The Performance Evaluation of Transfer Learning VGG16 Algorithm on Various Chest X-ray Imaging Datasets for COVID-19 Classification

Andi Sunyoto[1]*, Yoga Pristyanto[2], Arief Setyanto[3]
Fawaz Alarfaj[4], Naif Almusallam[5], Mohammed Alreshoodi[6]
Computer Science Department, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia[1, 2, 3]
Computer & Information Sciences Department, Imam Mohammad Ibn Saud Islamic University, Kingdom of Saudi Arabia[4, 5]
Department of Natural Applied Science, Applied College, Qassim University, Buraydah, Kingdom of Saudi Arabia[6]

***Abstract*—Early detection of the coronavirus (COVID-19) disease is essential in order to contain the spread of the virus and provide effective treatment. Chest X-rays could be used to detect COVID-19 at an early stage. However, the pathological features of COVID-19 on chest X-rays closely resemble those caused by other viruses. The visual geometry group-16 (VGG16) deep learning algorithm based on convolutional neural network (CNN) architecture is commonly used to detect various pathologies on medical images automatically and may have a role in the detection of COVID-19 on chest X-rays. Therefore, this research is aimed to determine the robustness of the VGG16 architecture on several chest X-ray databases that vary in terms of size and the number of class labels. Nine publicly available chest X-ray datasets were used to train and test the algorithm. Each dataset had a different number of images, class compositions, and interclass proportions. The performance of the architecture was tested using several scenarios, including datasets above and below 5,000 samples, label class variation, and interclass ratio. This study confirmed that the VGG16 delivers robust performance on various datasets, achieving an accuracy of up to 97.99%. However, our findings also suggest that the accuracy of the VGG16 algorithm drops drastically in highly imbalanced datasets.**

Keywords*—Covid-19; Chest X-Ray; CNN; transfer learning; VGG-16

I. Introduction

X-ray images are often used to detect changes in the lungs, such as pneumonia caused by viral infections. Pneumonia is also one of the key indicators of an infection caused by coronavirus disease (COVID-19). However, the manual evaluation of X-ray images is time-consuming and often subjective. Artificial intelligence (AI) could be used to automatically distinguish infected and infection-free patients by extracting specific shapes and spatial features visible on chest X-ray images. Many studies have been carried out using X-ray images to detect Middle East Respiratory Syndrome Coronavirus (MERS CoV) since there are features on chest X-rays and CT that resemble pneumonia manifestations [1]. A convolutional neural network (CNN) model has been developed to identify the nature of the pulmonary modulus on CT images and diagnose pneumonia on chest X-ray images [2].

COVID-19 symptoms include cough, fever, dyspnea, and respiratory problems. In more severe cases, COVID-19 can cause pneumonia, acute respiratory distress, septic shock, failure of internal organs, or even death [4]. Reverse-transcription-polymerase chain reaction (RT-PCR) of samples obtained from either blood or the respiratory system is often used to diagnose COVID-19. Furthermore, due to the highly infectious rate of COVID there is a high demand for this service which leads to further delays to obtain the test results. Therefore, in the emergency department, the initial diagnosis of symptomatic patients is more likely to be done through a plain chest X-ray or CT scan. The early identification of COVID-19 on plain X-rays or CT images is essential to isolate patients and hence minimize the spread of the disease as well as to treat infected patients more effectively.

Bilateral pulmonary parenchymal ground-glass and consolidative pulmonary opacities, with a rounded shape and a peripheral lung distribution, are common chest X-rays in COVID-19 patients [3]–[5]. Pneumonia is also an important indicator of COVID-19. However, these pathological features may closely resemble those caused by other viral infections, which makes it difficult for the radiologist to identify the type of infection. Deep learning algorithms based on convolutional neural network (CNN) architecture could be used to identify specific COVID-19 features on X-ray images.

CNN algorithms are easy to model and reliable. As a result, they are currently the most widely used artificial intelligence (AI) model for the detection of COVID-19 on X-ray images [6]–[13]. We reviewed several studies that made use of the CNN architecture to diagnose COVID-19 on chest X-rays, as shown in Table I. Our findings indicate that the CNN model developed by the visual geometry group with 16 depth layers (VGG16) has been applied in about 50% of the COVID-19 studies [6], [8], [9], [13]–[15]. The VGG16 also performed very well when compared with other established models. However, although studies based on the VGG16 models reported high levels of accuracy, research on COVID-19 is still evolving. Furthermore, most of the studies were based on a single dataset, potentially limiting the generalizability of the model. Therefore, the efficacy of the VGG16 model needs to be tested further on new emerging datasets. Because of these facts, we assessed additionally; we aimed to identify more

*Corresponding Author.

pathological features on chest X-rays indicative of COVID-19 and other infectious diseases. This research contributes by testing the performance of VGG-16 on large, popular datasets and determining its level of accuracy. The results of this study will help to detect indications of COVID-19 more accurately in x-ray images and obtain alternative diagnoses of symptoms similar to those of COVID-19.

II. METHODS

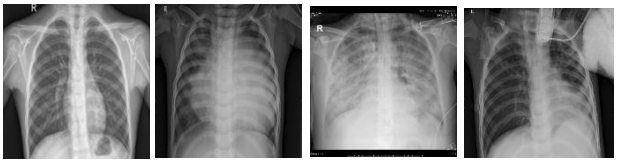
The research framework was divided into four steps: identification of publicly available chest X-ray datasets, image preprocessing, training and finally testing of the VGG16 algorithm, as shown in Fig. 1.

A. Data Collection

Several publicly available image data repositories were searched using the keywords "COVID-19 X-ray". The search yielded 28 datasets composed of chest X-ray images used to diagnose COVID-19. The files within these datasets were reviewed to ensure that they contained chest X-ray imaging files that could be read by the COVID-19 detection system. Only the data set including big data i.e. containing more than 1000 images were included in the study. A total of nine datasets met the inclusion criteria. The final total datasets consisted of a total of 38,181 X-ray images. The images were classified into four categories: normal, pneumonia, viral pneumonia, and COVID-19, as shown in Fig. 2. However, the number of classifiers varied between the different datasets. The dataset source, the total number of images, and the number of classifiers within each dataset are summarized in Table II. Table II shows sample images for each class and the percentage number of images per class.

B. Image Preprocessing

The VGG16 architecture uses a kernel size of 3x3 with an input image of 224x224x3 for the width, height, and channel, respectively [29]. In the preprocessing stage, the VGG16 input was scaled to 224x224 pixels. The images of the classes were randomly extracted, and finally 80% were used to train the algorithm, 20% were used for validation, and the final 10% were used for testing as shown in Fig. 3.



(a) Normal (b) Pneumonia (c) COVID-19 (d) Viral Pneumonia

Fig. 1. Chest X-rays Illustrating the Four Different Classifiers used in the Datasets.

TABLE I. SUMMARY OF RESEARCH STUDIES COMPARING THE ROBUSTNESS OF SEVERAL CNN ARCHITECTURES FOR THE DETECTION OF COVID-19 ON CHEST X-RAYS

Author	Methods	Best Result
[6]	ResNet50, InceptionV3, and VGG16	VGG16
[7]	DenseNet-169+SVM, VGG16, RetinaNet + Mask RCNN, VGG16 and Xception	ResNet50
[8]	VGG16, VGG19, ResNet, DenseNet, and InceptionV3	VGG16
[9]	VGG16, MobileNetV2, Xception, NASNetMobile and InceptionResNetV2	VGG16
[10]	VGG16, ResNet50, and EfficientNetBo	EfficientNetBo
[16]	VGG16	VGG16
[12]	VGG16, DenseNet-161, ResNet-18	ResNet-18
[13]	MobileNet-V2 and VGG16	VGG16
[17]	AlexNet, VGG16, GoogleNet, MobileNet-V2, SqueezeNet, ResNet-34, ResNet-50 and Inception-V3	ResNet-34

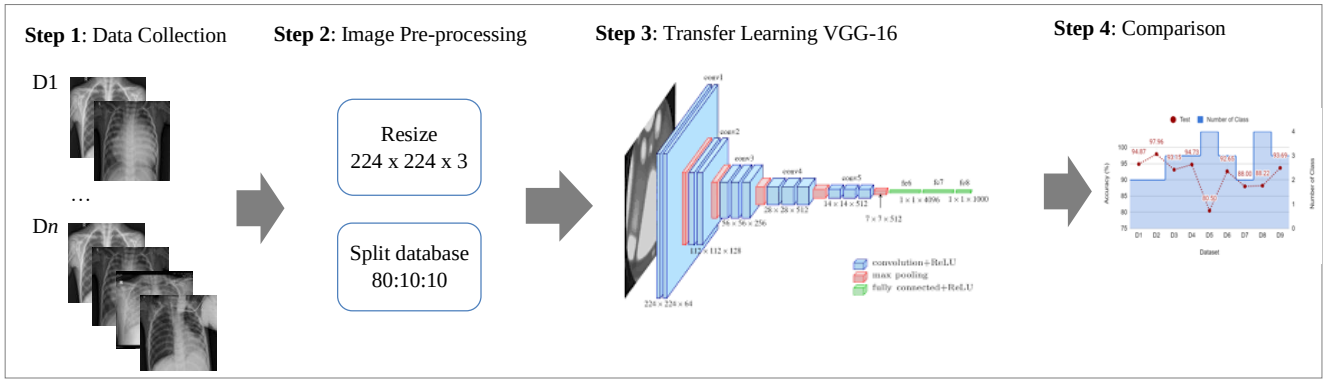


Fig. 2. Research Framework used to Conduct the Study.

TABLE II. DATASETS USED TO EVALUATE THE ROBUSTNESS OF THE VGG16 MODEL

Data	Name	Source	Classes	Files
D1	Chest X-ray Images (Pneumonia)	[18]	Normal:1583 Pneumonia : 4273	5,856
D2	CoronaHack -Chest X-ray-Dataset	[19], [20]	Normal:1576 Pneumonia : 4334	5,910
D3	COVID-19 Radiography Database	[21], [22]	COVID-19:219 Normal:1341 Viral Pneumonia : 1345	2,905
D4	Chest X-ray (COVID-19 & Pneumonia)	[23]–[25]	COVID-19:576 Normal:1583 Viral Pneumonia : 4273	6,432
D5	COVID-19 Detection X-ray Dataset	[18]–[20]	Bacterial Pneumonia:650 COVID-19:60 Normal:880 Viral Pneumonia : 412	2,002
D6	Covid-GAN and Covid-Net mini Chest X-ray	[18]–[20], [26]	COVID-19:461 Normal:1583 Pneumonia:4489	6,533
D7	COVID-19 X-ray Images5	[27]	COVID-19:60 Normal: 880	940
D8	Curated Chest X-ray Image Dataset for COVID-19	[25]	COVID-19:1281 Normal:3278 Pneumonia-Bacterial:3001 Pneumonia-Viral: 1656	6,515
D9	COVID-19 X-ray Dataset with Preprocessed Images	[19], [28]	COVID-19:361 Normal:365 Pneumonia:362	1,088

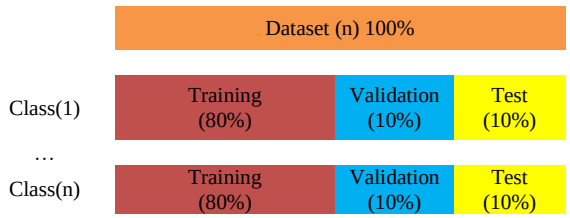


Fig. 3. Distribution of the Dataset for Training, Validation, and Testing.

C. Application of the Convolutional Neural Network

CNN architectures are commonly used in computer vision and involve a convolution operation between the input signal and the filter. An important step in the development of the CNN algorithm involves the use of data pooling and convolution operations. During data pooling, the datasets are downsampled by averaging the data (average pooling) or by obtaining the maximum value (max pooling). In this case, the input signal features were derived from the chest X-ray image, which is commonly represented as n*m*c, whereby *n* and *m* represent the image width and length, respectively, and c represents the color channels. For example, in a typical red-green-blue (RGB) image with a pixel matrix size of 256x100, the input matrix would be defined as 256x100x3. Convolution and pooling operations reduce the complexity by extracting only the important features. For example, an input signal consisting of 75,000 features can easily be reduced into 512 features by applying several convolution layers.

Several CNN-based architectures have been proposed in the last decade. Lenet-5 architecture was the first to propose a CNN-based architecture to solve a simple handwritten digit recognition problem [30]. The work was based on the older concepts of neural networks and back-propagation. The big leap of CNN-based algorithms was enabled by the availability of a large, labeled image dataset called Imagenet [31]. The dataset currently contains around 14 million labeled images, and it was initiated in 2009 by an artificial intelligence lab at Stanford University. Alexnet is the second most well-known CNN architecture that had won the Imagenet Large Scale Visual Recognition Challenge in 2014 [32].

The VGG16 model was initially proposed by Simonyan et al. [29] and it secured first place for object localization and second place for object classification in the Imagenet Large-Scale Visual Recognition Challenge 2014 (ILSVRC 2014). Since then, numerous other CNN architectures have been proposed including Inception net [33], ResNet [34], Inception-v4 and Inception-ResNet [33], Mobilenet [35], MobileNet V2 [36], EfficientNet [37] and XceptionNet [38].

The building block of the CNN architecture consists of two fundamental components: a convolution layer and a pooling layer. The filter size, padding, stride, activation function, and connection between layers can be manipulated to improve the performance of the algorithm. In order to improve the performance of the algorithm, the raw image signal has to be converted into a more straightforward representation before applying the classification task. The large number of images available on the Imagenet database could be used to train and compare the performance of various algorithms. Once an architecture has been trained and tested on the Imagenet images, researchers can ascertain the performance of their architectures and determine the weight parameters.

Transfer learning is a commonly used method in computer vision that applies the knowledge gained from the training of a network to solve a specific problem to a new similar scenario. This eventually reduces the time required for the training process allowing for the development of accurate algorithms in a shorter amount of time [29]. For example, if a pre-trained network previously developed to classify 1000 classes is now used for binary classification, the top layer (last layer) is adjusted so that the output is changed from 1000 into two classes only. The weight parameters also have to be updated in all network layers or for some of the layers. However, the learning process does not begin from scratch; instead, it starts with the pre-trained weight that has been applied to solve the previous problem.

Most model architectures such as VGG16, InceptionNet, mobilenet, and XceptionNet were trained on a large dataset such as Imagenet. As a result of the high computational cost incurred during training, an improved model was developed. Canziani et al. [39] conducted a comprehensive analysis of the performance of pre-trained models on computer vision challenges using data from the Imagenet [31] database. In computer vision, the transfer approach is popular because it enables the generation of accurate models in a shorter amount of time [40].

A typical CNN classification task has a feature generator and a classifier. The feature map generator input are the raw images, followed by a stack convolution and pooling layers. The main goal of the feature generator is to produce an array representing the input image in a smaller amount of data. On the other hand, the classifier's task is to categorize the feature into certain target classes. This task can be performed by classic classifiers such as support vector machines and decision trees. Another option is to place the fully connected layers on top of the feature generator. A fully connected layer is one whose neurons fully connect to all activations in the previous layer. The number of layers in the fully connected layer is significant and could be optimized by the researcher manually. The depth of the fully connected layers should be taken into account as it relates to the overfitting problem of the entire network. The deep learning approach independently computes the important input features during the learning process. Unlike the classical AI algorithm whereby the features extracted by the algorithm are based on the objective of the classification task and the image input, deep learning models learn hierarchical features by adjusting weight parameters on the CNN-based feature generator. The pattern of the input is then captured by the network and is then used as the input of the classifier. The pattern for a specific problem domain is accurately recorded as the weight parameter value. In transfer learning, the set of values can be applied to another specific problem domain.

Transfer learning involves two key steps. The first step involves selecting a pre-trained model, such as VGG16 [29], InceptionV3 [41], and ResNet5 [34], to fit the target problem. The second step involves the identification of features based on the size of the dataset and the similarities between the pre-trained dataset and the dataset we used. The comparison between the pre-trained dataset and our characteristics dataset could result in one of the following four transfer learning problems whereby the new problem dataset are:

1) large but dissimilar from the pre-trained dataset
2) large and highly similar to the pre-trained dataset
3) small and highly similar to the pre-trained dataset
4) small and dissimilar from the pre-trained dataset

In deep learning, a dataset consisting of 1000 labeled images per class is considered to be small. Dataset similarity refers to the availability of the same problem subset in the pre-trained dataset. For example, if the task is to recognize dogs and birds using a pre-trained network that has been trained on an Imagenet dataset since the dataset contains dog and bird classes, we can consider that the dataset is highly similar. However, in this paper, we classified the visual pathological features of COVID-19 patients through visible X-ray images which were not available in the Imagenet dataset. Hence, our transfer learning problem was due to a small dataset dissimilar from the pre-trained dataset.

This study employed the VGG16 [41], a pre-trained CNN-based architecture that consists of 16 CNN layers. The VGG architecture has already been applied in many medical image classification tasks [42]. This research provides an evaluation and comparison of the performance of various architectures in detecting COVID-19 on various chest X-ray datasets.

The comprehensive evaluation was made based on the accuracy achieved by the VGG16 architecture when applied to different datasets. Since the number of classes within the dataset varied, the multiclass confusion matrix was used to determine the robustness of the VGG16. Table III illustrates a confusion matrix with *n* classes, including the experimental results obtained for each scenario.

TABLE III. MULTICLASS CONFUSION MATRIX

		Predicted Number			
		Class 1	***Class 2***	**…**	***Class n***
Actual Number	***Class 1***	x_{11}	x_{12}	…	x_{1n}
	Class 2	x_{21}	x_{22}	…	x_{2n}
	…	…	…	…	…
	Class n	x_{n1}	x_{n2}	…	x_{nn}

Furthermore, the performance of the VGG16 model was quantified by calculating the total numbers of false-negative (TFN), false positive (TFP), true negative (TTN), total true positive (TTP) for each class i based on the general equations 1 to 4.

$$TFN_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} \quad (1)$$

$$TFP_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ji} \quad (2)$$

$$TTN_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \sum_{\substack{k=1 \\ k \neq i}}^{n} x_{jk} \quad (3)$$

$$TTP_i = \sum_{j=1}^{n} x_{ij} \quad (4)$$

In addition, the precision (P), recall (R), and specificity (S) for each class *i* were also calculated as shown in equations 5, 6 and 7. The total accuracy and F1-Score were calculated as shown in equations 8 and 9, respectively.

$$P_i = \frac{TTP_{all}}{TTP_{all} + TFP_i} \quad (5)$$

$$R_i = \frac{TTP_{all}}{TTP_{all} + TFN_i} \quad (6)$$

$$S_i = \frac{TTN_{all}}{TTN_{all} + TFP_i} \quad (7)$$

$$Accuracy = \frac{TTP_{all}}{Total\ Number\ of\ Testing} \quad (8)$$

$$F1 - Score = \frac{TTP_{all}}{Total\ Number\ of\ Testing} \quad (9)$$

III. RESULT AND DISCUSSION

In this study, we made use of nine open databases to measure the performance of the CCN transfer learning model, VGG16, to detect pneumonia and COVID-19 cases. Each database contained more than 1000 images and a total of 38,181 images were evaluated. The datasets were divided according to the number of images and classifiers. The robustness of the VGG16 model was then tested on datasets that were.

1) with more than 5000 images.
2) with less than 5000 images.

3) with a different number of classes.

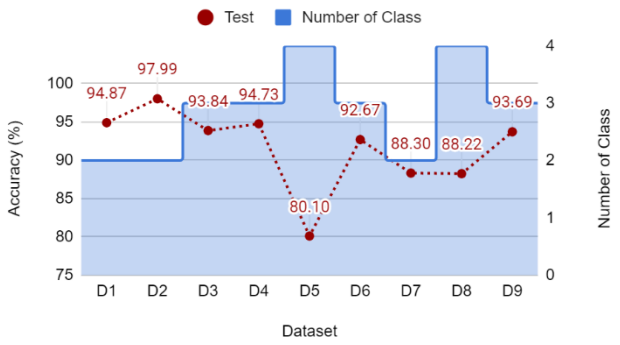


Fig. 4. Relationship between the Number of Classes in the Dataset and the Accuracy of the Algorithm.

Fig. 4 shows the relationship between the number of classes in the dataset and the resulting performance of the algorithm in terms of accuracy. The accuracy of the algorithm ranged from 80.10% to 97.99%. The algorithm performed better in datasets with three classifiers when compared with datasets with two classifiers. The algorithm had the worst performance in datasets with four classifiers.

The performance of the VGG16 algorithm was validated by splitting the dataset into three categories; training (80%), validation (10%), and test (10%). Fig. 5 illustrates the results of the confusion matrix from data test, while Table IV compares the accuracy, precision, recall, and F1-score of the VGG16 model in all of the nine databases evaluated in this study.

The testing accuracy analysis of the VGG16 algorithm in datasets containing more than 5,000 images is illustrated in Fig. 6. For this evaluation, we compared the accuracy of the VGG16 algorithm in D1, D2, D4, D6, and D8 datasets, which contained 5,856, 5,910, 6,432, 6,533, and 6,515 images, respectively. Following testing, the VGG16 algorithm in datasets with more than 5,000 images ranged from 97.99% to 88.22%. The findings of this analysis indicate that for both the validation and testing the detection accuracy of the algorithm decreased as the number of images increased.

The accuracy of the VGG16 algorithm on datasets containing less than 5000 images is illustrated in Fig. 7. For this analysis, the D7, D9, D5, and D3 datasets were used, which consisted of 940, 1,088, 2,002, and 2,905 images, respectively. For the testing data, the mean accuracy of the VGG16 algorithm was 94.31% with a range of 80.10-93.84%. As shown in Fig. 7, the difference in accuracy within each dataset was relatively small except for the D5 dataset, which showed an accuracy of 80.10%. However, further analysis showed that the low level of accuracy in the D5 dataset was caused by the very high-class difference ratio within this dataset.

Fig. 8 illustrates the performance of the VGG16 algorithm on datasets with two, three, and four classes. As evident in Fig. 8, the datasets with two and three classes did not differ much in terms of accuracy, but when a dataset has four classes, the accuracy decreases by 9.57%.

Based on the results of this study, we can conclude that the performance of the VGG16 algorithm is affected by the number of images and the number of classes within the dataset: For datasets with more than 5000 images, the accuracy of the algorithm decreased as the number of images in the dataset increased. The VGG16 model achieved a mean accuracy of 93.7%. Compared with previous studies, the VGG16 algorithm performed well despite its relatively simple architecture.

For datasets containing less than 5000 images, the number of images did not impact the algorithm's accuracy except for the D5 dataset. However, the lower performance of the algorithm in D5 was attributed to the larger class ratio within this dataset. The number of classes within a dataset affected the accuracy of the algorithm, whereby the VGG16 model performed worse in datasets with four classes. However, further research is recommended to test the efficacy of the transfer learning VGG16 model on the detection accuracy of COVID-19.

Compared to other popular transfer learning, the advantage of the VGG16 architecture is that it has only six layers in depth. The small layer makes the identification process fast. This fast time allows it to be applied to devices that have low specifications and are mobile. In real conditions, the dataset is not ideal, with different numbers and ratios between different classes. Based on the experiment, VGG16 solved cases of COVID-19 data with the characteristics of having a small class due to imbalanced data conditions. The limitation of VGG16 occurs in the unbalanced condition dataset, which has a large gap ratio (database D5 in Table II), and the number of classes is greater than four this show in Fig. 8.

TABLE IV. PERFORMANCE OF THE VGG16 ALGORITHM FOR ALL THE NINE DATABASES EVALUATED IN THE STUDY

Data	Name	Acc.	Prec.	Rec.	F1-Score
D1	Chest X-ray Images (Pneumonia) [18]	94.87	94.92	94.72	94.82
D2	CoronaHack -Chest X-ray-Dataset [19], [20]	97.99	97.65	97.29	97.47
D3	COVID-19 Radiography Database [21], [22]	93.84	90.48	88.68	89.57
D4	Chest X-ray (COVID-19 & Pneumonia) [23]–[25]	94.73	95.17	89.81	92.41
D5	COVID-19 Detection X-ray Dataset [18]–[20]	80.10	80.78	79.91	80.34
D6	Covid-GAN and Covid-Net mini Chest X-ray [18]–[20], [26]	92.67	89.43	85.75	87.55
D7	COVID-19 X-ray Images [27]	88.30	85.98	66.03	74.7
D8	Curated Chest X-ray Image Dataset for COVID-19 [25]	88.22	88.68	86.45	87.55
D9	COVID-19 X-ray Dataset With Preprocessed Images [19], [28]	93.69	93.69	94.03	93.86

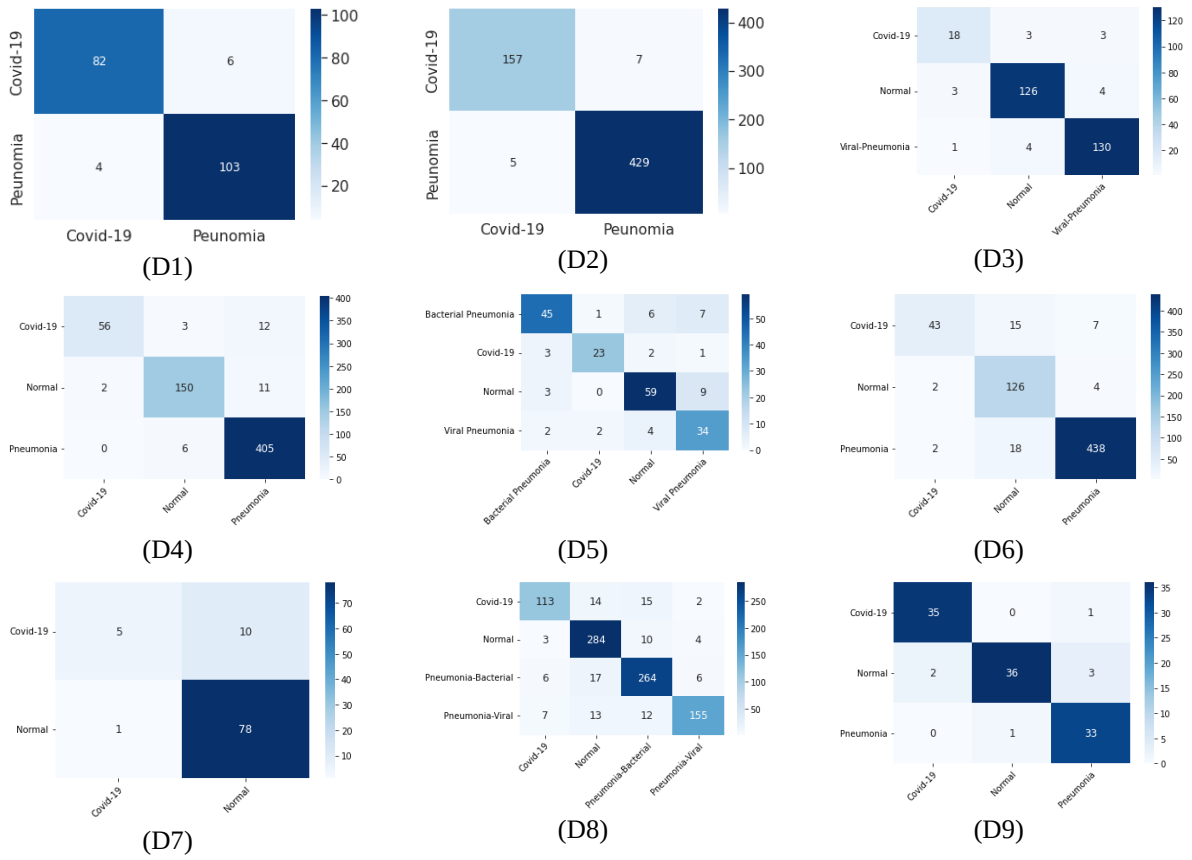


Fig. 5. Confusion Matrix Illustrating the Robustness of the VGG16 Algorithm in Different Datasets.

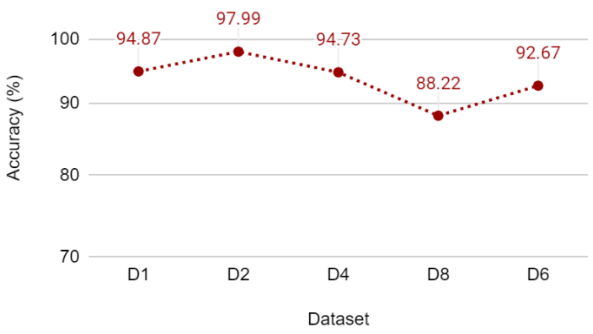


Fig. 6. Accuracy from Data Test of the VGG16 Algorithm in Datasets Containing more than 5,000 Images.

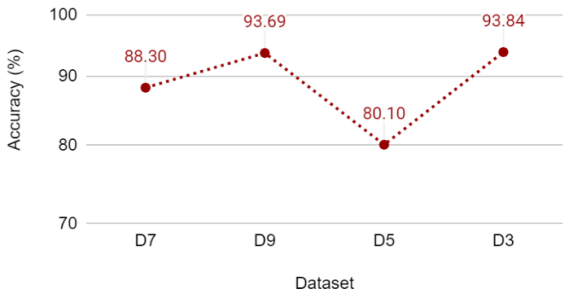


Fig. 7. Accuracy from Data Test of the VGG16 Algorithm in Datasets Containing Less than 5,000 Images.

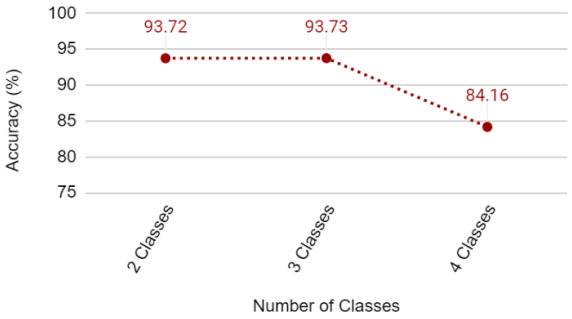


Fig. 8. Performance of the VGG16 Algorithm based on the Number of Classes.

IV. Conclusion

The aim of this study was to assess the performance of the VGG16 algorithm on different datasets. The experimental results confirmed the high accuracy of the VGG16 algorithm in detecting COVID-19. The study also confirmed the robustness of the VGG16 architecture when applied to datasets with various image numbers, classes, and class ratios on chest X-rays. However, in this study, we did not evaluate the impact of high-class ratios on the performance of the VGG16 algorithm. However, the class imbalance problem can easily be resolved via the application of data augmentation and class balancing techniques.

ACKNOWLEDGMENT

We would like to thank TopEdit (www.topeditsci.com) for its linguistic assistance during the preparation of this manuscript.

Funding statement: This research was funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University through Research Group no. RG-21-51-01.

Conflicts of interest: The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] A. Hamimi, "MERS-CoV: Middle East respiratory syndrome corona virus: Can radiology be of help? Initial single center experience," *The Egyptian Journal of Radiology and Nuclear Medicine*, vol. 47, no. 1, pp. 95–106, Mar. 2016, doi: 10.1016/j.ejrnm.2015.11.004.

[2] J. Choe, S. M. Lee, K.-H. Do, G. Lee, J.-G. Lee, *et al.*, "Deep Learning–based Image Conversion of CT Reconstruction Kernels Improves Radiomics Reproducibility for Pulmonary Nodules or Masses," *Radiology*, vol. 292, no. 2, pp. 365–373, Aug. 2019, doi: 10.1148/radiol.2019181960.

[3] D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, *et al.*, "Clinical Characteristics of 138 Hospitalized Patients With 2019 Novel Coronavirus–Infected Pneumonia in Wuhan, China," *JAMA*, vol. 323, no. 11, p. 1061, Mar. 2020, doi: 10.1001/jama.2020.1585.

[4] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, *et al.*, "Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China," *The Lancet*, vol. 395, no. 10223, pp. 497–506, Feb. 2020, doi: 10.1016/S0140-6736(20)30183-5.

[5] M. Chung, A. Bernheim, X. Mei, N. Zhang, M. Huang, *et al.*, "CT Imaging Features of 2019 Novel Coronavirus (2019-nCoV)," *Radiology*, vol. 295, no. 1, pp. 202–207, Apr. 2020, doi: 10.1148/radiol.2020200230.

[6] S. Guefrechi, M. Ben Jabra, A. Ammar, A. Koubaa, and H. Hamam, "Deep learning based detection of COVID-19 from chest X-ray images," *Multimedia Tools and Applications*, Jul. 2021, doi: 10.1007/s11042-021-11192-5.

[7] A. Manickam, J. Jiang, Y. Zhou, A. Sagar, R. Soundrapandiyan, *et al.*, "Automated pneumonia detection on chest X-ray images: A deep learning approach with different optimizers and transfer learning architectures," *Measurement*, vol. 184, p. 109953, Nov. 2021, doi: 10.1016/j.measurement.2021.109953.

[8] K. Sahinbas and F. O. Catak, "Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images," in *Data Science for COVID-19*, Elsevier, 2021, pp. 451–466. doi: 10.1016/B978-0-12-824536-1.00003-4.

[9] A. K. Rangarajan and H. K. Ramachandran, "A preliminary analysis of AI based smartphone application for diagnosis of COVID-19 using chest X-ray images," *Expert Systems with Applications*, vol. 183, p. 115401, Nov. 2021, doi: 10.1016/j.eswa.2021.115401.

[10] T. Zebin and S. Rezvy, "COVID-19 detection and disease progression visualization: Deep learning on chest X-rays for classification and coarse localization," *Applied Intelligence*, vol. 51, no. 2, pp. 1010–1021, Feb. 2021, doi: 10.1007/s10489-020-01867-1.

[11] R. K. Singh, R. Pandey, and R. N. Babu, "COVIDScreen: explainable deep learning framework for differential diagnosis of COVID-19 using chest X-rays," *Neural Computing and Applications*, Jan. 2021, doi: 10.1007/s00521-020-05636-6.

[12] A. Shelke, M. Inamdar, V. Shah, A. Tiwari, A. Hussain, *et al.*, "Chest X-ray Classification Using Deep Learning for Automated COVID-19 Screening," *SN Computer Science*, vol. 2, no. 4, p. 300, Jul. 2021, doi: 10.1007/s42979-021-00695-5.

[13] H. Swapnarekha, H. S. Behera, D. Roy, S. Das, and J. Nayak, "Competitive Deep Learning Methods for COVID-19 Detection using X-ray Images," *Journal of The Institution of Engineers (India): Series B*, Apr. 2021, doi: 10.1007/s40031-021-00589-3.

[14] R. M. James and A. Sunyoto, "Detection Of CT - Scan Lungs COVID-19 Image Using Convolutional Neural Network And CLAHE," in *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, Nov. 2020, pp. 302–307. doi: 10.1109/ICOIACT50329.2020.9332069.

[15] N. Narayan Das, N. Kumar, M. Kaur, V. Kumar, and D. Singh, "Automated Deep Transfer Learning-Based Approach for Detection of COVID-19 Infection in Chest X-rays," *IRBM*, Jul. 2020, doi: 10.1016/j.irbm.2020.07.001.

[16] M. Singh, S. Bansal, S. Ahuja, R. K. Dubey, B. K. Panigrahi, *et al.*, "Transfer learning–based ensemble support vector machine model for automated COVID-19 detection using lung computerized tomography scan data," *Medical & Biological Engineering & Computing*, vol. 59, no. 4, pp. 825–839, Apr. 2021, doi: 10.1007/s11517-020-02299-2.

[17] S. R. Nayak, D. R. Nayak, U. Sinha, V. Arora, and R. B. Pachori, "Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study," *Biomedical Signal Processing and Control*, vol. 64, p. 102365, Feb. 2021, doi: 10.1016/j.bspc.2020.102365.

[18] D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, *et al.*, "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," *Cell*, vol. 172, no. 5, pp. 1122-1131.e9, Feb. 2018, doi: 10.1016/j.cell.2018.02.010.

[19] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, *et al.*, "COVID-19 Image Data Collection: Prospective Predictions Are the Future," Jun. 2020.

[20] J. P. Cohen, P. Morrison, and L. Dao, "COVID-19 Image Data Collection," Mar. 2020.

[21] M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, *et al.*, "Can AI Help in Screening Viral and COVID-19 Pneumonia?," *IEEE Access*, vol. 8, pp. 132665–132676, 2020, doi: 10.1109/ACCESS.2020.3010287.

[22] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, *et al.*, "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images," *Computers in Biology and Medicine*, vol. 132, p. 104319, May 2021, doi: 10.1016/j.compbiomed.2021.104319.

[23] G. Maguolo and L. Nanni, "A Critic Evaluation of Methods for COVID-19 Automatic Detection from X-Ray Images," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.12823.

[24] E. Tartaglione, C. A. Barbano, C. Berzovini, M. Calandri, and M. Grangetto, "Unveiling COVID-19 from CHEST X-Ray with Deep Learning: A Hurdles Race with Small Data," *International Journal of Environmental Research and Public Health*, vol. 17, no. 18, p. 6933, Sep. 2020, doi: 10.3390/ijerph17186933.

[25] U. SAIT, L. k VGokul, T. Prajapati, SunnyBhaumik RahulKumar, and K. SanjanaBHalla, "Curated Dataset for COVID-19 Posterior-Anterior Chest Radiography Images (X-Rays)," p. SAIT, UNAIS; k v, Gokul Lal; Prajapati, Sunny; Bha, doi: 10.17632/9xkhgts2s6.1.

[26] C. D. Corporation, "Actualmed COVID-19 Chest X-ray Dataset Initiative," *Canada and Vision and Image Processing Research Group, University of Waterloo*, 2020. https://github.com/agchung/Actualmed-COVID-chestxray-dataset.

[27] D. Uddipta, "COVID-19 X-ray Images5." https://www.kaggle.com/uddiptadas/covid19-xray-images5.

[28] A. Gupta, Anjum, S. Gupta, and R. Katarya, "InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using Chest X-ray," *Applied Soft Computing*, vol. 99, p. 106859, Feb. 2021, doi: 10.1016/j.asoc.2020.106859.

[29] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014.

[30] L. Cun, L. Cun, B. Boser, J. S. Denker, D. Henderson, *et al.*, "Handwritten Digit Recognition with a Back-Propagation Network," *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, vol. 2, pp. 396--404, 1990, Accessed: Dec. 01, 2021. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.5076.

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, *et al.*, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on*

Computer Vision and Pattern Recognition, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[33] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," Feb. 2016.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Dec. 2015, doi: 10.1109/CVPR.2016.90.

[35] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017.

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[37] M. Tan and Q. v Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks."

[38] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1800–1807. doi: 10.1109/CVPR.2017.195.

[39] A. Canziani, A. Paszke, and E. Culurciello, "An Analysis of Deep Neural Network Models for Practical Applications," May 2016.

[40] [40] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017, doi: 10.1162/neco_a_00990.

[41] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *ArXiv*, vol. abs/1502.0, 2015.

[42] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, *et al.*, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, Dec. 2017, doi: 10.1016/j.media.2017.07.005.