Fast Hybrid Deep Neural Network for Diagnosis of COVID-19 using Chest X-Ray Images

Hussein Ahmed Ali[1], Nadia Smaoui Zghal[2], Walid Hariri[3], Dalenda Ben Aissa[4]
Microwave Electronics-Research laboratory Faculty of Sciences of Tunis, University Tunis El-Manar, Tunis El-Manar, Tunis[1, 4]
Control and Energy-Management Laboratory-(CEM Lab) ENIS, University of Sfax, Sfax, Tunisia[2]
LABGED Laboratory-Computer Science Department-Badji Mokhtar, Annaba University, Annaba, Algeria[3]

Abstract—**In the last three years, the coronavirus (COVID-19) pandemic put healthcare systems worldwide under tremendous pressure. Imaging techniques, such as Chest X-Ray (CXR) images, play an essential role in diagnosing many diseases (for example, COVID-19). Recently, intelligent systems (Machine Learning (ML) and Deep Learning (DL)) have been widely utilized to identify COVID-19 from other upper respiratory diseases (such as viral pneumonia and lung opacity). Nevertheless, identifying COVID-19 from the CXR images is challenging due to similar symptoms. To improve the diagnosis of COVID-19 using CXR images, this article proposes a new deep neural network model called Fast Hybrid Deep Neural Network (FHDNN). FHDNN consists of various convolutional layers and various dense layers. In the beginning, we preprocessed the dataset, extracted the best features, and expanded it. Then, we converted it from two dimensions to one dimension to reduce training speed and hardware requirements. The experimental results demonstrate that preprocessing and feature expansion before applying FHDNN lead to better detection accuracy and reduced speedy execution. Furthermore, the model FHDNN outperformed the counterparts by achieving an accuracy of 99.9%, recall of 99.9%, F1-Score has 99.9%, and precision of 99.9% for the detection and classification of COVID-19. Accordingly, FHDNN is more reliable and can be considered a robust and faster model in COVID-19 detection.**

Keywords—COVID-19; Chest X-ray (CXR); Deep Learning (DL); Convolutional Neural Network (CNN)

I. Introduction

The novel coronavirus disease (COVID-19) is a current pandemic that has sparked widespread concern worldwide. This virus causes severe respiratory illness [1]. Currently, COVID-19 is recognized as the worst disease on the earth. Usually, chest X-ray (CXR) imaging is used for radiographic examination of the suspected cases. In severely impacted or limited resource locations, CXR imaging is favored because of its accessibility, cheap cost, and speedy findings. Via Lenard and Crookes tubes, the first X-ray was coined by Wilhelm Conrad in 1895. The rapid development of the x-ray system and its low cost made it available in most clinical testing [2]. Patients can quickly get CXR images in their homes or quarantine facilities since CXR facilities are available even in the most distant locales. Recently, CXR images based on Artificial Intelligence (AI) have been utilized widely to detect COVID-19 instances [3, 4]. Given the fast spread of COVID-19, however, such testing might reduce the effectiveness of pandemic prevention and control. Therefore, AI techniques, such as Deep Learning (DL), are viable solutions for autonomous diagnosis due to their promising processing of visual data and a vast array of medical images [4].

This section investigates supreme papers that utilized Convolutional Neural Networks (CNN) and other DL architectures to diagnose COVID-19 via CXR images. The CXR image testing factor is determined by (DL) plans and consists mostly of measurements such as COVID-19 segmentation, data correction, and model training [5]. Manually, radiologists often perform COVID-19 classification. However, it is time-consuming, error-prone, and exhaustive since radiologists are expected to diagnose many COVID-19 patients. Furthermore, despite the dramatic increase in COVID-19 patients, the lack of qualified radiologists to make reliable diagnosis continues to be a major problem [6].

On the other hand, healthcare professionals take a sample from the nose to find whether it is a COVID-19 case. Nevertheless, the manual outcome of the test could be a false negative depending on the timing and quality of the test sample. The maximum accuracy, of this testing method, for COVID-19 infections, is 71% [6].

CXR is an alternate important diagnostic method for COVID-19 detection. However, if the CXR images require clarification, COVID-19 is sometimes diagnosed as another disorder. Due to faulty diagnoses, patients frequently receive incorrect medicines, complicating their health. This critical health issue to diagnose COVID-19 has inspired researchers to create more precise and automated CXR-based diagnostic approaches.

Machine learning (ML) methods and the most intense learning have been considered more accurate in identifying COVID-19 [7, 8]. The authors in [9] used DL methods-based CXR images to diagnose COVID-19. Although the presented models in the literature were validated with several CXR images, particularly the positive case images of COVID-19 cases, there was a class imbalance issue in previous studies. Additionally, the authors utilized the transfer learning approach in their model. The scholars applied dataset preprocessing to convert it into a gray level. Next, change the image intensity, resize and extract the best features, and use the transfer learning approach. The authors in [9] focused on getting high COVID-19 classification accuracy. This means applying the same preprocessing on CXR images developing them and then designing a new model in DL for COVID-19 diseases. Several authors (such as in [4, 10]) utilized only CNN to classify COVID-19 patients and didn’t achieve more accurate accuracy.

However, CNN alone is insufficient for reliable COVID-19 detection [6, 11]. Therefore, researchers explored the limitations of the existing models to propose crucial solutions that can increase detection accuracy.

Since the outbreak of COVID-19, the majority of state-of-the-art methods have primarily focused on using transfer learning techniques to implement their systems as shown in the recent survey published in [3]. One of the main challenges in using transfer learning techniques for detecting COVID-19 is the amount of time it takes to apply these methods to large datasets, especially when dealing with CXR images. This is because the model needs to be fine-tuned by retraining it on the new data, which can be time-consuming even with powerful computing resources. To address this issue, a study proposed using several image preprocessing techniques before inserting deep CNN. These techniques involve converting the images to grayscale, adjusting the image intensity, resizing and extracting the best features, and expanding features. This preprocessing transforms the data from two dimensions to one dimension, which can speed up implementation. Additionally, a unique fast deep learning proposal design was introduced to improve COVID-19 classification accuracy while reducing execution time and storage requirements.

Going forward, this article is organized as follows. Section II reviews some of the important proposed models in COVID-19 identification. Section III presents our proposed methodology. Section IV describes the utilized dataset. Section V explains the conducted preprocessing operations. Section VI introduces Linear Discriminant Analysis (LDA) that we use to extract CXR features. Section VII describes feature expansion as a last preprocessing for applying the deep proposal CNN. Section VIII presents the Fast Hybrid Deep Neural Network (FHDNN) and its summary layers. Section IX evaluates the proposal's performance and compares it with related counterparts. Finally, Section X concludes the paper and identifies some future directions.

II. Related Works

Recently, the DL technique has been adopted for analyzing medical images because of its reliable outcomes [12]. Due to no constraint on the dataset employed, DL is suitable for making predictions based on training data and information. [13]. Currently, the existing methods suffer from some crucial issues, such as being time-consuming and expensive. Therefore, developing DL technology to detect COVID-19 faster and more accurately has become necessary. CXRs and CT scans can help achieve this. However, CXRs are less expensive than CT scans, thus, they are preferable.

To classify COVID-19 cases, Aslan, M. F. et al. [14] suggested a DenseNet-SVM structure and utilized binary classification techniques. While the model achieved an accuracy of 96.29%, the author also used eight models of SVM-based CNN and an average accuracy of 95.21%. Alaska, T. B. et al. [15] utilized the LSTM model and validated their work using a 10-fold cross-validation strategy. The results of the models showed excellent classification performance and achieved an accuracy greater than 84%. Additionally, the recall and AUC scores reached 99.43% and 0.625, respectively. Furthermore, using Holdout validation, the CNN-LSTM achieved better performance, where the accuracy, recall, and AUCs cores went 92.3%, 93.68%, and 0.90 respectively. For detecting and diagnosing human lung disorders, four models were proposed by Ibrahim et al. [16]. The models have been used to classify three cases: COVID-19, lung cancer, and Pneumonia. CNN-VGG19 (GRU with ResNet152V2) achieved an accuracy of 98.05% (96.09%). The VGG-inspired model identified the COVID-19 patients successfully from other cases (regular and pneumonia). For multi-class and binary classification, evaluation metrics accuracy, F1-score, recall, precision, and AUC were 97%, 0.95, 95%, 95% and 0.95, and 98%, 0.97, 97%, 97%, and 0.97, respectively [16].

As mentioned earlier, CXR is one of the essential methods to diagnose pneumonia worldwide [17] since it is a fast, cheap [18], and standard clinical method [19-21]. In [22-24], scholars suggested a DL method-based dataset of CXR images. The authors utilized domain DL within environment-deep CNN. The author proposed that the technique was developed specifically for identifying COVID-19 patients from CXR data. For three classes, an overall accuracy of 96.48% was obtained. However, for four class classifications, the COVDC-Net method achieved accuracy reached 90.22%. The authors used 3,884 (1,414) images for three (four) classes. Using image analysis, early detection of the COVID-19 system was proposed in [25]. Using DL method-based CXR image; the authors calculated the impacts of COVID-19 on people. They utilized five pre-trained CNN-based models: ResNet152, ResNet101, ResNet50, Inception-ResNetV2, and InceptionV3. The highest classification accuracy was determined by ResNet50, reaching 96.1% for the dataset (comprised of two classes).

Similarly, the authors in [26] used transfer learning methods: VGGNet-19, VGG16, VGG19 LeNet-5, MobileNetV2, and the Fusion model. The researchers collected CXR images from various sources (from different classes of chests humans) to detect COVID-19 disease. On the other hand, the study [27] used a pre-trained approach called VGG16-CNN to detect COVID-19 cases using CXR images. While the study achieved an accuracy of 97.50% for multiple classifications, the authors tested the models and found varying proportions, such as an F1-score rate, precision rate, recall rate, and overall accuracy. The authors utilized CNN and pre-trained models for feature extraction and achieved several classification accuracies. Accordingly, the alone use of CNN is insufficient for better identifying COVID-19 cases. Fewer images and no preprocessing steps are the causes of poor performance. One of the major difficulties in employing transfer learning approaches to detect COVID-19 is the time required to apply these techniques to big datasets. Due to the model's need for fine-tuning to alter its parameters, this process takes a long time when dealing with big amounts of CXR picture data. This requires retraining the model on the new data, which can take a significant amount of time, even with powerful computing resources. As shown in the survey [4], many methods served to evaluate pre-trained deep CNN models including AlexNet, VGG-16, GoogleNet, MobileNet-V2, SqueezeNet, ResNet-34, ResNet-50, and Inception-V3 to distinguish COVID-19 from normal cases. Many hyperparameters tuning have been applied to find the best batch size, learning rate, optimizer, and a number of epochs. The obtained results were less than our fast HFDNN with high time consumption. Finally, the highest classification accuracy of recent papers reached 99%, obtained using the InstaCovNet-19 DL method [28]. Based on the studies mentioned above, they suffer from some significant limitations:

- Imbalance classes in the utilized datasets.
- Applied classification images using DL without image preprocessing techniques (such as Linear Discriminant Analysis (LDA)). For example, equalization of the intensity (use Histogram Equalization (HE)), removing noise from the images, resizing it, extracting the best features, then feature expansion.
- The evaluation metrics (i.e., accuracy, precision, recall, and F1-score) must improve.
- DL models typically need a lot of training time.
- More hardware is required for DL model execution.

Therefore, before inserting deep CNN, we applied several image preprocessing techniques to tackle the above drawbacks in this study. In the first step of dataset preprocessing, we convert the images into the gray level. Next, the image intensity is adjusted, and the best features are resized and extracted. Lastly, perform features expansion. This processing may ensure that the data is transformed from two dimensions to one dimension, which will speed up implementation. Furthermore, a unique fast deep learning proposal design can improve COVID-19 classification accuracy in less execution time and storage.

III. Methodology

The COVID-19 outbreak has necessitated rapid action against a massive threat to humans. To diagnose COVID-19, understanding and classifying CXR images are crucial. Deep CNN technologies, which aid radiologists, improve the effectiveness of imaging tools. Our goal in this research is to design a novel CNN model capable of achieving an almost perfect classification accuracy with fast execution time and low storage use. Here, we describe the architecture of our proposed method, namely FHDNN as illustrated in Fig. 1.

The proposed CNN model was built and trained on the COVID-19 four class dataset using the Python programming language and the Keras library. The work was developed using an I7- 4710H Intel processor, 2.50 GHz CPU, 8.0 GB RAM, and an NVIDIA Quadro K2100M graphic card.

After extracting features using LDA and feature expansion, the proposed FHDNN is designed to work with eighteen one-dimensional features. It takes a package of input features, processes them, and classifies them into specific categories. The FHDNN consists of multiple layers that are interlinked and arranged in a way that enhances its performance. When processing each input data set, the eighteen features pass through a series of convolution layers, pooling layers, and fully connected layers (FC) to classify objects with probabilistic values between zero and one.

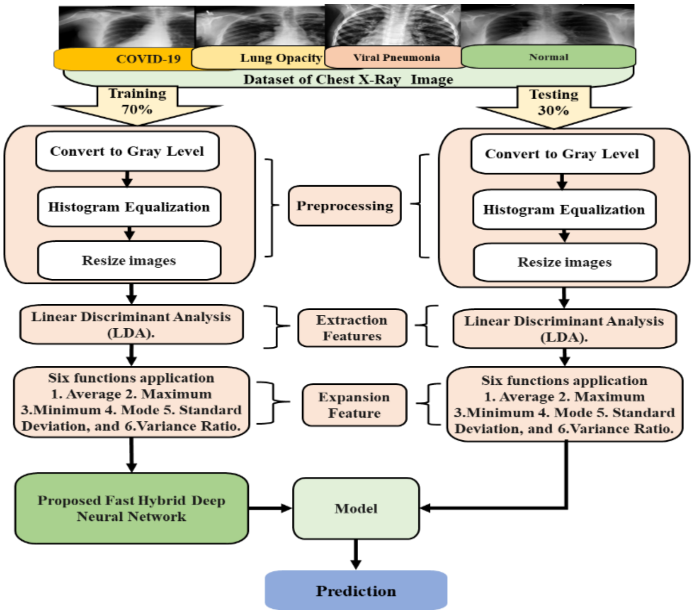
COVID-19
Lung Opacity
Viral Pneumonia
Normal
Dataset of Chest X-Ray Image
Training 70%
Testing 30%
Convert to Gray Level
Histogram Equalization
Resize images
Preprocessing
Convert to Gray Level
Histogram Equalization
Resize images
Linear Discriminant Analysis (LDA).
Extraction Features
Linear Discriminant Analysis (LDA).
Six functions application 1. Average 2. Maximum 3.Minimum 4. Mode 5. Standard Deviation, and 6.Variance Ratio.
Expansion Feature
Six functions application 1. Average 2. Maximum 3.Minimum 4. Mode 5. Standard Deviation, and 6.Variance Ratio.
Proposed Fast Hybrid Deep Neural Network
Model
Prediction

Fig. 1. The flowchart of the proposed system for COVID-19 diagnosis.

To conduct the training process, we divided the dataset into two groups: 70% for training and 30% for testing. Next, we applied the training to adjust the model's parameters and prevent overfitting. This ensures that the model has a high generalization power to accurately classify unseen images during testing.

On the other hand, the images used in our experiments were obtained from publicly available datasets on Kaggle called COVID-19 Radiography Database. The dataset used in our study is already labeled into four classes: COVID, normal, lung-opacity, and pneumonia. This large dataset is designed to assist researchers in evaluating their proposed methods in terms of accuracy and time consumption. A detailed description of this dataset, including the number of samples in each class, is presented in Section IV (Dataset). Accordingly, to solve this problem we are designing FHDNN to predict COVID-19 diagnosis. Before using the model, the dataset passes some steps to reduce storage size, increase implementation speed, and improve classification accuracy:

- Converting the dataset into the gray level and ensuring all images are converted from three channels to 1 channel.
- Adjust image intensity using the HE method.
- Resizing all dataset images for speed execution. The operation extracts the best features using LDA for the best classification.
- Performing feature expansion after preprocessing from CXR images is crucial in improving the accuracy of the algorithms. The expanded features' approaches are Average, Maximum, Minimum, Mode, Standard deviation, and Variance ratio.

Lastly, we implement FHDNN to provide fast and highly accurate diagnostics for COVID-19 diseases from CXR images. In the, following sections, we will detail the different parts of the developed methodology.

IV. Dataset

Although there are numerous COVID-19 infections across the world, a large number of CXR images are freely accessible online. Many authors published a Kaggle database [24] to make the database open in front of researchers worldwide. Usually, the CXR images of COVID-19, viral pneumonia, and lung opacity are available in the database. Here, we have four classes of images in the dataset as follows:

- Class 0: 3616 confirmed COVID-19 cases.
- Class 1: 10,192 normal cases.
- Class 2: 6,012 lung- Opacity cases.
- Class 3: 1345 pneumonia disease cases.

Lastly, Fig. 2 shows the viral pneumonia chest CXR. Across the dataset, the seam size (299*299) images are consistent.

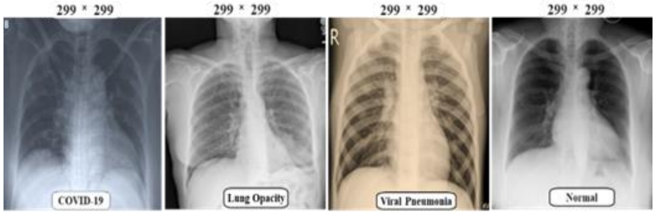

Fig. 2. Sample of CXR images for each class.

V. Dataset Preprocessing

Dealing with enormous amounts of data necessitates the use of the most efficient features for minimizing hardware requirements, such as memory and processing CPU. As mentioned earlier, the experiments' dataset of (21,165) images is classified into four classes. In the beginning, convert the images to one gray level (i.e., one color level). Grayscale is a monochromatic (gray) shade ranging from pure white on the lightest end to pure black on the darkest end. Less information must be provided for each pixel is the reason for differentiating images from any other color ones. Typically, the grayscale intensity is kept as an 8-bit integer, allowing 256 shades of gray ranging from black to white. The soothing impact on the eyes increases the gift of the green color, reduces the contribution of red color, and puts blue between these two [29] (formulated as follows):

$$\text{Grayscale} = \left((0.3 * Red) + (0.59 * Green) + (0.11 * Blue)\right) \quad (1)$$

During this stage, no image of RGB on the dataset has been identified, and the same color levels will all images have. Next, the second stage eliminates noise. CXR images appeared to have a gray level. However, there is an overlap in the image to ensure that it is converted to one group. Fig. 3 shows sample CXR images converted from multi-colors to grayscale.

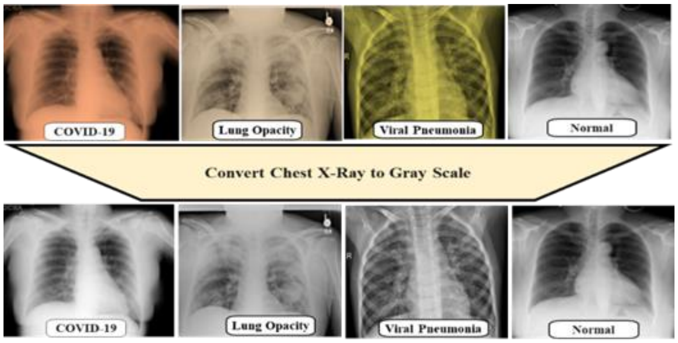

Fig. 3. CXR image samples from different datasets converted to gray level.

CXR images need to distribute lighting over the image by HE adequately. HE is a crucial topic in digital image processing used for contrast adjustment via an image's histogram (i.e., HE is used to enhance contrast). Furthermore, it modifies the image strength to increase opposition [30]. Image improvement aims to improve its quality, highlight desired elements, and make it less intrusive. In an image's histogram, the gray level values are plotted versus the number of pixels at that value. Mathematically, equilibrium can be formulated as follows.

$$\boldsymbol{G(x) = \left(\left(\frac{T(x)}{n}\right) * L\right) - 1} \quad (2)$$

Where the new image after equalization is denoted as G(x), L is the number of gray levels, n is the number of pixels, and T(x) refers to the cumulative sum of each gray level. Fig. 4 shows an application of HE on CXR images.

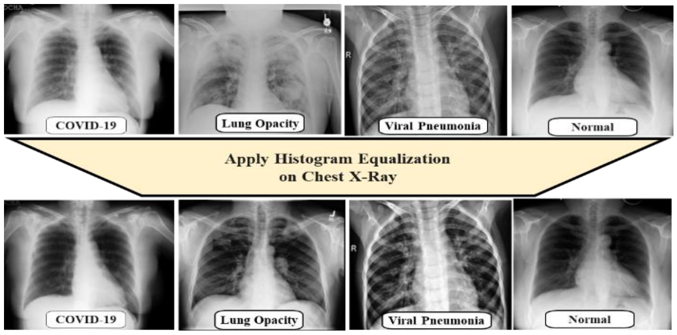

Fig. 4. Applying a Histogram of a CXR image before and after equalization (images are from different datasets).

Image speed and storage size are crucial in image processing, where the image plays an influential role in processing the data in terms of speed. Therefore, we reduced the number of images (of each dataset) to increase processing speed with low storage capacity. At this point, image size must be reduced for faster implementation and less storage space. For giving different resolutions and aspect ratios, resizing images have appeared in many applications. We resized the images from (299 x 299) to (50 x 50). Fig. 5 shows four samples resized from the original 299 x 299 to size 50 x 50. The required formula for resizing is written as follows [31]:

$$\boldsymbol{Ratio = W \,/\, H} \quad (3)$$

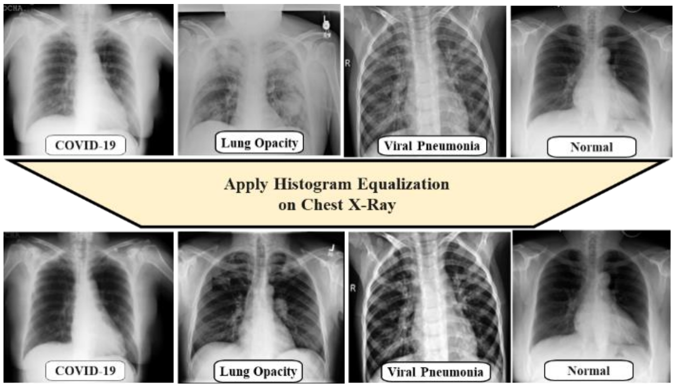

Fig. 5. Sample of resizing four images of the adopted classes.

VI. Features Extraction

The accuracy of the utilized algorithms (ML or DL) can be increased by extracting features (from CXR images) because of an increase in the infection probability. In the literature, since it obtains the best features of image characteristics [32], LDA is the most favored method for extracting discriminative feature pattern classification [33]. LDA starts after completing the earlier stages of image processing. However, when dealing with high-dimensional and small sample-size data, classical LDA suffers from the singularity problem. Under supervision, the LDA reduces dimensionality inside a single classification and separates the classifications so we can select the best image type [34]. The main pros of LDA are that it separates two or more classes and models the group differences in groups by projecting the spaces in a higher dimension into space with a lower extent. Both features create a new axis that minimizes the variance and maximizes the two variables' class distance. Calculate each class's average from the CXR image within the dataset by LDA Algorithm. The mathematical equations used in LDA are similar in action are identical in action [33]. Image preprocessing techniques are applied before model training to achieve better detection accuracy and reduce training duration. Specifically, preprocessing will ensure that the image is transformed from two dimensions to one dimension, consequently speeding up the implementation. Additionally, a propionate fast deep learning proposal design must be utilized for further detection accuracy and higher speed. After implementing LDA, the stages are gray level, histogram equalization, stages are resized, and the best features are extracted.

The development of DL algorithms products is a significant contribution to dataset reprocessing. In the beginning, the images are converted to a single channel with suitable resizing and highlighting desired elements extracting the features. Next, we perform classification between classes and reduce dimensionality within a single category. Accordingly, database preprocessing is essential for enhancing data quality to speed up deep learning with optimal performance.

VII. Feature Expansion

This section explains a feature expansion approach for the one-dimension. As mentioned earlier, LDA is implemented after completing preprocessing on a dataset of CXR images. The output of the LDA got three components from CXR images [35].

The proposed feature expansion is capable of increasing the number of features from three to eighteen. This procedure applies a window that slides through the pattern to expand the high and low elements. The window is a (one × ten) shaped matrix and will be transformed with the input pattern matrix. For data classification, feature expansion can improve classification accuracy by enabling deep CNN application on any dataset from CXR images. Furthermore, feature expansion is helpful if there is a variance in the CXR or a change in the x-ray equipment, such as reducing noise in an image. According to the literature, experiment results demonstrated that the feature expansion method improves the classification performance of CNN algorithms for detecting COVID-19. Finally, the expanded features' approaches (as explained below) are average, maximum, minimum, mode, standard deviation, and variance ratio.

A. Average

The average (or mean) of a set of numbers is calculated by dividing their summations by how many they are. Mathematically, the mean format $\bar{x}$ of a data set is the sum of all data values divided by the count or size n. The following equation can be used to determine the average or mean of a window within the dataset [35].

$$mean = x^{-} = \frac{\sum_{i=1}^{n} x^{i}}{n} \tag{4}$$

B. Maximum

Finding the maximum value within a window and checking with all data is important. A real-valued function f defined on a domain (X) has a global (or absolute) maximum point at

$$x_i . if\ f(x_i) \geq f(X) for\ all\ x\ in\ X \tag{5}$$

The value of the function at the top ended the total weight of the process [36].

C. Minimum

The minimum value of a function is determined by finding the lowest value within a window. The value of the process at a minimum point is called the minimum value of the position [36]. Symbolically, this can be written as follows:-

$$x_i . if\ f(x_i) \leq f(X) for\ all\ x\ in\ X \tag{6}$$

D. Mode

Mode is the number (or value) that appears most frequently in a data set. For data without any repeating values, there is no mode at all. Furthermore, the mode value depends on the given dataset. It is formulated as follows [37].

$$Mode = L + h\ \frac{(f_m - f_1)}{(f_m - f_1) + (f_m - f_2)} \tag{7}$$

Where L the lower limit of the modal class is, h is the size of the class interval, and f_m denotes the frequency of the modal class. f_1 (f_2) .This is the frequency of the class preceding (succeeding) the modal classes.

E. Standard Deviation

Standard deviation is a formula used to determine the values of scattered data. It is the deviation of the values or data from an average to a mean. A smaller (larger) standard deviation indicates that the results are nearer to (far from) their average. It is important to mention that no negative value can standard deviation be. It is formulated as follows [38].

$$the\ Standard\ deviation = \sqrt{\frac{\sum(\boldsymbol{X}-\boldsymbol{\mathcal{X}}^{-})^{2}}{\boldsymbol{n}-\boldsymbol{1}}} \quad (8)$$

Where X is each value in the sample t, $\mathcal{X}^{-}$ refers to sample means, and n is the number of values in the sample.

F. Variance Ratio

Statistically, variance (symbolized by σ^2) refers to the spread between numbers in a data set. In other words, variance calculates how far every number is from the average (i.e., from other numbers in the group). Both traders and analysts use it to find volatility through the dataset. Variance is calculated as follows [39]:

$$\boldsymbol{\sigma^{2}} = \frac{\sum_{i=1}^{n}(\,x_{i}-x^{-}\,)^{2}}{N} \quad (9)$$

Where $\mathcal{X}^{-}$ refers to the mean of all values in the dataset, $\boldsymbol{x_i}$ is each value in the dataset, and $\boldsymbol{n}$ is the number of values in the dataset.

VIII. Fast Hybrid Deep Neural Network

A. Preliminaries

In the state-of-the-art, researchers and authors who use transfer learning for COVID-19 detection need to consider the trade-offs between accuracy and computational resources, as well as the time required to fine-tune the model on large datasets. It is important to carefully balance these factors to ensure that transfer learning can be applied effectively in the state of the art for COVID-19 detection. These trade-offs are not easy to ensure. Although the ensemble model can enhance the efficiency of each model separately, its implementation requires fine-tuning the aforementioned deep models which is a very challenging task since it requires powerful machines.

Alternative methods have been proposed to extract deep features and feed them into an SVM classifier without utilizing transfer learning, as demonstrated in [40]. However, this approach is less time-consuming, but it lacks a high generalization power because the extracted features are tailored to the ImageNet dataset initially used to train the model. Consequently, the SVM classifier may not achieve a very high accuracy level.

To solve this problem, the proposed HFDNN aims to extract features using LDA and apply some feature engineering to feed a DNN for detection. This strategy will avoid fine-tuning existing models to large datasets, and considerably speed up the training process. Before the dataset was enlarged, it underwent preprocessing, and the best features were extracted using LDA. Next, we altered it from two dimensions to one dimension to shorten training time and hardware needs. By using this method, existing models won't need to be adjusted for large datasets, and the training process will be greatly accelerated. In this section, we explain our proposal, the fast hybrid deep Convolutional Neural Network model to detect COVID-19 from CXR images.

B. Modeling

A rapid diagnosis of COVID-19 is necessary to treat and control the disease. CXR is an essential tool that can be used to find COVID-19. Therefore, this study aims to improve classification accuracy and reduce training time and hardware requirements. Furthermore, it should accurately identify COVID-19 cases so doctors can treat patients appropriately. Therefore, we proposed to utilize a novel deep-learning method to predict COVID-19 using CXR images based on deep CNN. The proposed model implements the preprocessing on the dataset from CXR images in some steps before applying the proposed deep CNN for the COVID-19 classification. As explained later, our model's (FHDNN) objectives are the automatic identification of four classes: 1) normal, 2) lung opacity, 3) viral pneumonia, and 4) COVID-19. As it is named, FHDNN is a quick deep CNN used to classify CXR images for detected COVID-19. The proposed FHDNN deals with features (eighteen features) that must be one-dimensional. FHDNN takes an input package of features, processes it, and classifies it under specific categories. It consists of several layers intertwined with each other and arranged in a way that makes it more highly. For each input data set, eighteen features pass through a series of convolution layers with filters, Pooling, and Fully Connected layers (FC) to classify objects with probabilistic values between (zero and one). The architecture of FHDNN for COVID-19 detection is depicted in Fig. 6. FHDNN comprises twenty-seven layers categorized as follows:

- Eight Conv1D layers: Eight convolutional layers are used for feature extraction of type 1D.
- Seven MaxPooling 1D layers.
- Three Dense layers: They are divided into 1) One layer representing a fully connected layer in a convolutional neural network; and 2) two dense layers are placed before the Flatten to obtain robust features closer to them.
- Eight LeakyReLU layers: The Activation function is LeakyReLU.
- One Flatten layer: The full connection step in CNNs.

C. Layers of FHDNN

As mentioned earlier, the FHDNN is expected to achieve accurate classification results due to dealing with one-dimension CXR images (after preprocessing datasets). Table I lists and explains the FHDNN layers in terms of Filters, Strides, and Padding. We listed detailed explanations and clarification of each layer with the activation and setting of all parameters in each layer. The difference is the structure of the input data and how the filter (also called a feature detector or a convolution kernel) moves across the data. It is worth mentioning that the attained FHDNN model layers and their parameters in this network are one-dimensional (will be explained and shown according to the order of layers).

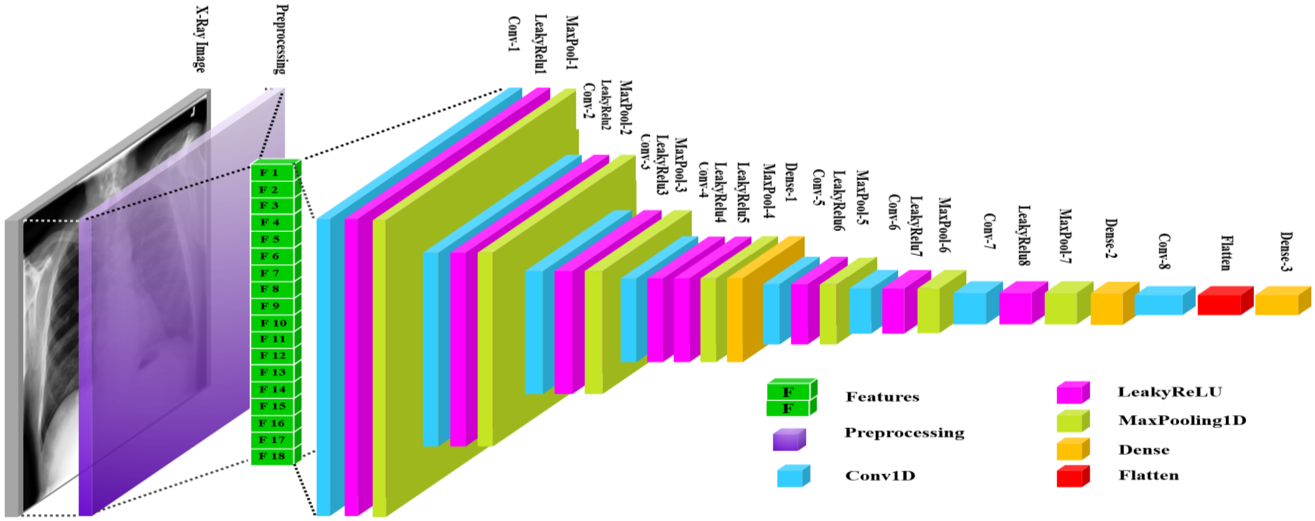

Fig. 6. FHDNN architecture.

TABLE I. INFORMATION OF FHDNN LAYERS

No	Layer	Filter	Strides	Padding
1.	Max pooling 1D-1	16	1	-
2.	LeakyReLU-1	-	-	-
3.	Convolution 1D -1	16	1	Valid
4.	Max pooling 1D-2	32	1	-
5.	LeakyReLU-2	-	-	-
6.	Convolution 1D-2	32	1	Valid
7.	Max pooling 1D-3	32	1	-
8.	LeakyReLU-3	-	-	-
9.	Convolution 1D-3	32	1	Valid
10.	Max pooling 1D-4	128	1	
11.	LeakyReLU-4	-	-	-
12.	Convolution 1D-4	128	1	Valid
13.	Dense 1D-1	128	-	-
14.	Max pooling 1D-5	256	1	-
15.	LeakyReLU-5	-	-	-
16.	Convolution 1D-5	256	1	Valid
17.	Max pooling 1D-6	512	1	-
18.	LeakyReLU-6	-	-	-
19.	Convolution 1D-6	512	1	Valid
20.	Dense 1D-2	1024	-	-
21.	Max pooling 1D-7	1024	1	-
22.	LeakyReLU-7	-	-	-
23.	Convolution 1D-7	1024	1	Valid
24.	LeakyReLU-8	-	-	-
25.	Convolution 1D-8	50	1	-
26.	Dense 1D-3	4	-	-
27.	Flatten 1D-1	50	-	-

The utilized layers are arranged as follows:

- Eight convolutional layers in the proposed lightweight CNN model have sequential filters (16, 32, 32, 128, 256, 512, 1024, and 50). This layer applies kernels that slide through the pattern to extract low- and high-level features [41]. The kernel is a (one × three) shaped matrix to be transformed with the input pattern matrix. The stride is (one × one) that parameter is the number of steps tuned for shifting over the input matrix. During the design process, the padding was omitted from the border elements. The padding is designated as 'VALID' in the remaining convolutional layers and does not consider the boundary components.

- Eight LeakyReLU layers are set to (alpha=0.3). The activation function LeakyReLU is used as an activation function after every convolutional layer. The LeakyReLU layers were used twice after the convolutional layer with filter size 128. Next, the third layer after the LeakyReLU layer is the pooling layer.

- The pooling layer is usually applied to the created feature maps. It aims to reduce the number of feature maps and network parameters by applying corresponding mathematical computation. A (one × one) max pool is used seven times after each convolutional block.

- As we propose a lightweight FHDNN model, three dense layers have been used. The dense layer performs a matrix-vector multiplication, and the values used in the matrix are parameters that can be trained and updated with the help of backpropagation. The output generated by the dense layer is an 'n-dimensional vector. The activation function (linear) uses the two dense layers by setting the vector sequential as follows (128, 1024). The activation function (softmax, 4) is the last dense layer to classify four classes.

- Lastly, to flatten the matrix into a vector, a flattening layer was used once. Additionally, the most and last crucial layer of FHDNN is fully connected. This layer functions like a multilayer perceptron. Rectified linear unit (LeakyReLU) activation function is commonly used on a fully connected layer. In contrast, the softmax activation function is used to predict output images in the last layer of a fully connected layer. This step will look at the softmax regression used for multi-class classification problems.

Now, we can explain the experimental parameters obtained from each layer at the proposed system procedures implementation on the CXR Images for diagnosis of COVID-19. As listed in Table II, the FHDNN model includes a mix of convolution neural network layers. Afterward, the parameters of every layer in the FHDNN are presented to show the success of the submitted diagnosis of COVID-19 by using CXR images. The total number of parameters is 3,303,494, and the batch size equals 64. Finally, Table II also summarizes the proper trainable parameters for each layer of the FHDNN model.

TABLE II. SUMMARY OF THE FHDNN MODEL

Layer (type)	**Output Shape**	**Number of parameters**
Conv1d (Conv1D)	(None, 16, 16)	64
Leaky_re_lu (LeakyReLU)	(None, 16, 16)	0
Max_pooling1d (MaxPooling1D)	(None, 16, 16)	0
Conv1d_1 (Conv1D)	(None, 14, 32)	1568
Leaky_re_lu_1 (LeakyReLU)	(None, 14, 32)	0
Max_pooling1d_1 (MaxPooling1D)	(None, 14, 32)	0
Conv1d_2 (Conv1D)	(None, 12, 32)	3104
Leaky_re_lu_2 (LeakyReLU)	(None, 12, 32)	0
Max_pooling1d_2 (MaxPooling1D)	(None, 12, 32)	0
Conv1d_3 (Conv1D)	(None, 10, 128)	12416
Leaky_re_lu_3 (LeakyReLU)	(None, 10, 128)	0
Leaky_re_lu_3 (LeakyReLU)	(None, 10, 128)	0
Max_pooling1d_3 (MaxPooling1D)	(None, 10, 128)	0
Dense (Dense)	(None, 10, 128)	16512
Conv1d_4 (Conv1D)	(None, 8, 256)	98560
Leaky_re_lu_5 (LeakyReLU)	(None, 8, 256)	0
Max_pooling1d_4 (MaxPooling1D)	(None, 8, 256)	0
Conv1d_5 (Conv1D)	(None, 6, 512)	393728
Leaky_re_lu_6 (LeakyReLU)	(None, 6, 512)	0
Max_pooling1d_5 (MaxPooling1D)	(None, 6, 512)	0
Conv1d_6 (Conv1D)	(None, 4, 1024)	1573888
Leaky_re_lu_7 (LeakyReLU)	(None, 4, 1024)	0
Max_pooling1d_6 (MaxPooling1D)	(None, 4, 1024)	0
Dense_1 (Dense)	(None, 4, 1024)	1049600
Conv1d_7 (Conv1D)	(None, 2, 50)	153650
Flatten (Flatten)	(None, 100)	0
Dense_2 (Dense)	(None, 4)	404

IX. PERFORMANCE EVALUATION

A. Evaluation Metrics

After the model is trained, we evaluate its performance using the images of the testing dataset. In this work, we adopted the following evaluation matrices: Accuracy, Precision, Recall, and F1-Score. Before delving into the results, we need to explain these metrics. After the model's movement, the FHDNN's performance on the testing dataset was evaluated. It is important to mention that this study did not use cross-validation. Therefore, the holdout validation technique has been used instead because the dataset has enough samples for testing and training.

A confusion matrix table is usually used to explain the classifier's performance (i.e., model classification) on a data set in which its actual cases are known. The confusion matrix comprises four parameters (TP, FP, TN, and FN). While TP and TN refer to the patient being classified correctly, FP and FN mean it misdiagnosed the case (i.e., classified incorrectly). This section briefly defines the evaluation metrics in AD classification. Following, we explain the evaluation metrics.

1) Accuracy: It is widely used to measure the model's accuracy validation (training) or classification. As mentioned above, via a confusion matrix, the number of TP, TN, FP, and FN are calculated, which further helps check the proposed model's efficacy. Accordingly, as formulated in Eq. (1), the accuracy is the number of correct classifications (i.e., TP and TN) divided by the total number of classifications.

$$\boldsymbol{Accuracy} \; = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \tag{10}$$

2) Precision: It calculates how many cases classified as positive (infection) by the model was supposed to be indicated as positive. It is defined as the ratio of classified TPs to the total number of classified positives (correctly (TP) and incorrectly (FP). Thus, it is formulated as follows.

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \tag{11}$$

3) Recall: The third measure of the confusion matrix is Recall (or sensitivity). The recall is the number of positive cases foretold adequately by the model is estimated by the recall, sensitivity, or "actual positive rate. Recall calculates how well the model has classified the positive examples. It is defined as the ratio of classified TPs to the real positives and is written as follows.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{12}$$

4) F1-score: It is the harmonic mean of precision and recall. Perfect precision and Recall can achieve the highest F-score (i.e., close to 1).

$$\boldsymbol{F1 - score} \; = \boldsymbol{2} * \frac{(\boldsymbol{Recall} * \boldsymbol{Precision})}{\boldsymbol{Recall} + \boldsymbol{Precision}} \tag{13}$$

Preparing the dataset preprocessing is one of the prime aspects of training and testing deep learning for the desired results. The CXR images have passed through multi-steps in preprocessing before entering the FHDNN layers, providing an accurate effect in the classification procedure.

B. Model Performance

The training time required to train the FHDNN algorithm is important to look at the accuracy obtained from the classification algorithm. For instance, 100 epochs are recommended in FHDNN training. When using FHDNN, the training time is 14 sec, which differs from the attained feature extraction algorithms. The obtained error rate (α) from training the classifier is 0.0001. Table III lists the results of applying FHDNN with the four classes (COVID-19 class, normal, lung-opacity, and viral Pneumonia, based on the performance confusion matrix. As listed in Table III, the performance metrics of the proposed modes achieved the following results. Firstly, the accuracy of the model reached 99.9%. Secondly, the model's foretelling of positive samples is assessed by precision. Meanwhile, the models' level of precision showed that FHDNN has the most fantastic weighted average value of 99.9% for the four classes. Thirdly, the actual positive rate (sensitivity) measures the number of positive cases correctly foretold by the model. The Sensitivity of the model reached 99.9%. Lastly, the F1-score achieved an almost perfect value, 99.9%.

TABLE III. PERFORMANCE OF THE PROPOSED MODEL

Type	Accuracy	Precision	Recall	F-score
COVID-19	99.9%	99.9%	99.9%	99.9%
Normal	99.9%	99.9%	99.9%	99.9%
Lung- Opacity	99.9%	99.9%	99.9%	99.9%
Viral Pneumonia	99.9%	99.9%	99.9%	99.9%

C. FHDNN vs. Counterparts

Using CXR images, many DL models-based COVID-19 classification systems have been utilized widely. For example, AlexNet, ResNet, VGG16, VGG19, and GoogleNet. To demonstrate the superiority of our work, we compare it with the achievements of some CNN-based models in the literature. The comparisons are based on: classification classes, accuracy, precision, recall, F-score, and training time. Additionally, we compared our proposal with 12 studies that utilized CNN models.

Before comparing the models with ours, it is important to mention that the convolutional layers for MobileNetV2, DenseNet-121, ResNet50, VGG19, and ChestX-Ra6 are 53, 121, 50, 16, and 6, respectively. Their training times are 6457.20, 7571.84, 7018.10, 8215.94, and 6150.62 sec, respectively (see Fig. 7). It is important to mention that different computer configurations could provide different results. Although some studies calculated the training time, they consumed a very long training time. Specifically, the models completed the training in thousands of seconds (i.e., hours). On the other hand, our model needed only 13 sec. Some examples can be found in references [27] [42-52]. Except for the study [50], most models performed on 3 and 2-class classifications. However, our model is better than [50] in all evaluation metrics (accuracy, precision, recall, and F-score). Furthermore, our training time is significantly less than it. Similar to our work, the models [42] classified four classes; however, our model outperformed in performance evaluation metrics and training time. Accordingly, our model outperformed the counterparts.

When it comes to detecting COVID-19 using transfer learning techniques, one of the significant challenges is the amount of time it takes to apply these methods to large datasets. This process is particularly time-consuming when dealing with large amounts of data of CXR images, as the model needs to be fine-tuned to adjust its parameters. This requires retraining the model on the new data, which can take a significant amount of time, even with powerful computing resources. As a result, researchers and practitioners who use transfer learning for COVID-19 detection need to consider the trade-offs between accuracy and computational resources, as well as the time required to fine-tune the model on large datasets. It is important to carefully balance these factors to ensure that transfer learning can be applied effectively in the state of the art for COVID-19 detection. These trade-offs are not easy to ensure. To solve this problem, the proposed HFDNN aims to automate COVID-19 disease classification to maintain high accuracy and reduce execution time. The dataset was first preprocessed before being extended and the best features were retrieved using LDA. Next, in order to decrease training time and hardware requirements, we changed it from two dimensions to one dimension. This strategy will avoid fine-tuning existing models to large datasets, and considerably speed up the training process as shown in Fig. 7.

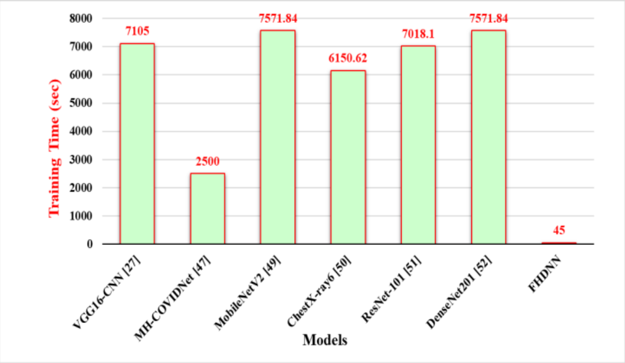

Fig. 7. Training time of each model (sec).

D. Performance matrix and comparison with related works

In this subsection, we explain the performance matrix and comparison with related works. Though the model is complex in architecture, it has twenty-seven convolutional layers. Still, the model classification performance is better than the other models. One of the main concerns of this study is designing a model that provides higher classification accuracy and reduces the training time with the lowest requirement hardware for large or small amounts of data. For that reason, a lightweight FHDNN model has been designed to measure whether it performed well for these criteria in the case of a multiclass environment.

1) Accuracy: Fig. 8 shows the tested obtained accuracy using some of the models of DL and our work. It is clear from the figure that our model achieved the highest classification accuracy, which has an accuracy of 99.9%.

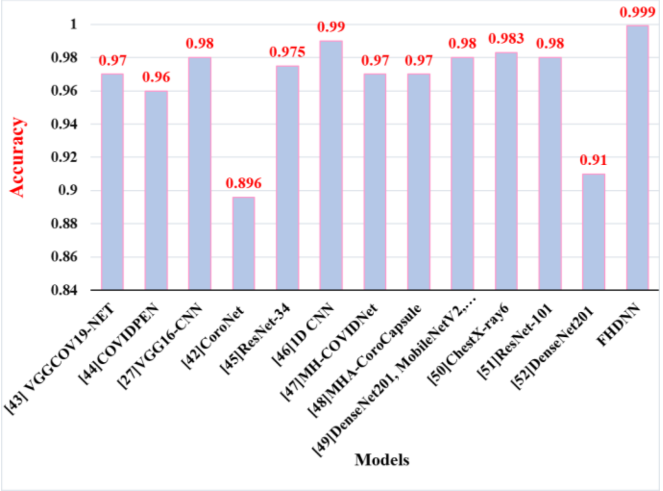

Fig. 8. The accuracy of the models.

2) Precision: The second is the precision that assesses the model precision in foretelling positive samples. Meanwhile, the six models' level of precision showed that FHDNN has the most excellent weighted average value (99.9%) for four classes than other models. Fig. 9 shows that the precision of FHDNN has the most excellent weighted average value (0.999) for four classes than other models of CNN.

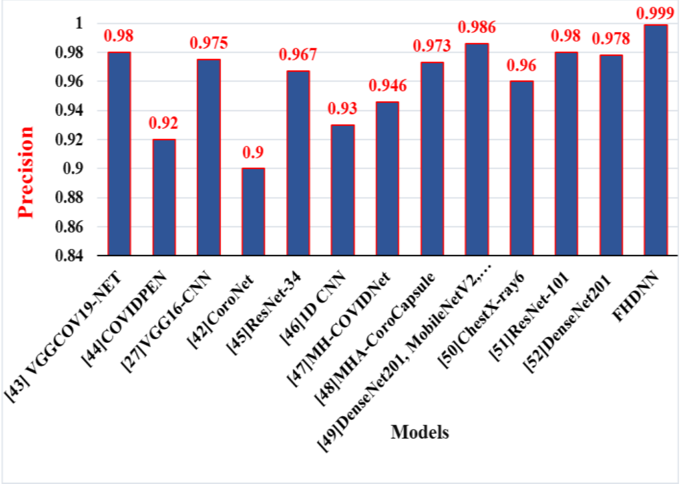

Fig. 9. The precision of the models.

3) Recall: As shown in Fig. 10, the results show FHDNN method outperforms the close models of CNN in a variety of weighted average sensitivities. The FHDNN determined weighted average has (0.999) of sensitivity, but the VGG19 has (0, 98), GoogleNet has (0, 92), DenseNet201 has (0.92), etc.

4) F1-score: As mentioned earlier, the F1-score is the harmonic mean of precision and recall. As shown in Fig. 11, the F1- score value of the different models also shows that the FHDNN has a value (of 99.9%), the highest compared to the related work comparison models. The results state that DL methods outperform the comparative models in classification F1-score (percentage).

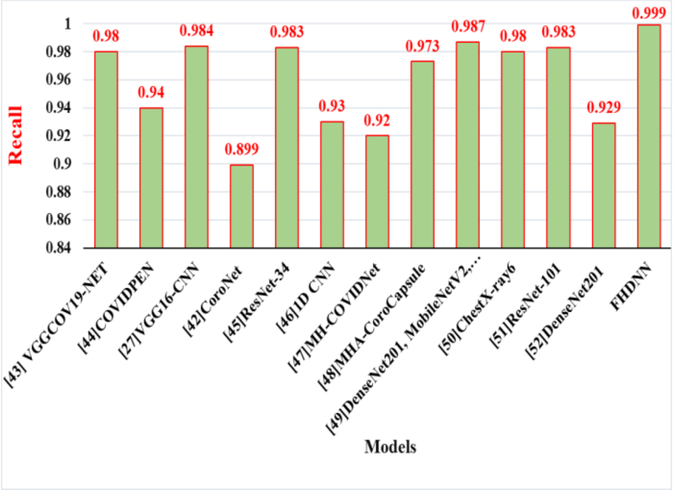

Fig. 10. Recall (sensitivity) of the models.

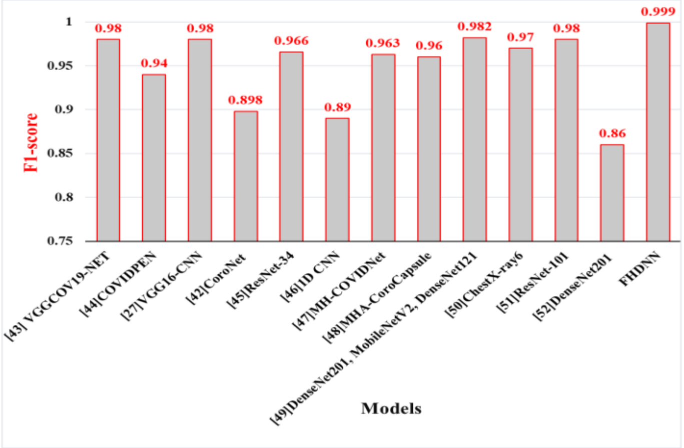

Fig. 11. F1-score of the models.

This article presented a DL method for fast detection and classification of COVID-19 disease. The preprocessing stage has been performed on the dataset to achieve better speed and accuracy. The comparison demonstrated that we achieved better results than some relevant previous works, knowing that it dealt with an extensive and unbalanced database. Furthermore, different DL models were applied to the same dataset for comparison, but they got lower results in all measures than in our proposal, FHDNN. The result states that FHDNN possesses the highest important parameters that measure each model's performance compared with other CNN models. Meanwhile, our model's accuracy, precision, and sensitivity show that FHDNN has 99.9% of the most outstanding value. The F1-score value of the DL models also states that the FHDNN has 99.9% and stays the most excellent weighted average compared to the CNN comparison models with less than FHDNN a value.

X. Conclusion

COVID-19 is one of the cases in which doctors and medical physicians face difficulties in correctly identifying these diseases. CXR imaging has several advantages over other imaging and detection techniques. If trained correctly, a machine can classify CXR images more accurately than a human. Numerous results have reported COVID-19 detection from a smaller or large set of CXR images. However, the literature did not say the effect of image enhancement and lung x-ray preprocessing of a large or small dataset of CXR images.

Therefore, this article proposed a novel lightweight new deep CNN (FHDNN) model to detect COVID-19 diseases using CXR images. To calculate our model's performance for detecting COVID-19 diseases, we trained our model using 21,165 CXR images after completing all preprocessing stages. These are classes unbalanced using various preprocessing techniques such as gray image level, resizing, and then enhancement by histogram equalization. Furthermore, we proposed extracting the best features by LDA, then the feature expansion approach and converting from two-Dimension to one-Dimension reduce time training. The proposed FHDNN model for multiple classifications (normal, pneumonia, lung-opacity, and COVID-19) achieved 99.9% in all performance metrics (accuracy, precision, recall, and F1-score). These results demonstrated the superiority of our model over the proposed CNN models (previous works). Our FHDNN model has achieved excellent performance in diagnosing COVID-19 by using CXR Images, which can help medical physicians diagnose these diseases correctly. In the future, we intend to apply the model and improve it for diagnosing other diseases by CXR images.

REFERENCES

[1] T. Islam, S. Absar, S. A. I. Nasif, and S. S. Mridul, "Deep Neural Network models for diagnosis of COVID-19 Respiratory diseases by analyzing CT-Scans and Explain-ability using trained models," in 2022 International Conference on Inventive Computation Technologies (ICICT), 2022: IEEE, pp. 16-23.

[2] C. A. Larabell and K. A. J. C. o. i. s. b. Nugent, "Imaging cellular architecture with X-rays," vol. 20, no. 5, pp. 623-631, 2010.

[3] W. Hariri and A. J. S. c. Narin, "Deep neural networks for COVID-19 detection and diagnosis using images and acoustic-based techniques: a recent review," vol. 25, no. 24, pp. 15345-15362, 2021.

[4] H. S. Alghamdi, G. Amoudi, S. Elhag, K. Saeedi, and J. J. I. A. Nasser, "Deep learning approaches for detecting COVID-19 from chest X-ray images: A survey," vol. 9, pp. 20235-20254, 2021.

[5] M. Agarwal et al., "A novel block imaging technique using nine artificial intelligence models for COVID-19 disease classification, characterization, and severity measurement in lung computed tomography scans on an Italian cohort," vol. 45, no. 3, pp. 1-30, 2021.

[6] A. M. Ayalew, A. O. Salau, B. T. Abeje, B. J. B. S. P. Enyew, and Control, "Detection and classification of COVID-19 disease from X-ray images using convolutional neural networks and histogram of oriented gradients," vol. 74, p. 103530, 2022.

[7] R. Sadik et al., "Covid-19 pandemic: a comparative prediction using machine learning," vol. 1, no. 1, pp. 1-16, 2020.

[8] M. S. Anggreainy, F. H. Kansil, and A. M. Illyasu, "Comparative Performance Analysis of Machine Learning Classifier for COVID-19 Detection using Chest X-Ray Images," in 2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE), 2022: IEEE, pp. 337-341.

[9] H. A. Ali, W. Hariri, N. S. Zghal, and D. B. Aissa, "A Comparison of Machine Learning Methods for best Accuracy COVID-19 Diagnosis Using Chest X-Ray Images," in 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2022: IEEE, pp. 349-355.

[10] S. Akbar, H. Tariq, M. Fahad, G. Ahmed, and H. J. J. E. Syed, "Contemporary Study on Deep Neural Networks to Diagnose COVID-19 Using Digital Posteroanterior X-ray Images," vol. 11, no. 19, p. 3113, 2022.

[11] M. Ahsan, M. A. Based, J. Haider, and M. J. S. Kowalski, "COVID-19 detection from chest X-ray images using feature fusion and deep learning," vol. 21, no. 4, p. 1480, 2021.

[12] K. Suzuki, "Overview of deep learning in medical imaging," Radiological physics and technology, vol. 10, no. 3, pp. 257-273, 2017.

[13] X.-W. Chen and X. Lin, "Big data deep learning: challenges and perspectives," IEEE access, vol. 2, pp. 514-525, 2014.

[14] M. F. Aslan, K. Sabanci, A. Durdu, and M. F. Unlersen, "COVID-19 diagnosis using state-of-the-art CNN architecture features and Bayesian Optimization," Computers in Biology and Medicine, p. 105244, 2022.

[15] T. B. Alakus and I. Turkoglu, "Comparison of deep learning approaches to predict COVID-19 infection," Chaos, Solitons & Fractals, vol. 140, p. 110120, 2020.

[16] G. S. George, P. R. Mishra, P. Sinha, and M. R. Prusty, "COVID-19 detection on chest X-ray images using Homomorphic Transformation and VGG inspired deep convolutional neural network," Biocybernetics and Biomedical Engineering, 2022.

[17] A. K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, and J. J. Rodrigues, "Identifying pneumonia in chest X-rays: A deep learning approach," Measurement, vol. 145, pp. 511-518, 2019.

[18] B. Antin, J. Kravitz, and E. Martayan, "Detecting pneumonia in chest X-Rays with supervised learning," Semanticscholar. org, 2017.

[19] N. N. 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, 2020.

[20] E. Ayan and H. M. Ünver, "Diagnosis of pneumonia from chest X-ray images using deep learning," in 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), 2019: Ieee, pp. 1-5.

[21] G. Gaál, B. Maga, and A. Lukács, "Attention u-net based adversarial architectures for chest x-ray lung segmentation," arXiv preprint arXiv:2003.10304, 2020.

[22] A. Sharma, K. Singh, D. J. B. S. P. Koundal, and Control, "A novel fusion based convolutional neural network approach for classification of COVID-19 from chest X-ray images," vol. 77, p. 103778, 2022.

[23] M. E. Chowdhury et al., "Can AI help in screening viral and COVID-19 pneumonia?," vol. 8, pp. 132665-132676, 2020.

[24] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database.

[25] A. Narin, C. Kaya, Z. J. P. A. Pamuk, and Applications, "Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks," vol. 24, no. 3, pp. 1207-1220, 2021.

[26] W. A. Hamwi and M. M. J. I. i. M. U. Almustafa, "Development and integration of VGG and dense transfer-learning systems supported with diverse lung images for discovery of the Coronavirus identity," p. 101004, 2022.

[27] A. A. Ramadhan and M. J. A. S. Baykara, "A Novel Approach to Detect COVID-19: Enhanced Deep Learning Models with Convolutional Neural Networks," vol. 12, no. 18, p. 9325, 2022.

[28] S. Bharati, P. Podder, M. Mondal, and V. Prasath, "Medical imaging with deep learning for COVID-19 diagnosis: a comprehensive review," arXiv preprint arXiv:2107.09602, 2021.

[29] K. Padmavathi, K. J. I. J. o. S. Thangadurai, and Technology, "Implementation of RGB and grayscale images in plant leaves disease detection–comparative study," vol. 9, no. 6, pp. 1-6, 2016.

[30] C. Nithyananda and A. Ramachandra, "Review on histogram equalization based image enhancement techniques," in 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016: IEEE, pp. 2512-2517.

[31] W. Siddique, I. V. Shevchuk, L. El-Gabry, N. B. Hushmandi, T. H. J. H. Fransson, and M. Transfer, "On flow structure, heat transfer and pressure drop in varying aspect ratio two-pass rectangular channel with ribs at 45," vol. 49, no. 5, pp. 679-694, 2013.

[32] Z. Fan, Y. Xu, and D. Zhang, "Local linear discriminant analysis framework using sample neighbors," IEEE Transactions on Neural Networks, vol. 22, no. 7, pp. 1119-1132, 2011.

[33] M. Anggo and L. Arapu, “Face recognition using fisherface method,” in Journal of Physics: Conference Series, 2018, vol. 1028, no. 1: IOP Publishing, p. 012119.

[34] T. V. Bandos, L. Bruzzone, G. J. I. T. o. G. Camps-Valls, and R. Sensing, “Classification of hyperspectral images with regularized linear discriminant analysis,” vol. 47, no. 3, pp. 862-873, 2009.

[35] L. J. T. A. S. Zhang, “Sample mean and sample variance: Their covariance and their (in) dependence,” vol. 61, no. 2, pp. 159-160, 2007.

[36] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” in Traces and emergence of nonlinear programming: Springer, 2014, pp. 247-258.

[37] N. M. Kopelman, J. Mayzel, M. Jakobsson, N. A. Rosenberg, and I. J. M. e. r. Mayrose, “Clumpak: a program for identifying clustering modes and packaging population structure inferences across K,” vol. 15, no. 5, pp. 1179-1191, 2015.

[38] C. Leys, C. Ley, O. Klein, P. Bernard, and L. J. J. o. e. s. p. Licata, “Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median,” vol. 49, no. 4, pp. 764-766, 2013.

[39] T. C. Urdan, Statistics in plain English. Routledge, 2011.

[40] Khamparia, Aditya, et al. "An internet of health things - driven deep learning framework for detection and classification of skin cancer using transfer learning." Transactions on Emerging Telecommunications Technologies 32.7 (2021): e3963.

[41] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, pp. 436-444, 2015.

[42] A. I. Khan, J. L. Shah, and M. M. Bhat, “CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images,” Computer methods and programs in biomedicine, vol. 196, p. 105581, 2020.

[43] A. J. N. C. Karacı and Applications, “VGGCOV19-NET: automatic detection of COVID-19 cases from X-ray images using modified VGG19 CNN architecture and YOLO algorithm,” vol. 34, no. 10, pp. 8253-8274, 2022.

[44] A. K. Jaiswal, P. Tiwari, V. K. Rathi, J. Qian, H. M. Pandey, and V. H. C. J. M. Albuquerque, “Covidpen: A novel covid-19 detection model using chest x-rays and ct scans,” 2020.

[45] S. R. Nayak, D. R. Nayak, U. Sinha, V. Arora, R. B. J. B. S. P. Pachori, and Control, “Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study,” vol. 64, p. 102365, 2021.

[46] E. A. Abbood and T. A. Al-Assadi, “GLCMs Based multi-inputs 1D CNN Deep Learning Neural Network for COVID-19 Texture Feature Extraction and Classification,” Karbala International Journal of Modern Science, vol. 8, no. 1, pp. 28-39, 2022.

[47] M. J. B. S. P. Canayaz and Control, “MH-COVIDNet: Diagnosis of COVID-19 using deep neural networks and meta-heuristic-based feature selection on X-ray images,” vol. 64, p. 102257, 2021.

[48] F. Li, X. Lu, and J. J. I. T. o. M. I. Yuan, “MHA-CoroCapsule: Multi-Head Attention Routing-Based Capsule Network for COVID-19 Chest X-Ray Image Classification,” vol. 41, no. 5, pp. 1208-1218, 2021.

[49] V. C. Shinde and P. S. Kulkarni, “Automatic COVID-19 Detection from Chest X-Rays using Deep Learning Techniques,” in 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022: IEEE, pp. 587-594.

[50] M. Nahiduzzaman, M. R. Islam, and R. J. E. S. w. A. Hassan, “ChestX-Ray6: Prediction of multiple diseases including COVID-19 from chest X-ray images using convolutional neural network,” vol. 211, p. 118576, 2023.

[51] A. Gupta, 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, 2021.

[52] G. Dhiman, V. Chang, K. Kant Singh, and A. Shankar, “Adopt: automatic deep learning and optimization-based approach for detection of novel coronavirus covid-19 disease using x-ray images,” Journal of biomolecular structure and dynamics, vol. 40, no. 13, pp. 5836-5847, 2022.