروه هندسه بصری برای پیش‌بینی کووید-۱۹

سید علیرضا بشیری موسوی[1,*]، امید خلف بیگی[2] و آرش محجوبی فرد[3]

[1] گروه مهندسی برق و کامپیوتر، مرکز آموزش عالی فنی و مهندسی بوئین زهرا، بوئین زهرا، قزوین، ایران.

[2] گروه مهندسی برق و کامپیوتر، دانشگاه خوارزمی، تهران، ایران.

[3] گروه مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه قم، قم، ایران.

ارسال ۲۰۲۴/۱۰/۱۸؛ بازنگری ۲۰۲۴/۱۱/۲۳؛ پذیرش ۲۰۲۴/۱۲/۰۴

چکیده:

استفاده از رویکردهای هوشمند در تشخیص بیماری کووید-۱۹ براساس الگوریتم‌های یادگیری ماشین به عنوان یک کار مشترک، توجه متخصصان تشخیص الگو و پزشکی را به خود جلب کرده است. قبل از اعمال الگوریتم‌های یادگیری ماشین بر روی داده‌های استخراج شده از بیماری‌های عفونی، تکنیک‌هایی مانند RAT و RT-qPCR توسط مهندسان داده‌کاوی برای تشخیص بیماری مسری استفاده می‌شد که از نقاط ضعف آن می‌توان به فقدان کیت‌های آزمایش، قرار دادن متخصص و بیمار در یک مکان و دقت پایین اشاره کرد. این مطالعه یک چارچوب یادگیری سه مرحله‌ای شامل استخراج‌کننده ویژگی توسط مدل هندسه بصری گروه ۱۶ (VGG16) برای حل مشکلات ناشی از کمبود نمونه‌ها، یک لایه پیچشی سه کاناله و یک طبقه‌بند براساس یک شبکه عصبی سه لایه معرفی می‌کند. نتایج بدست آمده نشان می‌دهد که کووید VGG16 (CoVGG16) دارای دقت (Accuracy) ۹۶,۳۷٪ و ۱۰۰٪، دقت (precision) ۹۶,۵۲٪ و ۱۰۰٪ و دقت (recall) ۹۶,۳۰٪ و ۱۰۰٪ برای پیش بینی COVID-19 در مجموعه‌ی تست دو مجموعه داده است (یک نوع تصاویر مبتنی بر سی تی اسکن و یک نوع عکس با اشعه ایکس که از مخازن Kaggle جمع آوری شده‌اند).

کلمات کلیدی: پیش‌بینی کووید-۱۹، شبکه‌ی عصبی پیچشی، یادگیری انتقالی، بینایی کامپیوتر، پردازش تصویر.