

Designing Hybrid CNN-SVM Model for COVID-19 Classification Based on X-ray Images Using LGBM Feature Selection

Sri Hartini^a, Zuherman Rustam^{a,*}, Rahmat Hidayat^b

^a Department of Mathematics, Universitas Indonesia, Depok 16424, Indonesia

^b Department of Information Technology, Politeknik Negeri Padang, 25164, Indonesia

Corresponding author: *rustam@ui.ac.id

Abstract— COVID-19 still exists at an alarming level; hence, early diagnosis is important for treating and controlling this disease due to its rapid spread. The use of X-rays in medical image analysis can play an essential role in fast and affordable diagnosis. This study used a two-level feature selection in hybrid deep convolutional features obtained from the extraction of X-ray images. The transfer learning-based approach was implemented using five convolutional neural networks (CNNs) named VGG16, VGG19, ResNet50, InceptionV3, and Xception. The combination of two or three CNNs' performance as a feature extractor was then carefully analyzed. We selected the features obtained from multiple CNNs in a particular layer with a specified percentage of features in the first level for getting relevant features from various models. Then, we combined those features and did the second level of feature selection to select the most informative features. Both levels of feature selection were carried out using the light gradient boosting machine (LightGBM) algorithm. The final feature set has been used to classify COVID-19 and non-COVID-19 chest X-ray images using the support vector machines (SVM) classifier. The proposed model's performance was evaluated and analyzed on the open-access dataset. The highest accuracy was 99.80% using only 5% of the features extracted from ResNet50 and Xception. The other way of combining the ensemble of deep features and a few recent works for the classification of COVID-19 were also compared with the proposed model. As a result, our proposed model has achieved the best success rate for this dataset and may be deployed to support decision systems for radiologists.

Keywords— Chest X-ray; convolutional neural network; coronavirus; feature selection; light gradient boosting machine; transfer learning.

Manuscript received 15 Nov. 2021; revised 25 Dec. 2021; accepted 16 Jan. 2022. Date of publication 31 Oct. 2022.
IJASEIT is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

The coronavirus disease (COVID-19) is an extremely contagious and dangerous virus infection caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) that resulted in a global pandemic and a huge loss of human life [1]. It first appeared in December 2019 in Wuhan City, China, where it caused severe pneumonia of unknown origin [2]. COVID-19 was proclaimed a worldwide public health emergency by World Health Organization (WHO) on January 30, 2020 [3], and designated a pandemic three months later [4]. Since November 22, 2021, WHO reports that the overall number of people affected by this disease was 256,966,237, with about 5,151,643 deaths [5].

Early detection and treatment of COVID-19 are critical for illness prevention and control, and it can drastically reduce the spread of this disease and improve patient recovery rates. WHO recommends that all diagnoses of COVID-19 need to be performed using the reverse transcription-polymerase

chain reaction (RT-PCR). However, this procedure is time-consuming, inconvenient, and expensive [6], producing inaccurate results and a slow turnaround time [7]. Testing facilities are also insufficient in many places affected by the COVID-19 outbreak [8]. In addition, some people, particularly small children, find deep nasal swabs irritating. In time-sensitive conditions, relying only on the RT-PCR test may be insufficient for diagnosing this disease.

A vaccination program that is secure and safe would be a huge success. However, only a few effective vaccines have been discovered to the present, and it is believed that it will take an inordinate amount of time to protect the entire world against the hazards of COVID-19 [9]. Furthermore, the coronavirus evolves, impeding the development of a vaccine. In this situation, radiological scans may be essential for diagnosing this condition accurately. A few radiologists propose chest X-rays to diagnose COVID-19 cases, given that most radiological laboratories and hospitals have X-ray machines capable of capturing chest images [10]. When the number of COVID-19 patients increases rapidly,

overburdening public health systems may result in a shortage of doctors and radiologists to review X-ray images. Computer-aided diagnostic (CAD) systems may also be a potential choice in this context for supporting doctors with medical diagnoses. It is possible to receive a second opinion from these systems because they employ computational approaches for image processing and analysis. This can be helpful in cases when establishing a diagnosis by the human eye is difficult.

Numerous seminal studies have been published on the prediction of COVID-19 using X-ray images. El-Kenawy et al. [11] extracted features from ResNet50 and enhanced a multi-layer perceptron classifier using the advanced squirrel search optimization technique. The experiment was validated against the pneumonia dataset, comprising 5863 images with an average accuracy of 99.26%. In addition, the authors implemented their model on a chest X-ray COVID-19 dataset acquired from GitHub and achieved a mean accuracy of 99.7%. Sahlol et al. [12] pursued a similar technique by selecting the required qualities using an architecture called Inception and a swarm-based feature selection method. The authors assessed the accuracy at 98.7% and 99.6% using two publicly available COVID X-ray datasets.

In another study, Bhowal et al. [9] presented a transfer learning-based technique to extract features, VGG16, Xception, and InceptionV3, using pre-trained models. After performing feature selection, the obtained features will be sent to a multi-layer perceptron for classification. This technique was evaluated using a dataset comprising 752 COVID-19 and 1584 pneumonia records, and 1639 normal chest X-ray images gathered from various sources. The study achieved 93.45% accuracy utilizing two-tier feature selection algorithms: Coalition game and Nystrom sampling. According to it, feature selection can improve performance and has additional benefits, such as shorter processing times and reducing the number of related, irrelevant, or noisy variables.

Numerous studies have also been conducted to optimize the performance of classifiers that utilize CNN-extracted features. Abraham and Nair [13] examined the effectiveness of multi-efficacy CNNs in identifying COVID-19 based on X-ray images. This disease was also predicted using a CNN feature concatenation. The features were selected using correlation and a BayesNet classifier. The technique was validated using two publicly available datasets and yielded positive results. The technique obtained 91.16% accuracy in the first dataset and 97.44% accuracy in the second dataset. In addition, the study affirmed that the use of multiple pre-trained CNN outperformed a single CNN. Barua et al. [14] merged nine deep features using the fully connected layer of AlexNet, VGG16, and VGG19. In the next step, SVM was used to classify the most informative features, and this method was achieved at 99.64 percent accuracy.

Furthermore, Turkoglu [15] also introduced the COVIDetectionNet model. This study used a pre-trained CNN-based AlexNet architecture with transfer learning. The SVM technique was utilized to categorize the essential features extracted from all levels of the architecture using the Relief algorithm. The proposed model was evaluated on 6092 X-ray images of Normal, COVID-19, and Pneumonia with a 99.18 percent accuracy. Moreover, Reda et al. [16] combined

characteristics collected from three pre-trained CNN models, namely resnet18, resnet25, and densenet201. After optimizing each feature vector using the binary Butterfly algorithm, these characteristics were transferred to an ELM to classify chest X-ray images. The study succeeded in detecting covid-19 with 99.48% accuracy. Therefore, based on the study, combining the deep features extracted from different CNN architectures can enhance the accuracy and efficiency of the classification process.

This study investigates another technique for enhancing performance by combining CNN features. The feature selection process was implemented to ensure that the computational process ran as efficiently as possible. VGG16, VGG19, ResNet50, InceptionV3, and Xception are five convolutional neural networks (CNNs) employed in this study. These were combined with the transfer learning technique, which does not restrict the layer used to generate features to the fully connected layer. Furthermore, two-level feature selection using LightGBM, which is a powerful boosting method, was used to guarantee that only the most informative features are selected and used as input to the SVM classifier to achieve better classification results than utilizing only features generated from a single CNN.

The following is a concise explanation of the proposed method's primary contribution:

- 1) Unlike most studies that focus solely on the use of features generated from fully connected layers of the CNN architecture, this study employs features obtained from various layers, including the convolutional and pooling layers.
- 2) To further improve COVID-19 classification, deep features extracted from two or three different CNN models are concatenated to maximize each CNN's informative feature extraction capabilities.
- 3) A two-level feature selection algorithm that identifies superfluous and uninformative features was used to collect relevant features from numerous models in the first level. They were obtained from multiple CNNs in a particular layer with a defined proportion of features, and the most informative ones were selected at the second level. The LightGBM algorithm was used to do both stages of feature selection.
- 4) A hybrid study was conducted utilizing features derived from several CNN models. Furthermore, two-level feature selection was performed through LightGBM and SVM was used as a high-performance classifier.
- 5) A high-accuracy decision-making model has been offered as a second opinion to aid radiologists in diagnosing COVID-19.

II. MATERIAL AND METHODS

The proposed COVID-19 X-ray classification approach starts by extracting discriminant features from raw pictures without major pre-processing or segmentation using a dataset of COVID-19 and normal X-ray images. Additionally, a light gradient boosting machine (LightGBM) derives important features from data using a two-level feature selection technique. Finally, the optimal feature is passed to Support Vector Machines (SVM) to obtain the classification result.

This section discusses the details of the dataset and methodologies used in this study, such as CNN as a feature extractor, LightGBM as feature selection, and SVM as a classifier. The proposed model and the other way of combining the ensemble of deep features as the main comparison of the proposed model are also discussed.

A. Dataset

The dataset published by Chowdhury et al. [17] on Kaggle had two classes, namely COVID-19 and normal, and was utilized. Furthermore, 2400 X-ray images were analyzed in total, namely 1200 COVID-19 and 1200 normal. Fig. 1 depicts a sample of chest X-ray images from the database.

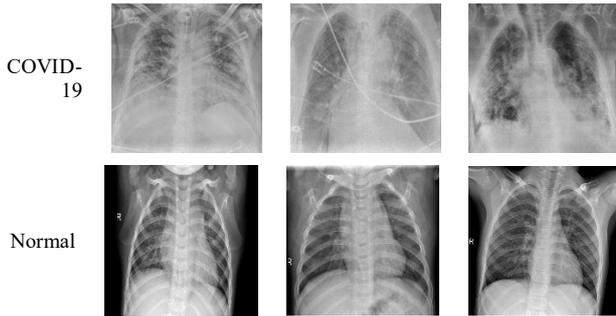


Fig. 1 sample of chest X-ray images

B. Data Pre-processing

The images were randomly divided into various categories for the chest X-ray datasets, namely 60% training, 20% validation, and 20% testing. Table 1 shows the number of images utilized for training, validation, and testing in the experimental trials.

TABLE I
NUMBER OF IMAGES IN EACH DATA CLASS FOR PERFORMANCE EVALUATION OF THE PROPOSED METHOD

	COVID-19	Normal	Total
Training data (60%)	720	720	1440
Validation data (20%)	240	240	480
Testing data (20%)	240	240	480

The training set is used to train the model, the validation set is used to choose parameters, such as the layer position for feature extraction and the feature percentage, depending on the accuracy of the trained model, and the testing set is used to evaluate the model's performance against new data. It is critical to note that the identical image sets established for training, validation, and testing were utilized in all experiments, and this implies that the model was uninformed of the image sets used for validation and testing.

It is critical to pre-process the image before utilizing it as the model's input. This can be performed by resizing them to the same size to support the current dataset to the convolutional neural network. This study employed the pre-trained networks VGG16, VGG19, ResNet50, InceptionV3, and Xception for experimental analysis. Furthermore, the size of the input X-ray images utilized in this investigation varies. VGG16, VGG19, and ResNet50 had 224x224 inputs, while InceptionV3 and Xception had 299x299 inputs.

C. Platforms and Hardware

This experiment was implemented in Python using the Keras [18] package and Tensorflow [19] as the backend deep learning framework. It runs on Google Colaboratory [20], which provides a Tesla V100 GPU with 16,94 GB of GPU memory and 54.8 GB of RAM. In addition, it can execute in the background, which means that the notebook will continue to function even when the internet connection is lost or the browser is closed.

D. Convolutional Neural Network as Feature Extractor

Deep learning (DL) is a subfield of artificial intelligence (AI) that enables the construction of end-to-end models capable of providing remarkable results without necessarily requiring feature extraction [21, 22]. This type of architecture enables these networks to find intricate details hidden from simple networks. The greater the number of convolution layers utilized, the more comprehensive the obtained features are [23]. CNN-based models are more efficient and accurate than more standard machine learning techniques, and these models extract salient features and achieve a high degree of classification accuracy. CNN's fundamental premise is to capture local features at their earliest levels and combine them to build more sophisticated features. By avoiding matrix multiplication, CNN overcomes the shortcomings of Feed-Forward Neural Networks and Multi-Layer Perceptron [24]. The operation used in CNN is convolution, which is frequently signified in Eq. (1) where x denotes the input and w the kernel, while the output s is called the feature map [25]. This highly effective strategy was implemented in this study due to the high-dimensional nature of COVID-19 diagnosis using X-ray images.

$$s(t) = (x * w)(t) \quad (1)$$

There is a constraint on developing a computer-aided diagnostic system using CNN. To produce more accurate predictions, the CNN model must be trained on a big dataset containing various potential variants [26, 27]. Numerous CNN architectures have been built in the literature for handling a wide variety of classes. However, these designs were meant to be robust when trained on big datasets but tend to overfit when trained on smaller datasets. Acquiring enormous labeled medical images suitable for DL-based COVID-19 screening has become problematic. However, it is possible to use pre-trained CNN models on a large database such as ImageNet [28]. The notion of transfer learning, which involves the transfer of features and weights from a pre-trained model to new training models and problems with less data, was used in this study. This is because robust deep learning models may be generated with far less training data through transfer learning rather than starting from scratch, maximizing classification performance over unseen test images. In addition, pre-trained CNNs can detect COVID-19 effectively [29-31]. Thus, rather than employing the CNN model for COVID prediction, it was used to extract relevant features and then feed the generated feature vector to LightGBM as a Feature Selection (FS) approach, followed by an SVM classifier-based classification of the input chest X-ray images.

The CNN models utilized in this study include VGG16 [32], VGG19 [32], ResNet50 [33], InceptionV3 [34], and

Xception[35], all of which were pre-trained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset. Furthermore, default parameter values were used for all pre-trained networks to create multi-CNN.

VGG16 and VGG19 are part of an innovative design from the Visual Geometry Group (VGG). Simonyan and Zisserman [32] presented a straightforward and efficient design paradigm for CNN. This involves adding a layer of 3×3 filters to the heap of 5×5 and 11×11 filters, making it more efficient. By minimizing the number of parameters, an additional benefit of reducing computational complexity was realized using small-size filters.

ResNet is comparable to the VGG network but is approximately eight times deeper. It was designed by He et al. [33] on the basis of deep architectures that have demonstrated strong convergence properties and high accuracy. ResNet50 was constructed using numerous stacked residual units and evolved with 50 layers, 49 of which are convolutional and one completely linked at the network's end. Convolutional, pooling, and layering units of the preceding layer ($x_l - 1$). After several processes, such as convolution with variable-size filters or batch normalization, followed by the application of an activation function such as ReLU to ($x_l - 1$), the result obtained was $F(x_l - 1)$. The final residual output was x_l , which is expressed mathematically in Eq. (2).

$$x_l = F(x_l - 1) + x_l - 1 \quad (2)$$

The residual network has many fundamental residual blocks. The operations of these blocks can be modified according to the residual network architecture type [33, 36].

InceptionV3 is a convolutional deep neural network design that is commonly utilized for classification problems. Furthermore, Szegedy et al. [34] pioneered the model notion in the GoogleNet [37] architecture, and suggested InceptionV3 by modifying the inception module. The InceptionV3 network is composed of several symmetric and asymmetric construction blocks, each of which has several branches of convolutions, average and maximum pooling, concatenation, dropouts, and fully linked layers. This network contains a total of 42 layers and 29.3 million parameters.

The primary trait of Xception is its extreme inception architecture and has basic concept of depth-separable convolution [35]. Xception model enhanced the original inception block by replacing a single dimension (3×3) with a 1×1 convolution to minimize computational complexity. By utilizing the decoupling channel and spatial correspondence, the Xception network becomes significantly more computationally efficient. It applies 1×1 convolutions in transferring the convolved output to the embedding short dimension. Furthermore, it also applies k spatial transformations. This is important to note that k in this case denotes the cardinality which defines the breadth of the transformation as determined by the number of outcomes. However, in Xception, the computations were simplified by explicitly convolving each channel around the spatial axes. These axes are then employed to achieve cross-channel correspondence as the 1×1 convolutions (pointwise convolutions). This convolution is used in Xception to normalize the channel's depth. The Xception transformation strategy improves learning speed and performance but does not lower the parameter count [38].

A combination of pre-trained CNN features is expected to enhance the performance of computer-aided diagnostic systems. Therefore, concatenating feature sets maximizes the ability of each CNN to extract meaningful and discriminative features. This study aims to provide a method for predicting COVID-19 utilizing features collected from various pre-trained neural networks. Initially, features were generated from chest X-ray images using a CNN model. However, the resulting feature vector size is extremely huge, such as 200,000 feature attributes, not to mention the size of combined feature vectors.

TABLE II
THE NUMBER OF FEATURES EXTRACTED FROM DIFFERENT LAYER OF DIFFERENT CNN

CNN Architecture	Name of Layer	Number of Features
VGG16	block3_pool	200,704
	block4_pool	100,352
	block5_conv1	100,352
	block5_conv2	100,352
	block5_conv3	100,352
	block5_pool	25,088
	fc1	4,096
	fc2	4,096
VGG19	block3_pool	200,704
	block4_pool	100,352
	block5_conv1	100,352
	block5_conv2	100,352
	block5_conv3	100,352
	block5_conv4	100,352
	block5_pool	25,088
	fc1	4,096
ResNet50	fc2	4,096
	conv4_block1_add	200,704
	conv4_block2_add	200,704
	conv4_block3_add	200,704
	conv4_block4_add	200,704
	conv4_block5_add	200,704
	conv4_block6_add	200,704
	conv5_block1_add	100,352
	conv5_block2_add	100,352
	conv5_block3_add	100,352
InceptionV3	avg_pool	2,048
	mixed0	313,600
	mixed1	352,800
	mixed2	352,800
	mixed3	221,952
	mixed4	221,952
	mixed5	221,952
	mixed6	221,952
	mixed7	221,952
	mixed8	81,920
	mixed9_0	49,152
	concatenate	49,152
	mixed9	131,072
mixed9_1	49,152	
concatenate_1	49,152	
mixed10	131,072	
Xception	avg_pool	2,048
	add_11	102,400
	avg_pool	2,048

There are two methods for combining CNN feature vectors. The first one is after combining all retrieved features; the feature vector is sent to LightGBM. The retrieved features

from each CNN are first picked and merged in the second method. However, because the features are mixed from several sources, there is a possibility that certain features will perform inadequately when combined. As a result, the second phase of feature selection is used. This approach is referred to as two-level feature selection using the LightGBM algorithm. This ensemble approach concatenates the output of many CNNs into a common subset.

Moreover, unlike most of the studies, which focus exclusively on fully connected layers of the CNN architecture, this study extracts information from a variety of layers, including the convolutional and pooling layers. The length of the feature vector generated by the various layers of each CNN architecture is shown in Table 2.

E. Light Gradient Boosting Machine as Feature Selection

Feature extraction is critical in computer vision, image processing, and pattern recognition systems. It has been utilized in various models to extract deep features [9]. As previously stated, CNN models extract and choose only the most representative features from the input image. Furthermore, it is envisaged that the integration of pre-trained CNN features would improve the performance of computer-aided diagnostic systems. However, the number of features extracted by CNNs is not necessarily relevant, as the number of features extracted is directly connected to the architecture utilized, specifically while training images are few. The more depth a network has, the more features it can extract. As a result, the possibility that some of the extracted features are redundant was investigated.

Features are rarely directly fed into classifiers because of their high dimensionality, resulting in decreased accuracy, increased training time, and overfitting [12]. As a result, working with a smaller collection of features can have some advantages, including faster processing times and the elimination of features that contribute nothing to the classification process [26], [27], [39]. However, the feature selection process does not always improve classification accuracy [40], and achieving comparable performance with fewer features is a desirable development. The consequences of removing too many features on accomplishment were examined using LightGBM as the feature selection method.

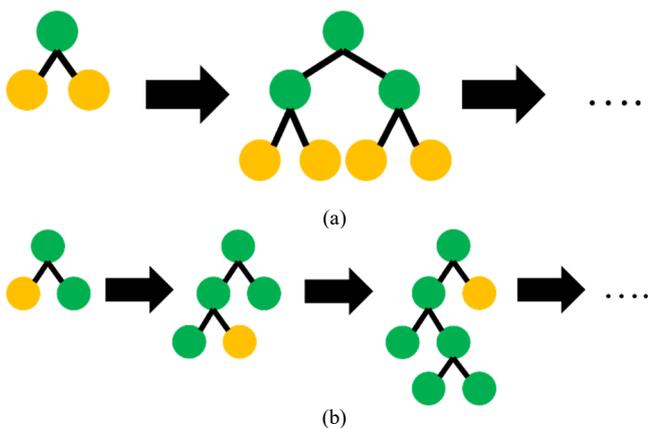


Fig. 2 Illustration of (a) level-wise and (b) leaf-wise tree growth

Based on the fact that LightGBM [41] has a quicker training time and greater efficiency, it was employed in

feature selection. It implements gradient boosting decision trees (GBDT), an ensemble approach for serially combining weak learners (boosting). Combining decision trees improves the model by having each new learner fit the residuals from the prior tree. The final model incorporates the data from each phase, resulting in a strong learner. Moreover, LightGBM is similar to XGBoost but differs in a few key areas, most notably in the way the tree or base learners are created. In contrast to previous ensemble approaches, LGBM develops tree leaf-wise rather than level-wise, which helps to minimize loss throughout the sequential boosting phase. The difference between level-wise and leaf-wise tree growth is shown in Fig. 2.

After a comprehensive review of trial and error using several hyperparameters, boosting type='gbdt', learning rate=0.05, number of estimators=100, and objective='binary' was selected for the proposed task. Furthermore, this feature selection was only used on the training set, and the features from the validation and test sets were chosen using the selected feature indices. At the end of the algorithm, the optimal feature subset was determined and then utilized for classification.

F. Support Vector Machine as Classifier

Vapnik [42] pioneered the use of Support Vector Machines (SVM) for classification. The SVM approach divides data into classes by constructing an optimal margin separator (hyperplane). The optimal hyperplane is one that is placed at the greatest distance between support vectors of different classes, and the nearest data point to the hyperplane is designated as the support vector. As a result, optimization processes are required to create a hyperplane that can adequately generalize the data and is located at the same and greatest distance from the support vector for each class. Fig. 3 illustrates the hyperplane, margin, and support vector.

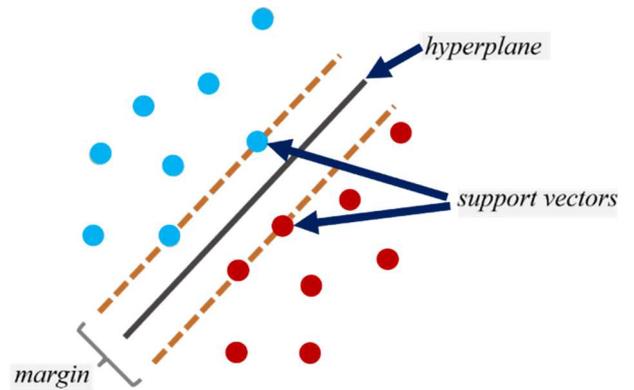


Fig. 3 Illustration of the hyperplane, margin, and support vector on SVM (modification from [43])

Let $T = \{(x_i, y_i)\}$, $i = 1, 2, \dots, N$ and $x_i \in \mathbb{R}^n$ is a data point. N represents the number of images and $y_i \in \{-1, +1\}$ denotes the normal and COVID X-ray images, respectively. If there is a vector w and scalar b , a proper inequality can be created that divides the data set into two classes; the data set can be separated linearly. The inequalities are given in Eq. (3) [44].

$$y_i(w \cdot x_i + b) \geq 1 \text{ for } i = 1, 2, \dots, N \quad (3)$$

Eq. (4) expresses that the hyperplane equation separating classes -1 and $+1$ is given where \mathbf{w} is perpendicular to the hyperplane and b is the bias.

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \quad (4)$$

The problem that SVM must solve is an optimization problem in which we must maximize the margin, as shown in Eq. (5) while adhering to the constraints specified in Eq. (6), where N is the number of data points.

$$\min_{\mathbf{w}, b} \left(\frac{1}{2} \|\mathbf{w}\|^2 \right) \quad (5)$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i = 1, 2, \dots, N \quad (6)$$

Cortes and Vapnik [44] introduced a modified form of SVM that allows for errors in classification. This type of SVM is also known as soft margin SVM. The SVM soft margin aims to find the optimal hyperplane with the minimum number of misclassifications. The SVM soft margin optimization problem is shown in Eq. (7). It was modified by adding variables ζ and C to the previous optimization problem.

$$\min_{\mathbf{w}, b, \zeta} \left(\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \zeta_i \right) \quad (7)$$

subject to the constraints

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \zeta_i \quad (8)$$

$$\zeta_i \geq 0, \quad \forall i = 1, 2, \dots, N \quad (9)$$

G. Proposed Method

This study describes a comprehensive machine learning-based approach for automatically identifying COVID-19 from X-ray images. To extract discriminative features from X-ray images, pre-trained CNN models such as VGG16 [32], VGG19 [32], ResNet50 [33], InceptionV3 [34], and Xception[35] were used. After collecting deep features, the LightGBM algorithm was used to choose the most informative features. Then, these features were merged and used as the input to SVM. Furthermore, this study reveals that merging the deep features retrieved from the various layers of different CNN architectures improves the classification process efficiency. A two-tiered feature selection strategy shows this. First, relevant features were selected from numerous models by combining the features generated from multiple CNNs in a single layer with a predetermined proportion of features as shown in Fig. 4. Afterwards, the features were aggregated, and a second round of feature selection was performed to determine which features enhance classification accuracy as shown in Fig. 5. This proposed method was also compared to the other way of combining the ensemble of deep features that is directly combining all extracted features and then sent to LightGBM.

In order to quantify and examine the performance of the selected approaches, an SVM classifier was used with accuracy as the performance parameter. This formula is shown in the equation below.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{total number of data}} \times 100\% \quad (10)$$

The following parameters were used in calculating the accuracy, namely TP represents the number of correctly

predicted COVID-19, while TN is the number of correctly predicted normal x-ray images.

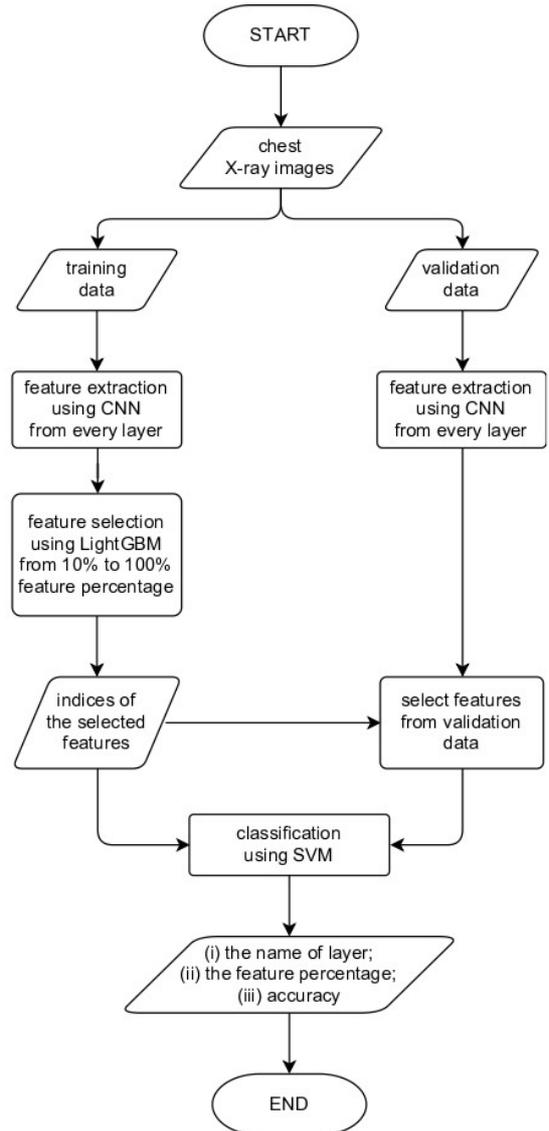


Fig. 4 The first level of the feature selection algorithm

III. RESULTS AND DISCUSSION

As previously explained, the COVID-19 disease was diagnosed using the chest X-ray dataset [17]. In order to evaluate the proposed automated COVID-19 detection and classification system, the following tests were conducted:

- Experiment 1—The dataset was used to examine the classification performance of the five CNN models.
- Experiment 2—Using 10% to 90% of features, the hybrid approach CNN-LightGBM-SVM was applied in different layers of CNN. The experiment's outcome with the best feature percentage was then compared to CNN-SVM with no feature selection.
- Experiment 3—Features derived from several CNNs are integrated and then chosen using LightGBM prior to being sent to the SVM classifier.
- Experiment 4—Finally, CNN-SVM is used to construct two-level feature selection using LightGBM, and the outcome is compared to experiment 4.

The first experiment classified the COVID-19 images using the original five pre-trained CNN models. The features were first acquired from the dataset's images in each model, and then the image was classified using the softmax classifier. The Adam optimizer was employed with the following parameters: learning rate=0.00005, batch size=32, and epoch=50. The experimental analysis is summarized in Table

3. The accuracy stated here is on the test set, and it can be shown that VGG16 surpasses other deeper architectures, such as ResNet50, InceptionV3, and Xception. Remarkably, deeper models outperform shallower models on the most recent COVID-19 datasets. This is most likely due to the size and quality of presently available datasets, which result in the model being overfitting.

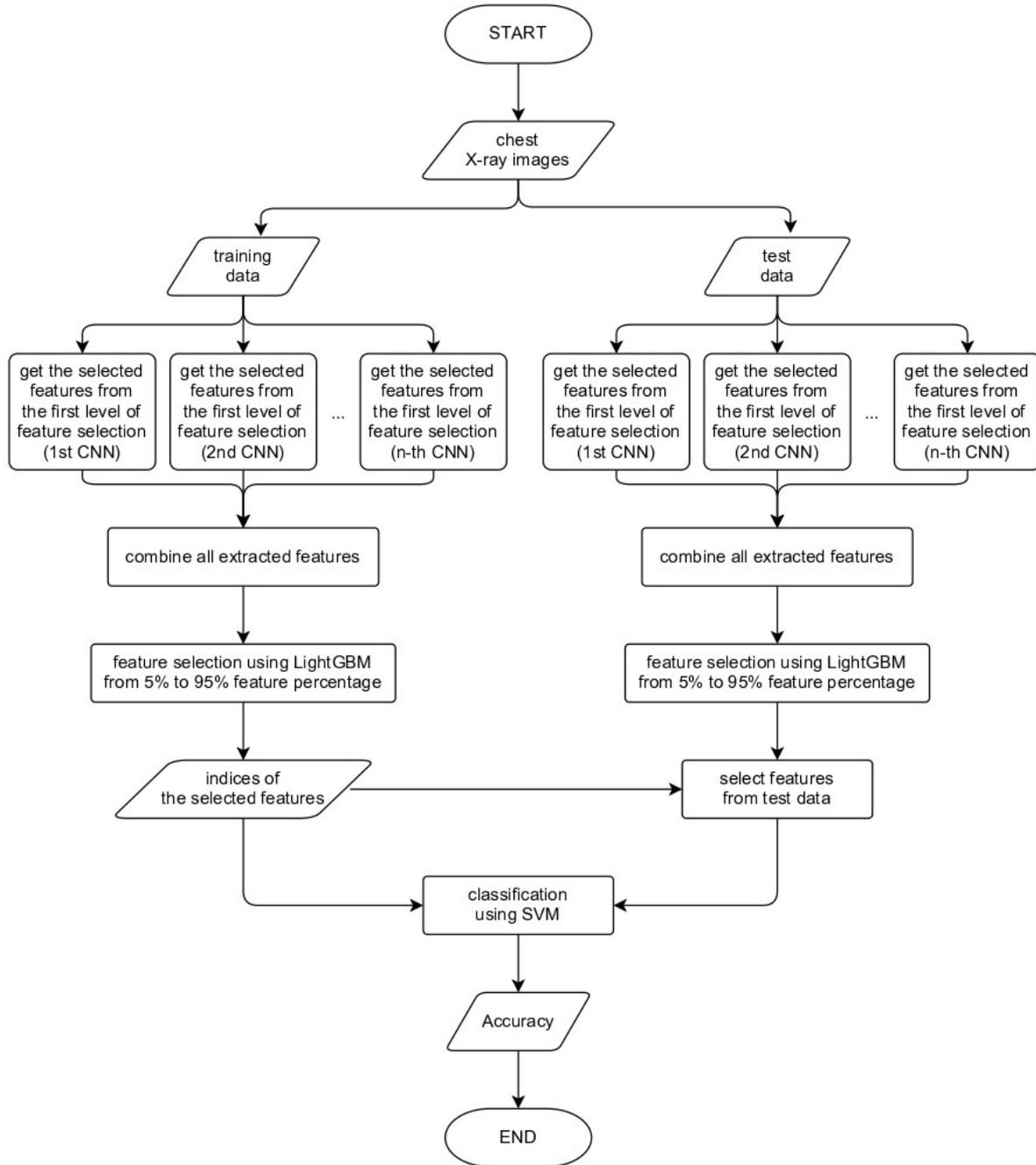


Fig. 5 The second level of feature selection algorithm

TABLE III
THE PERFORMANCE OF CNN MODEL

CNN Model	Accuracy (%)	Running Time (s)
VGG16	99.59	18.25
VGG19	97.75	18.24
ResNet50	62.78	19.99
InceptionV3	50.92	23.54
Xception	50.92	21.15

The best result was achieved using the VGG16 model, which had a 99.59 percent accuracy and required 18.25 seconds to test. This is unsurprising, given that Simonyan and Zisserman [32] noted in their study that the VGG architecture delivers great performance even when employed with basic pipelines. However, this study demonstrates that the proposed method used improves classification accuracy.

The second experiment examines the performance of CNN-LightGBM-SVM employing 10% to 90% of the validation dataset's extracted features. The best feature percentage was then used to demonstrate the CNN-LGBM-SVM method's performance. It is worth noting that the hybrid technique CNN-LGBM-SVM is used across many layers of CNN. In addition, $C=1.0$, kernel=Radial Basis Function (RBF), and $\gamma=1/(n \text{ features} * X.\text{var}())$ were used as the SVM hyperparameter value.

Tables IV to VIII compared the performance of CNN-SVM with and without feature selection using LightGBM. In general, the findings indicated that the suggested CNN could extract robust features, which enabled the classifiers to attain promising performance in terms of X-ray image classification since all the classifiers produced comparable results. In most situations, incorporating LightGBM with CNN-SVM improves accuracy. As a result, it demonstrates that a combination of deep features and a feature selection technique is useful for image classification. Furthermore, Table IV to VIII demonstrates that the classification accuracy obtained from this experiment is identical to that obtained from the original VGG16 model. When ResNet50 was used, the greatest accuracy of CNN-LightGBM-SVM obtained was 99.59%. However, compared to the original ResNet50 model, using ResNet50 as a feature extractor and SVM as a classifier is sufficient in boosting accuracy to 99.39%, which is a 58.31% improvement. By utilizing LightGBM to execute feature selection, the accuracy was increased to 99.59%.

TABLE IV
THE COMPARISON OF VGG16-SVM AND VGG16-LIGHTGBM-SVM PERFORMANCE

VGG16's layer	VGG16-SVM		VGG16-LGBM-SVM	
	Acc (%)	Time (s)	Acc (%)	Time (s)
block3_pool	98.98	64.21	99.18	50.33
block4_pool	98.77	37.72	99.18	23.95
block5_conv1	98.98	34.35	99.18	19.62
block5_conv2	99.18	32.75	99.39	19.74
block5_conv3	98.77	36.16	98.77	19.66
block5_pool	98.98	21.98	98.98	18.97
fc1	98.77	19.34	98.36	18.86
fc2	98.57	19.54	98.16	18.67

TABLE V
THE COMPARISON OF VGG19-SVM AND VGG19-LIGHTGBM-SVM PERFORMANCE

VGG19's layer	VGG19-SVM		VGG19-LGBM-SVM	
	Acc (%)	Time (s)	Acc (%)	Time (s)
block3_pool	98.57	49.89	98.77	41.30
block4_pool	98.77	38.77	98.77	24.92
block5_conv1	98.77	35.69	99.18	20.23
block5_conv2	98.77	34.28	98.98	20.17
block5_conv3	98.77	34.39	98.77	20.57
block5_conv4	98.16	37.08	97.96	20.63
block5_pool	98.57	22.76	99.18	21.99
fc1	98.77	19.31	98.77	19.44
fc2	98.57	19.38	98.36	19.33

TABLE VI
THE COMPARISON OF RESNET50-SVM AND RESNET50-LIGHTGBM-SVM PERFORMANCE

ResNet50's layer	ResNet50-SVM		ResNet50-LGBM-SVM	
	Acc (%)	Time (s)	Acc (%)	Time (s)
conv4_block1_add	99.39	51.20	99.18	25.59
conv4_block2_add	99.18	50.07	99.59	43.51
conv4_block3_add	99.39	54.55	99.39	45.87
conv4_block4_add	99.18	57.63	99.18	48.96
conv4_block5_add	99.18	55.73	99.18	50.04
conv4_block6_add	99.18	54.25	98.77	23.99
conv5_block1_add	99.18	32.81	98.77	21.84
conv5_block2_add	98.36	35.77	98.77	25.12
conv5_block3_add	98.16	39.18	98.36	34.32
avg_pool	98.16	20.72	98.16	20.96

In the fourth experiment, a multi-CNN model was used as a feature extractor, which provides a feature vector for initial feature screening using LightGBM techniques. Afterward, this study incorporated the selected feature, and the effect of the utilization percentage of the total features was also examined. Table IX specifies the layer positions for each CNN. A collection of features was extracted from the model's specific layers utilizing a predefined percentage of features based on the model's previously shown promising performance in the CNN-SVM. The performance of the multi-CNN-LightGBM-SVM is summarized in Table XI.

TABLE VII
THE COMPARISON OF INCEPTIONV3-SVM AND INCEPTIONV3-LIGHTGBM-SVM PERFORMANCE

InceptionV3's layer	InceptionV3-SVM		InceptionV3-LGBM-SVM	
	Acc (%)	Time (s)	Acc (%)	Time (s)
mixed0	99.18	74.16	98.98	67.65
mixed1	98.98	90.12	98.98	80.38
mixed2	99.18	109.81	99.18	97.48
mixed3	98.57	67.09	98.77	59.81
mixed4	98.98	63.72	99.18	58.02
mixed5	98.77	58.32	98.77	56.69
mixed6	98.77	60.56	98.77	56.15
mixed7	98.57	58.83	98.57	54.10
mixed8	98.57	34.99	98.57	36.35
mixed9_0	97.75	33.38	97.55	31.53
concatenate	98.36	32.97	98.57	31.41
mixed9	98.16	49.83	98.36	45.90
mixed9_1	97.34	37.00	97.55	35.39
concatenate_1	97.55	33.51	97.55	31.88
mixed10	97.96	55.12	97.96	50.13
avg_pool	97.55	23.29	97.34	24.13

TABLE VIII
THE COMPARISON OF XCEPTION-SVM AND XCEPTION-LIGHTGBM-SVM PERFORMANCE

Xception's layer	Xception-SVM		Xception-LGBM-SVM	
	Acc (%)	Time (s)	Acc (%)	Time (s)
add_11	98.77	38.79	98.16	20.94
avg_pool	98.36	21.10	98.57	25.91

As shown in Table XI, this approach has the highest accuracy of 99.39%. Although it is reliable, it falls short of the performance of the original VGG16. This is possible

because of the enormous dimension of features that remain owing to the usage of feature percentages rather than a set number of features.

A multi-CNN model was used as a feature extractor in the fifth experiment. It generates a feature vector with certain dimensions that is processed through LightGBM approaches for initial feature screening. After the chosen feature was merged, the collection of features derived from the images must be informative to facilitate classification and adequate to prevent classification mistakes. As a result, the second feature selection was used to exclude features that are undesirable for usage in this union of features.

TABLE IX
THE CHOSEN LAYER NAME TO USE IN MULTI-CNN-LIGHTGBM-SVM IN THE EXPERIMENT 4

CNN Architecture	Layer Name
VGG16	block5_conv2
VGG19	block5_conv3
ResNet50	conv5_block1_add
InceptionV3	mixed8
Xception	add_11

The percentage of total features and the layer position where they would be generated were examined. In the first level of feature selection, the length of the feature vector obtained from each CNN is specified in Table X. A collection of features was retrieved from the model's particular layers using a specified percentage of features based on the model's prior promising performance in the CNN-LightGBM-SVM. Table XII details the performance of the proposed method.

According to Table XII, the best-performing model was built utilizing feature maps from two separate pre-trained deep learning models, namely ResNet50 and Xception. Combining the features gained through these approaches resulted in the feature maps. Furthermore, combining deep features extracted from several CNN models has been proven to increase performance in this and other experiments. Since the goal is to obtain greater performance with fewer features, LightGBM algorithms were used to select these feature maps. Following a detailed study of the findings in Tables III-XII, it was determined that employing two-level feature selection with LightGBM enhanced accuracy by 0.21 percent when compared to the original VGG16 model. This number confirms the suggested method's efficacy, since the test set was unknown to the created model. In addition, the proposed model improves prediction accuracy while reducing the feature dimension to 7,055 features by eliminating those that were redundant.

TABLE X
THE CHOSEN LAYER NAME AND FEATURE PERCENTAGE FOR OUR PROPOSED METHOD IN EXPERIMENT 5

CNN Architecture	Layer Name	Feature Percentage (%)	Selected Feature Length
VGG16	block5_conv2	10	10,035
VGG19	block5_pool	70	17,561
ResNet50	conv4_block2_add	70	140,492
InceptionV3	mixed4	10	22,195
Xception	avg_pool	30	614

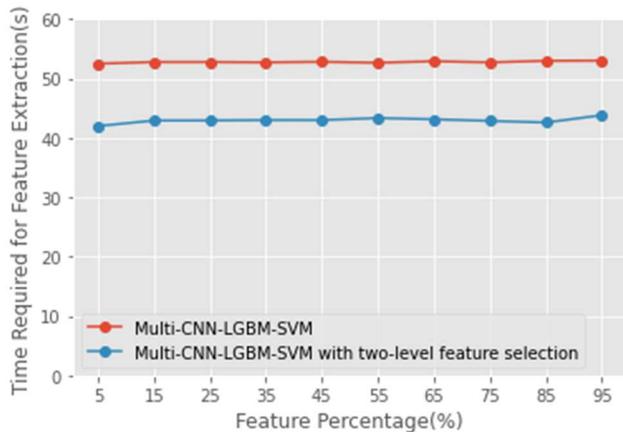
TABLE XI
THE PERFORMANCE OF MULTI-CNN-LIGHTGBM-SVM IN EXPERIMENT 4

Combined Architecture	Feature Percentage (%)									
	5	15	25	35	45	55	65	75	85	95
VGG16 and VGG19	98.77	97.96	98.77	98.98	99.18	99.18	98.77	98.98	98.98	99.18
VGG16 and ResNet50	98.77	97.96	98.77	98.98	99.18	99.18	99.18	99.18	99.18	99.18
VGG16 and InceptionV3	98.36	97.96	98.77	98.98	99.39	99.39	99.39	99.39	99.18	99.18
VGG16 and Xception	98.98	97.96	98.77	98.98	99.18	99.18	99.18	99.18	99.18	99.18
VGG19 and ResNet50	98.36	98.57	98.36	99.18	98.77	98.77	98.77	98.77	98.77	98.77
VGG19 and InceptionV3	98.36	97.96	98.36	99.18	98.77	98.98	98.98	98.98	98.77	98.77
VGG19 and Xception	98.77	98.57	98.36	99.18	98.77	98.77	98.77	98.77	98.77	98.77
ResNet50 and InceptionV3	98.36	98.36	98.36	98.77	99.18	99.18	99.18	99.18	99.18	99.18
ResNet50 and Xception	98.57	98.36	98.57	98.98	99.18	99.18	99.18	99.18	99.18	99.18
InceptionV3 and Xception	98.57	98.57	98.77	98.98	98.77	98.98	98.98	98.98	98.77	98.77
VGG16, VGG19, and ResNet50	98.98	98.57	99.18	99.39	99.39	99.39	99.39	99.39	99.39	99.39
VGG16, VGG19, and InceptionV3	98.77	98.77	98.98	99.18	99.39	98.77	98.98	98.98	98.98	98.98
VGG16, VGG19, and Xception	98.98	98.77	99.39	99.39	99.39	98.77	98.16	97.75	98.36	98.16
VGG16, ResNet50, and InceptionV3	98.57	98.36	98.36	98.77	98.98	98.57	98.98	98.98	98.98	98.98
VGG16, ResNet50, and Xception	97.96	98.57	98.57	98.77	99.18	98.36	98.16	97.34	97.34	97.34
VGG16, InceptionV3, and Xception	98.36	98.36	98.36	98.77	98.98	99.18	99.18	99.18	99.18	99.18
VGG19, ResNet50, and InceptionV3	97.34	97.96	97.96	98.57	99.18	98.77	98.77	98.77	98.77	98.77
VGG19, ResNet50, and Xception	97.14	97.34	97.55	98.77	99.18	98.77	98.77	98.77	98.77	98.77
VGG19, InceptionV3, and Xception	98.16	97.96	97.96	98.57	99.18	98.77	98.77	98.77	98.77	98.77

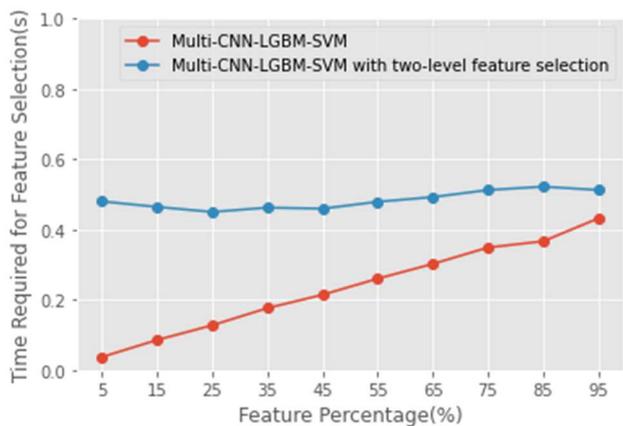
TABLE XII
THE PERFORMANCE OF OUR PROPOSED METHOD MULTI-CNN-LIGHTGBM-SVM WITH TWO-LEVEL FEATURE SELECTION IN EXPERIMENT 5

Combined Architecture	Feature Percentage (%)									
	5	15	25	35	45	55	65	75	85	95
VGG16 and VGG19	99.39	99.39	99.39	99.18	99.18	99.39	99.39	99.39	99.39	99.39
VGG16 and ResNet50	97.75	97.75	98.16	99.39	99.39	99.39	99.39	99.39	99.39	99.39
VGG16 and InceptionV3	98.98	98.98	98.98	99.18	99.39	99.39	99.39	99.39	99.39	99.39
VGG16 and Xception	98.98	98.77	98.77	98.77	99.39	99.39	99.39	99.39	99.39	99.18
VGG19 and ResNet50	98.77	98.98	99.18	99.39	99.18	99.18	99.59	99.59	99.59	99.59
VGG19 and InceptionV3	98.98	98.98	98.98	98.77	99.18	99.18	99.18	99.18	99.18	99.18

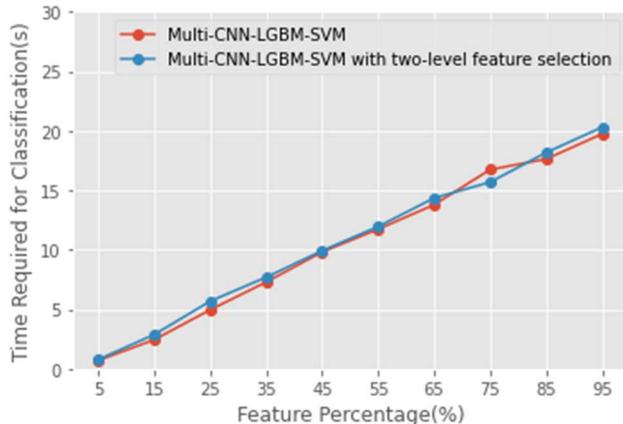
VGG19 and Xception	98.36	98.57	98.77	98.57	98.36	98.77	98.77	98.98	99.18	99.18
ResNet50 and Xception	99.80	99.39	99.39	99.39	99.39	99.39	99.39	99.18	99.18	99.59
InceptionV3 and Xception	99.18	99.18	98.98	99.18	98.98	99.18	99.18	99.18	99.18	99.18
VGG16, VGG19, and ResNet50	98.57	98.77	98.77	99.39	99.39	99.39	99.39	99.39	99.39	99.39
VGG16, VGG19, and InceptionV3	98.36	98.36	98.36	99.39	99.39	99.39	99.39	99.39	99.39	99.39
VGG16, VGG19, and Xception	99.59	99.59	99.59	99.39	99.39	99.39	99.39	99.39	99.39	99.39
VGG16, ResNet50, and Xception	97.75	97.96	98.16	99.18	99.39	99.39	99.39	99.39	99.39	99.39
VGG16, InceptionV3, and Xception	98.36	98.36	98.36	99.18	99.39	99.39	99.39	99.39	99.39	99.39
VGG19, ResNet50, and Xception	98.16	98.57	99.18	99.39	99.18	99.18	99.59	99.59	99.59	99.59
VGG19, InceptionV3, and Xception	97.55	97.75	97.96	98.36	99.18	99.18	99.18	99.18	99.18	99.18



(a)



(b)



(c)

Fig. 6 The comparison of time required for multi-CNN-LGBM-SVM with and without two-level feature selection in terms of (a) feature extraction, (b) feature selection, and (c) classification process.

Fig. 6 illustrates the time needed for extracting, selecting, and classifying the test data used in experiments 4 and 5. In terms of time complexity, Fig. 6(a) shows that extracting features from the multi-CNN-LGBM-SVM model used in experiment 4 takes much longer than the latter model. This might occur as a result of the first-level feature selection process, which eliminated a large number of features. Given the latter method's use of two-level feature selection, it's reasonable that the time required to select the feature is greater, as seen in Fig. 6(b). It is important to note that the superiority of multi-CNN-LightGBM-SVM is shown by the time efficiency associated with feature extraction and selection. After a comprehensive investigation of these graphs, it was found that the application of two-level feature selection speeds up the feature extraction process by ten seconds, even though the feature selection procedure takes 0.5 seconds longer. Meanwhile, the time needed to do classification using SVM is same for both models. As shown in Fig. 6(c), the more features utilized, the longer the SVM model takes to classify the data.

In this study, the proposed method was compared with previous studies' performance results that used a similar approach in implementing multi-CNN in classifying X-ray images, and these comparative results are presented in Table XIII.

TABLE XIII
THE COMPARISON OF OUR WORK WITH SOME SIMILAR PUBLISHED WORKS

Method	CNN Architecture	Acc (%)
CNN-Correlation FS-BayesNet [13]	SqueezeNet, DarkNet-53, MobileNetV2, Xception, ShuffleNet	97.44
COVID19FcNet9 (CNN-INCA-SVM) [14]	AlexNet, VGG16, and VGG19)	98.84
COVIDetectionNet (CNN-Relief-SVM) [15]	AlexNet	99.18
CNN-binary BOA-ELM [16]	ResNet18, ResNet50, and DenseNet201	99.48
Our proposed method (multi-CNN-LGBM-SVM with two-level feature selection)	ResNet50 and Xception	99.80

It is impossible to compare findings fairly owing to the disparity in datasets, performance measures, and validation procedures. However, as seen in Table XIII, the proposed technique outperforms all state-of-the-art methods and was found to be robust and efficient due mainly to its use of fewer features and a reasonable processing time. Therefore, as seen by the observed results, the second level of feature selection

can effectively select the most relevant features after the first screening. The proposed model for this dataset was better than the previous works. As a result, it can be concluded that the proposed two-level feature selection using LightGBM algorithm in multi-CNN-LightGBM-SVM is more capable of accurately predicting COVID-19 than its predecessors.

IV. CONCLUSION

The total number of COVID-19 infections remains high. As a result, early diagnosis of this disease is critical for effectively isolating affected individuals and breaking the transmission chain. A few radiologists recommend chest X-ray images for COVID-19 diagnosis since they are readily available and accessible in all hospital environments. In addition, it has been demonstrated that X-rays may be utilized efficiently to diagnose COVID-19. However, manual reading of many X-rays may increase erroneous detections due to workload and human eye vision problems. As a result, developing computer-aided diagnosis systems that utilize accessible radiological imaging is a viable alternative for undertaking diagnostic assessment and analysis of COVID-19 cases and can assist medical professionals.

This study presented a three-step deep learning model comprising a feature extractor based on transfer learning using a convolutional neural network (CNN), a two-level feature selection, and a feature classifier. The first phase involves selecting two or three CNNs from a group of five proposed CNNs to produce features from chest X-ray images. The retrieved feature set may contain information that is redundant or irrelevant. Therefore, eliminating irrelevant data is required before the classification step. In this process, LightGBM was used to perform a two-level feature selection. The first level selects the relevant feature from an extracted feature from a particular layer based on a defined proportion of features. Each feature vector is run through LightGBM to reduce unnecessary features and extract the most relevant ones in order to optimize CNN performance. As a secondary feature selection step, all the previously selected features obtained from different CNNs are combined, and LightGBM is used to achieve the final optimal features. In the last classification phase, support vector machines (SVM) is employed to classify COVID-19 and non-COVID-19 chest X-ray images. As a result, the suggested model improved 0.21% accuracy from 99.59 using VGG16 to 99.80% accuracy utilizing hybrid CNN-SVM with ResNet50 and Xception features. This performance outperforms numerous CNNs and recent COVID-19 image-processing studies.

Performance improves when dimensionality reduction techniques are used, showing a significant degree of correlation between features that should be reduced to improve the classifier's performance while also lowering the processing times. Thus, diagnostic effectiveness can be increased while radiologists avoid the substantial workload associated with the first COVID-19 screening. In future studies, it is intended to extend the suggested model as a software device to boost its accessibility and apply the proposed algorithms to a variety of medical image processing applications that employ additional imaging modalities.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Rahman et al. [45] for making available an open-access COVID-19 X-ray dataset.

REFERENCES

- [1] M. A. Shereen, S. Khan, A. Kazmi, N. Bashir, and R. Siddique, "COVID-19 infection: Emergence, transmission, and characteristics of human coronaviruses," *Journal of Advanced Research*, vol. 24, pp. 91-98, 2020/07/01/ 2020, doi: 10.1016/j.jare.2020.03.005.
- [2] T. Singhal, "A Review of Coronavirus Disease-2019 (COVID-19)," *The Indian Journal of Pediatrics*, vol. 87, no. 4, pp. 281-286, 2020/04/01 2020, doi: 10.1007/s12098-020-03263-6.
- [3] C. Sohrabi et al., "World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19)," *International Journal of Surgery*, vol. 76, pp. 71-76, 2020/04/01/ 2020, doi: 10.1016/j.ijsu.2020.02.034.
- [4] D. Cucinotta and M. Vanelli, "WHO Declares COVID-19 a Pandemic," (in eng), *Acta Biomed*, vol. 91, no. 1, pp. 157-160, 2020, doi: 10.23750/abm.v91i1.9397.
- [5] WHO. "WHO Coronavirus (COVID-19) Dashboard." <https://covid19.who.int/> (accessed 2021/11/22).
- [6] C. Perivolaropoulos and V. Vlachas, "A reduction of the number of assays and turnaround time by optimizing polymerase chain reaction (PCR) pooled testing for SARS-CoV-2," *Journal of Medical Virology*, vol. 93, no. 7, pp. 4508-4515, 2021/07/01 2021, doi: 10.1002/jmv.26972.
- [7] W. Wang et al., "Detection of SARS-CoV-2 in Different Types of Clinical Specimens," *JAMA*, vol. 323, no. 18, pp. 1843-1844, 2020, doi: 10.1001/jama.2020.3786.
- [8] F. Shi et al., "Review of Artificial Intelligence Techniques in Imaging Data Acquisition, Segmentation, and Diagnosis for COVID-19," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 4-15, 2021, doi: 10.1109/RBME.2020.2987975.
- [9] P. Bhowal, S. Sen, and R. Sarkar, "A two-tier feature selection method using Coalition game and Nystrom sampling for screening COVID-19 from chest X-Ray images," (in eng), *J Ambient Intell Humaniz Comput*, pp. 1-16, 2021, doi: 10.1007/s12652-021-03491-4.
- [10] C. Qin, D. Yao, Y. Shi, and Z. Song, "Computer-aided detection in chest radiography based on artificial intelligence: a survey," *BioMedical Engineering OnLine*, vol. 17, no. 1, p. 113, 2018/08/22 2018, doi: 10.1186/s12938-018-0544-y.
- [11] E.-S. M. El-Kenawy et al., "Advanced Meta-Heuristics, Convolutional Neural Networks, and Feature Selectors for Efficient COVID-19 X-Ray Chest Image Classification," (in eng), *IEEE access : practical innovations, open solutions*, vol. 9, pp. 36019-36037, 2021, doi: 10.1109/ACCESS.2021.3061058.
- [12] A. T. Sahlol, D. Youstri, A. A. Ewees, M. A. A. Al-qaness, R. Damasevicius, and M. A. Elaziz, "COVID-19 image classification using deep features and fractional-order marine predators algorithm," *Scientific Reports*, vol. 10, no. 1, p. 15364, 2020/09/21 2020, doi: 10.1038/s41598-020-71294-2.
- [13] B. Abraham and M. S. Nair, "Computer-aided detection of COVID-19 from X-ray images using multi-CNN and Bayesnet classifier," *Biocybernetics and Biomedical Engineering*, Article vol. 40, no. 4, pp. 1436-1445, 2020, doi: 10.1016/j.bbe.2020.08.005.
- [14] P. D. Barua et al., "Automatic COVID-19 Detection Using Exemplar Hybrid Deep Features with X-ray Images," *International Journal of Environmental Research and Public Health*, vol. 18, no. 15, 2021, doi: 10.3390/ijerph18158052.
- [15] M. Turkoglu, "COVIDetectionNet: COVID-19 diagnosis system based on X-ray images using features selected from pre-learned deep features ensemble," *Applied Intelligence*, 2020/09/18 2020, doi: 10.1007/s10489-020-01888-w.
- [16] A. Reda, S. Barakat, and A. Rezk, "A Transfer Learning-Enabled Optimized Extreme Deep Learning Paradigm for Diagnosis of COVID-19," *Computers, Materials & Continua*, vol. 70, no. 1, 2022, doi: 10.32604/cmc.2022.019809.
- [17] M. E. H. Chowdhury 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.
- [18] F. Chollet and others, "Keras," 2015. [Online]. Available: <https://github.com/fchollet/keras>.

- [19] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.
- [20] E. Bisong, "Google Colaboratory," 2019, pp. 59-64.
- [21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436-44, 05/28 2015, doi: 10.1038/nature14539.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," presented at the Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Lake Tahoe, Nevada, 2012.
- [23] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, *Advances in deep learning*. Springer, 2020.
- [24] E. S. M. El-Kenawy, A. Ibrahim, S. Mirjalili, M. M. Eid, and S. E. Hussein, "Novel Feature Selection and Voting Classifier Algorithms for COVID-19 Classification in CT Images," *IEEE Access*, vol. 8, pp. 179317-179335, 2020, doi: 10.1109/ACCESS.2020.3028012.
- [25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.
- [26] E. D. Carvalho, R. R. V. Silva, F. H. D. Araújo, R. d. A. L. Rabelo, and A. O. de Carvalho Filho, "An approach to the classification of COVID-19 based on CT scans using convolutional features and genetic algorithms," (in eng), *Computers in biology and medicine*, vol. 136, pp. 104744-104744, 2021, doi: 10.1016/j.combiomed.2021.104744.
- [27] S. Sen, S. Saha, S. Chatterjee, S. Mirjalili, and R. Sarkar, "A bi-stage feature selection approach for COVID-19 prediction using chest CT images," *Applied Intelligence*, vol. 51, no. 12, pp. 8985-9000, 2021/12/01 2021, doi: 10.1007/s10489-021-02292-8.
- [28] M. Zabir, N. Fazira, Z. Ibrahim, and N. Sabri, "Evaluation of Pre-Trained Convolutional Neural Network Models for Object Recognition," *International Journal of Engineering and Technology(UAE)*, vol. 7, pp. 95-98, 08/13 2018, doi: 10.14419/ijet.v7i3.15.17509.
- [29] M. Toğaçar, B. Ergen, and Z. Cömert, "COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches," *Computers in Biology and Medicine*, vol. 121, p. 103805, 2020/06/01/ 2020, doi: 10.1016/j.combiomed.2020.103805.
- [30] H. Mukherjee, S. Ghosh, A. Dhar, S. M. Obaidullah, K. C. Santosh, and K. Roy, "Shallow Convolutional Neural Network for COVID-19 Outbreak Screening Using Chest X-rays," (in eng), *Cognit Comput*, pp. 1-14, 2021, doi: 10.1007/s12559-020-09775-9.
- [31] H. Panwar, P. K. Gupta, M. K. Siddiqui, R. Morales-Menendez, and V. Singh, "Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet," *Chaos, Solitons & Fractals*, vol. 138, p. 109944, 2020/09/01/ 2020, doi: 10.1016/j.chaos.2020.109944.
- [32] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv 1409.1556*, 09/04 2014.
- [33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
- [34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826.
- [35] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251-1258.
- [36] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, 2021/03/31 2021, doi: 10.1186/s40537-021-00444-8.
- [37] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.
- [38] M. Rahimzadeh and A. Attar, "A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2," *Informatics in Medicine Unlocked*, Article vol. 19, 2020, Art no. 100360, doi: 10.1016/j.imu.2020.100360.
- [39] R. Bandyopadhyay, A. Basu, E. Cuevas, and R. Sarkar, "Harris Hawks optimisation with Simulated Annealing as a deep feature selection method for screening of COVID-19 CT-scans," (in eng), *Applied soft computing*, vol. 111, pp. 107698-107698, 2021, doi: 10.1016/j.asoc.2021.107698.
- [40] A. Narin, "Accurate detection of COVID-19 using deep features based on X-Ray images and feature selection methods," *Computers in Biology and Medicine*, vol. 137, p. 104771, 2021/10/01/ 2021, doi: 10.1016/j.combiomed.2021.104771.
- [41] G. Ke *et al.*, "LightGBM: a highly efficient gradient boosting decision tree," presented at the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 2017.
- [42] V. Vapnik, "The Nature of Statistical Learning Theory," vol. 8, 2000, pp. 1-15.
- [43] M. Maia, J. S. Pimentel, I. S. Pereira, J. Gondim, M. E. Barreto, and A. Ara, "Convolutional Support Vector Models: Prediction of Coronavirus Disease Using Chest X-rays," *information*, vol. 11, no. 12, 2020, doi: 10.3390/info11120548.
- [44] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995/09/01 1995, doi: 10.1007/BF00994018.
- [45] T. Rahman, M. E. H. Chowdhury, and A. Khandakar, "COVID-19 Radiography Database." <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database> (accessed 2021/01/09).