A deep and handcrafted features-based framework for diagnosis of COVID-19 from chest x-ray images

Ferhat Bozkurt

Abstract
Automatic early diagnosis of COVID-19 with computer-aided tools is crucial for disease treatment and control. Radiology images of COVID-19 and other lung diseases like bacterial pneumonia, viral pneumonia have common features. Thus, this similarity makes it difficult for radiologists to detect COVID-19 cases. A reliable method for classifying non-COVID-19 and COVID-19 chest x-ray images could be useful to reduce triage process and diagnose. In this study, we develop an original framework (HANDEFU) that supports handcrafted, deep, and fusion-based feature extraction techniques for feature engineering. The user interactively builds any model by selecting feature extraction technique and classification method through the framework. Any feature extraction technique and model could then be added dynamically to the library of software at a later time upon request. The novelty of this study is that image preprocessing and diverse feature extraction and classification techniques are assembled under an original framework. In this study, this framework is utilized for diagnosing COVID-19 from chest x-ray images on an open-access dataset. All of the experimental results and performance evaluations on this dataset are performed with this software. In experimental studies, COVID-19 prediction is performed by 27 different models through software. The superior performance with accuracy of 99.36% is obtained by LBP+SVM model.

KEYWORDS
COVID-19, feature engineering, feature extraction, machine learning

1 | INTRODUCTION
COVID-19, which is emerging as an upper respiratory and lung infection, first appeared in Wuhan province, China in late 2019. It is mostly infected by airway and contact, and causing severe damage to the infected lungs. The virus has since spread rapidly and has become a global epidemic. The number of cases and related deaths continue to increase every day. One of the major steps to prevent the spread of the COVID-19 is to effectively scan people suspected of being infected to prevent the virus from spreading further across the population. In this context, any technological tool that allows fast and high-accuracy scanning of COVID-19 infection is useful to health professionals. Lung imaging with x-ray is one of the most widely used and accessible modality for rapid examination of lung conditions. Chest x-ray images are ready to be quickly analyzed by radiologists. Chest x-ray images have been proved to be useful in monitoring the effects COVID-19 causes on lung tissue. Therefore, chest x-ray images can be used to diagnose COVID-19. Lung x-ray radiography is an effective tool for triaging non-COVID-19 patients with pneumonia to efficiently allocate hospital resources. However, there are many common features between x-ray images of COVID-19 and pneumonia caused by other viral infections, like common influenza. These common observations in images are ground-glass opacities, air space consolidation, bronchovascular...
thickening, vascular thickening, and bronchial wall thickening. In this context, chest x-ray images of COVID-19 and pneumonia are similar and contain similar features. This similarity makes it difficult for radiologists to diagnose COVID-19 cases. A reliable method for classifying COVID-19 and non-COVID-19 chest x-ray images can accelerate the triage process of non-COVID-19 cases and maximize the allocation of hospital resources to COVID-19 cases.

Real-world data (like chest x-ray images) given to models in machine learning (ML) are in the form of feature vectors and these feature vectors are extracted from the raw data. Feature engineering is the process of extracting relevant features from data for an ML model. The count of features is as important as the features themselves. If the features in the feature vectors representing real-world data given to the model are not sufficient, the model will not be able to perform its main task. If there is a feature vector that contains more than necessary and unrelated features, the model will still not produce accurate results. In the ML process, not only the model but also the features that will represent the data are selected. Well-selected features facilitate subsequent modeling steps and increase the resulting model's ability to complete the desired task. However, if the features are not selected properly, a much more complex model may be required to achieve the same level of performance. Thus, we study on performances of feature extraction techniques and feature vectors of chest x-ray images in this study.

The scope and contributions of this study could be summarized as follows:

- In this study, a general framework (HANDEFU) that could be used in different computer vision and ML problems is developed.
- This original framework supports handcrafted, deep, and fusion-based feature extraction techniques for feature engineering.
- Any feature extraction technique and model could then be added dynamically to the library of software at a later time upon request.
- This framework is utilized for diagnosing COVID-19 from chest x-ray images.
- The user can build a feature extraction method and classification to train the model. Then, all performance evaluations on test data are performed with this software.

The rest of this article is organized as follows. In Section 2, the related works in literature are presented. In Section 3, the methodology which contains stages of the framework is explained. In Section 4, utilized dataset for testing is stated. Then, experimental results are presented and findings are discussed in Section 5. Finally, Section 6 concludes the article and presents some future work opportunities.

## 2 RELATED WORK

In the literature, there exist numerous approaches to diagnose COVID-19 from chest x-ray images based on various ML, deep learning, and hybrid methods. The studies based on machine and deep learning techniques are gaining increasing popularity for radiology images. When the related studies on this subject are examined, the most common and biggest problem is the lack of a sufficiently large data set. In general, studies have aimed to overcome this problem with various methods. Studies conducted on this subject can be given as follows: related studies using a similar open-access dataset with this study are summarized as follows. In some studies, different-sized image samples belonging to the classes may have been taken from the same data set at different times. Therefore, it can be observed that similar techniques give different performances in such a situation. For instance, Chowdhury et al. created this data set, made it available for open-access, and increased the size of the data over time. They studied the utility of artificial intelligence for accurate detection of COVID-19 from chest x-ray images rapidly. They obtained 99.7% classification accuracy with DenseNet201 model (used dataset contained 423 COVID-19, 1579 normal, and 1485 viral pneumonia chest x-ray images). Ahammed et al. used a CNN model to detect COVID-19 positive patients in early phase. Their model achieved 94.03% accuracy. The limitation of this study small number of data (285 COVID-19 images) was used for training and this is not adequate for deep-learning and COVID-19 prediction. Aslan et al. proposed a hybrid model with CNN-based transfer learning and BiLSTM. Their hybrid architecture gives 98.70% accuracy for COVID-19 infection detection on chest x-ray images on this similar data set. Gupta et al. proposed CVDNet that is deep CNN model to classify COVID-19 patients from other cases (normal or pneumonia) using chest x-ray images. They achieved an average accuracy of 97.20% for two-class (binary) and 96.69% for three-class (multiple) classifications. Similarly, Asif et al. proposed the detection COVID-19 pneumonia patients using deep CNN on same dataset (this consisted of 1345 viral pneumonia, 864 COVID-19, and 1341 normal images). They obtained the best performance with more than 98% classification accuracy. Cavallo et al.15 segmented manually lung areas by using polygonal regions of interest from this dataset. They extracted texture features with co-occurrence matrix by using gray level histogram. They extracted a total of 308 features from per ROI, and 110 COVID-19 chest x-ray images were selected for the final analysis. They obtained 91.8% accuracy with ensemble ML and 92.9% accuracy with artificial neural network. The common limitations of these studies are the small number of data and owning to the small number of COVID-19 samples. Therefore, the lack of images of the COVID-19 class does not make learning perfect and makes it difficult to detect COVID-19.

The rest of this article is organized as follows. In Section 2, the related works in literature are presented. In Section 3, the methodology which contains stages of the framework is explained. In Section 4, utilized dataset for testing is stated. Then, experimental results are presented and findings are discussed in Section 5. Finally, Section 6 concludes the article and presents some future work opportunities.
Other related studies could be summarized as follows. Rasheed et al.\textsuperscript{14} proposed the use of logistic regression and CNN for diagnosis of COVID-19. They investigated their method with PCA and without PCA to obtain high accuracy. They obtained overall accuracy between 95.2\% and 97.6\% without PCA, and between 97.6\% and 100\% with PCA for identification. In order to detect COVID-19 from chest x-ray images, Ahmed et al.\textsuperscript{15} proposed a COVID-Net which is combination of residual network and parallel convolution. They achieved accuracy of 97.99\%. Sharifrazia et al.\textsuperscript{16} proposed a fusion of CNN, SVM, and Sobel filter to detect COVID-19 using x-ray images. They used high pass filter with Sobel to get the edges of the images. The highest classification accuracy achieved with this method is 99.02\%. Shankar and Perumal\textsuperscript{17} proposed FM-HCF-DLF model that is a novel fusion method hand-crafted with deep learning features for COVID-19 classification and diagnosis. Their proposed model outcome superior performance with accuracy of 94.08\%. Joshi et al.\textsuperscript{18} proposed a deep learning-based system for chest x-ray images to detect the COVID-19. The average test accuracy of 97.11\% was achieved for multi-class and 99.81\% for binary classification. Jain et al.\textsuperscript{19} studied on deep learning-based CNN models for detection of COVID-19 on chest x-ray images. They compared the performance of Xception, Inception V3, and ResNeXt models about accuracy. When compared to other methods, Xception gives the highest accuracy with 97.97\%. To detect COVID-19 patients, Sarker et al.\textsuperscript{20} proposed a deep learning-based approach using the model of DenseNet-121. The test accuracy of 96.49\% was achieved for binary classification and 93.71\% for multi-class classification. Turkoglu\textsuperscript{21} proposed COVIDetectioNet model for diagnosis of COVID-19. This model utilizes features selected from combination of deep features. A transfer learning approach was used with a pretrained CNN-based AlexNet architecture and SVM was used for classification. In experimental results, an accuracy of 99.18\% was achieved by using this model. Since the data was imbalanced in some studies, the diversity of the data was increased with the image augmentation technique for the training model. For instance, Rahman et al.\textsuperscript{22} achieved 95.11\% classification accuracy with DenseNet201 model (used large dataset that contains 18,479 images with 3616 COVID-19, 6012 non-COVID, and 8851 normal images). They utilized from augmentation technique (an image rotation-based technique) to generate training images before CNN models for training. Similarly, Luz et al.\textsuperscript{23} utilized from this augmentation technique for effective training. They achieved 93.5\% classification accuracy with EfficientNet architecture (used large dataset that contains 13,770 images with 183 COVID-19, 5521 viral pneumonia, and 8066 normal images).

The common shortcoming of state-of-art studies is that most deep learning-based models are trained on unbalanced datasets and therefore it may lack robustness. There exists a small number of studies on creating a scalable deep learning model using image enhancement/preprocessing techniques and efficient feature extraction. Due to unbalanced data processing and inability to extract necessary features from images, the classification accuracy does not reach the desired level. Also, proposed models in literature cannot guarantee to reproduce the promising results when these are evaluated on a larger dataset. It is a handicap that running a CNN architecture with many iterations on a small dataset leads to overfitting. In order to overcome these limitations, this framework combines image preprocessing with handcrafting, deep and fusion based feature extraction techniques to create a scalable model.

### 3 | METHODOLOGY

In this study, a general framework (HANDEFU) including different feature extraction techniques and classifiers is composed for use in computer vision and machine-learning problems as shown in Figure 1. This original framework supports handcrafted, deep, and fusion based feature extraction techniques. The handcrafted based feature extraction methods as LBP, HOG, GABOR are present in the framework. The user is able to build shallow neural network, deep neural network models, and popular deep-learning techniques through the HANDEFU framework. The framework has a dynamic structure. Therefore, other popular deep learning techniques could be added to the library for deep-feature extraction. At present, the framework contains popular deep learning techniques as shallow-fully connected network model, deep-fully connected network model, shallow-convolutional network model, deep-convolutional network model, AlexNet, VGG16, VGG19, ResNet50, InceptionV3, Xception, and DenseNet121 model. Moreover, it is aimed to obtain high performance by creating a robust model by combining handcrafted features and deep learning features. The user can select and combine different deep and handcrafted feature extraction methods as InceptionV3+LBP, VGG19+GLCM, and VGG19+HOG through the framework. It is necessary to classify the handcrafted or deep learning features obtained in the previous stage with the help of a classifier. In this way, a supervised learning or pretrained model is built for prediction. Thus, predictions can be made from test data with an established model. At this stage, the user is given the opportunity to classify the features with other supervised learning techniques through the framework. The classifier options as kNN, Naive Bayes, neural network (simple 3-layer NN), MLP-Classifier, and softmax classifier exist in the framework. These options can be further increased by dynamically adding them to the framework library.

As shown in Figure 1, HANDEFU framework is utilized for diagnosis of COVID-19 from chest x-ray images. The user loads the dataset as training and test data. Then, the user is given the opportunity for preprocessing. Then, the user builds a feature extraction method and classification to train the model through the framework. Furthermore, all performance evaluations on test data are performed with this software. All of the stages will be explained in detail as follow.
3.1 Preprocessing

Images may need to be preprocessed before the feature extraction. The purpose of the preprocessing at this stage is to facilitate the computing to be done in the later stages of the framework. At this stage, the user is given the opportunity to choose a number of preprocessing options such as smoothing, noise reduction, normalization, and image resizing. The image could be enhanced in different ways. For instance, smoothing reduces undesired details or noise in the image and speeds up following processing. When taken as a single image, some artifacts may cause an inhomogeneity problem in the image. Normalization is the process in which multiple images are placed in an average statistical distribution in terms of size and pixel values. A single image can also be normalized within itself. Thus noisy images are eliminated by intensity normalization. Brightness and contrast can be adjusted through a linear transformation of image intensities with dynamically windowing of each image.

Figure 2 shows a sample of chest X-ray image before and after preprocessing. Gaussian smoothing and median filtering significantly reduce the noise of the image. The two-dimensional (2D) Gaussian filter is often preferred for smoothing and noise reduction. Regions with a diagnosis of COVID-19 are pointed by a radiologist as shown in Figure 2. The radiologist shows the changes made after pre-processing. Thanks to these filters, ground-glass opacities and consolidations become clearer. Also, some reticular patterns which are related to COVID-19 are sharpened after preprocessing as shown in Figure 2. The equation of 2D Gaussian filter is given in Equation (1). $\sigma^2$ denotes variance of the distribution and $(x, y)$ is Cartesian coordinates of the image.\(^{24}\)

$$GF_{2D}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left(x^2 + y^2 / 2\sigma^2\right)}$$

The sizes of the images in the database used for image classification and pattern recognition may differ from each other. In some feature extraction functions, the number of extracted features depends on the size of the image (e.g., LBP, HOG, and Gabor). In order to conduct...
methods, the number of features obtained from the images must be the same. For this, all images must be resized to a fixed size. In addition, when the size and resolution of the images are reduced by the resizing process, the time required for processing the images in the framework decreases and the performance of the framework is positively affected. In this study, the images are resized from 1024×1024 to 250×250.

### 3.2 Hand-crafted feature extraction

The fundamental elements of the pattern recognition process are preprocessing, feature extraction, feature selection, training, and classification as shown in Figure 3. The feature can be basically explained as measurable or observable information about the pattern. Feature extraction enables obtaining characteristic features of the pattern by eliminating irrelevant and excessive information. This is basically a kind of dimensional reduction process. Since very high dimensional data are used especially in image processing applications, using these data in their original form greatly increases the processing time. Thanks to hand-crafted feature extraction methods, it is ensured that the original data is reduced to smaller sizes by eliminating unnecessary and high-size information and preserving its characteristic features.

Feature extraction, which is one of the pattern recognition stages, plays an important role in the success of the system. Appropriately selected features represent the characteristics of the classes and positively affect recognition success. The purpose of feature extraction is to transport as much of the class information as possible to smaller sizes. In other words, the purpose of feature extraction is to find a \( T \) transform that expresses the \( X \) space in the best possible way from a \( k \)-dimensional \( X \) observation space to a smaller \( l \)-dimensional \( Y \) feature space as given in Equation (2). Here, numerical features can be in the form of a set of vectors that characterize the class of the pattern.

\[
X [k \times k] \rightarrow Y [l \times l] \quad k > l
\]  

\[(2)\]

**Figure 3** An overview of handcrafted-based feature extraction and classification
Figure 3 shows an overview of handcrafted based feature extraction and classification. In this stage, a number of handcrafted based feature extraction methods in HANDEFU framework (e.g., LBP, HOG, Gabor) will be explained in detail.

### 3.2.1 Local binary pattern (LBP)

LBP feature extraction is used in applications such as texture classification, face recognition, facial expression recognition, gender, and age identification (classification).\(^{26}\) LBP is a pattern recognition method based on binary coding between each pixel in the image and its neighbors. LBP generates binary codes by comparing the 3×3 neighborhoods of each pixel in the gray image. The 3×3 image obtained from each pixel is considered as circular, and the central pixel is accepted as the threshold value. When the neighbor pixel value is compared with the center pixel value, it is encoded in binary as 1 if the value of the neighbor pixel is greater than or equal to the value of the center pixel, else encoded as 0. In this way, the LBP code of the center pixel is obtained by generating an 8-bit binary code and converting it to a decimal number. This process is computed for each pixel of the image as shown in Figure 4. Places in an image (such as corners, edges, line regions, luminous, or unilluminated regions) represent the features of the image. These textural features of the image could be extracted with the LBP code. In this way, it is possible to obtain different textural features of the images with the generated LBP code.

The LBP compares the values between the center pixel at \((x, y)\) position and the neighboring pixels surrounding it. LBP code is obtained when an 8-bit code is created and converted to a decimal number. For each pixel in the image, the binary code is generated by thresholding the neighboring pixels according to the center pixel. The generation of LBP codes is performed by Equation (3). In Equation (3) \((x, y)\) is the coordinates of the center pixel, \(g_i\) is the gray value of the center pixel, and \(g_n\) is the values of the pixels in 3×3 neighborhood surrounding the center pixel. The \(f(k)\) function in Equation (4) is the 8 neighborhood values of gray pixels.\(^{27,29}\)

\[
LBP(x, y) = \sum_{n=0}^{n-1} f(g_n - g_i)2^n
\]

\[
f(k) = \begin{cases} 
1 & \text{if } k \geq 0 \\
0 & \text{if } k < 0 
\end{cases}
\]

Regional histograms could be used to get more efficient results. For this purpose, the x-ray image is divided into 8×8 = 64 square regions and histograms are obtained by calculating the LBP codes separately for each region.\(^{27}\) These histograms are concatenated and used in classification as feature vectors.

### 3.2.2 Histogram of oriented gradient (HOG)

In recent years, the use of edge orientation histograms within the scope of image recognition studies has gained great momentum. Local gradient orientation histograms (HOG) was first suggested by Dalal and Triggs.\(^{30}\) HOG is defined as the characteristics of the orientation (\(\theta\)) and magnitude
The pixel intensity at the point \((x, y)\). \(^{31}\)

\[
f_x(x, y) = l(x + 1, y) - l(x - 1, y)
\]

\[
f_y(x, y) = l(x, y + 1) - l(x, y - 1)
\]

Gradient magnitude \((m)\) and gradient orientations \((\theta)\) are calculated by using horizontal \(f_x\) and vertical \(f_y\) image gradients as follows. \(^{31}\)

\[
m(x, y) = \sqrt{f_x(x, y)^2 + f_y(x, y)^2}
\]

\[
\theta(x, y) = \tan^{-1} \left( \frac{f_y(x, y)}{f_x(x, y)} \right)
\]

\(\theta(x, y)\) can be divided into equal regions between 0°–180° or 0°–360° depending on the application when computing the histogram. Gradient direction regions are defined as 9 regions with an angle difference of 20° between 0° and 180° in this step. The gradient magnitude of each pixel in the cell is distributed to the histogram regions according to the angle. Thus, a big size histogram is obtained by combining all the histograms created in one block as shown in Figure 5. This histogram is then normalized using by Equation (9). In this equation, \(V_k\) is the large histogram vector obtained from a block and \(v\) is the normalized HOG feature vector. L2-Norm is used for normalization. \(^{31}\) In this stage, a computed HOG feature vector is ready to be given as input of a classifier.

\[
v = \frac{V_k}{\sqrt{\|V_k\|^2 + 1}}
\]

### 3.2.3 Gabor feature extraction

This technique shows great similarity with the human visual system in terms of its frequency and orientation characteristics. Gabor wavelets are used in computer vision applications and modeling biological visions (particularly applications about texture descriptor as fingerprint, palm, and face recognition). \(^{32}\) In wavelet transform, spatial information is obtained as well as the frequency information of the image. In the wavelet transform
method, frequency data could be regionalized. For this reason, it performs the transformation of the function not only according to the frequency parameters but also according to the scale parameters when compared with the Fourier transform. This technique, which is especially preferred in texture analysis, uses Gaussian elements modulated by sine and cosine functions. Gabor filter could be applied dimensionally to 1D or 2D images. A Gabor filter is obtained by modulating a sine wave with a Gaussian function. The 2D Gabor filter kernel matrix is calculated by Equation (10). In this equation, the first exponential expression (Gaussian function) is a descending term depending on $x'$ and $y'$, while the second exponential expression is a periodic term containing a complex number.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right) \exp\left(i\left(\frac{2\pi x'}{\lambda} + \psi\right)\right)$$

In the equation, $\lambda$ is the coefficient that determines the wavelength of the cosine factor. If the coefficient is 1, since the cosine expression $(\cos(2\pi x')) = 1$ will always be 1, the coefficient should be chosen 2 or a larger integer. $\theta$ is used to calculate the values of $x'$ and $y'$ and is the orientation angle of the Gabor kernel to be created. The variables $x'$ and $y'$ are calculated by the following formula for a given value of $\theta$.

$$x' = x \cos \theta + y \sin \theta$$
$$y' = -x \sin \theta + y \cos \theta$$

$\psi$, is the phase angle of the kernel matrix whose angle is to be formed. The filter can be shifted on the $x$-axis by changing this value. $\sigma$ is the coefficient that determines the standard deviation (SD) of the Gaussian function. $\gamma$ allows the SD value to be determined for $y'$. If this value is 1, the resulting kernel matrix will be of equal length since it has equal SD for $x$ and $y$. When a different ratio is chosen, the kernel matrix will be formed in a rectangular shape. The application of the generated Gabor filter to an image is a 2D convolution process and is performed with Equation (12). In the equation, $I(p, q)$ refers to the image to be applied Gabor filter and $g(x - p, y - q; \lambda, \theta, \psi, \sigma, \gamma)$ denotes the Gabor filter.

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \int \int I(p, q) \ g(x - p, y - q; \lambda, \theta, \psi, \sigma, \gamma) \ dpdq$$

In this study, Gabor feature extraction is performed by applying 40 Gabor filters obtained from 5 scales and 8 orientations as shown in Figure 6. For example, the size of the chest x-ray images used in experiments is 256x256 pixels. A feature vector is with length $(m \times n \times u \times v)/(d_1 \times d_2)$, $d_1 = 4$ and $d_2 = 4$ is the factor of downsampling along rows and columns, $(u = 5)$ is the number of scales and $(v = 8)$ is the number of orientations. By using 40 Gabor filters, the dimension of the feature vector for one image is $(256 \times 256 \times 8 \times 5)/(4 \times 4) = 163,840$.

### 3.3 Deep feature extraction

In order to define a model with classical ML techniques or to set up a detection (prediction) system, first the feature vector should be extracted. Experts in the field are required for the extraction of the feature vector. These processes both take much time and keep the expert very busy. For this reason, these techniques cannot process raw data without pre-processing and expert assistance (as supervised learning). Deep learning (DL) has made great progress by overcoming this problem that those working in the field of ML have been dealing with for many years. Deep learning is a field of ML research that has emerged from studies on artificial neural networks in recent years. Because deep networks do the learning process on raw data, unlike traditional ML and image processing techniques. While processing the raw data, it acquires the required information with the representations which are obtained in different layers.
The DL algorithm is scalable with large data sets and is effective in identifying complex patterns from feature-rich data sets. The main fields that deep learning is successfully utilized are computer vision, speech recognition, and natural language processing. The reason for the widespread success of deep learning is its method of directly predicting output from the representation set. An important advantage of deep learning over traditional techniques is that it does not require an explicit feature extraction stage. The network takes the raw input and maps it to the desired output. Features are learned automatically by the network without manual intervention. Although these models require high processing power, this is no longer seen as a problem with the development of central and graphical processing power today. DL models are the improvement of traditional networks with an increasing number of hidden layers. Hidden layers greatly improve the performance of these networks. Deep learning techniques can automatically reveal image features that are not obtained from the original images. There is no human intervention or manual determination of the features in the extraction of the features. Deep learning has recently been widely used in pattern recognition and image classification.

It is possible to build different DL models thanks to the development of big data and GPUs. These designed models perform the learning process themselves without the features determined by the user from the input data. It achieves this learning process by discovering different features of the data in different layers. The basic model of these architectures is considered to be CNN (convolutional neural network). CNNs are the improved versions of artificial neural network ANNs. The deep neural network as a result of further increasing the number of hidden layers in ANNs can be defined as CNN. This depth in the CNN was achieved by using 2D filters. In addition to this difference in depth, CNNs perform the learning process in a hierarchical structure. As shown in the figure, defining an object in CNN is possible by discovering a sub-feature of that object in each layer.

CNN as shown in Figure 7, is a mathematical structure consisting of three types of layers, typically convolution, pooling, and fully-connected layers. While the convolution and pooling layers perform feature extraction, it achieves the final output for classifying the extracted features using the fully-connected layer.

In digital images, pixel values are stored in a 2D matrix, this is a small convolution matrix called a feature extractor mask that is applied to each point of the image. When one layer feeds its output to the next layer, the extracted features can become hierarchically and increasingly complex. At this stage, with the masks used, learning processes take place in the layers and the values of the layers are computed.

**Convolutional layer:** The convolution process, which forms the basis of CNNs, aims to apply a filter matrix to the input by the convolution process and use the results for the next layer. This layer is based on traversing small size filters such as $2 \times 2$, $3 \times 3$, and $5 \times 5$ over the entire image. Thus, a new image is obtained by extracting the more distinctive features in the image. The weights of the filter matrix used for the convolution process are determined at the learning stage of the CNN. The filter matrix is shifted with the specified stride and the convolution operation is applied. The result obtained is given as the input for the next layer, if not the last layer. If it is the last layer, it represents the output image. Filter coefficients ($f$) are multiplied by equal-sized windows ($w$) in the image and calculated by taking their sums as in Equation (13). As a result, a new image is obtained based on distinctive high-level features.

$$w(x, y) * f(x, y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} w(i, j)f(x+i, y+j)$$

**Pooling layer:** Pooling layer is used to reduce the computational load by reducing the parameters within the network. There are many types of pooling layers. Maximum, average, and L2 pooling options are available in the HANDEFU framework. In maximum pooling, the highest of the pixel values remaining in the filter window, and in the average pooling process, the average of all the pixel values within the filter window is kept as a single value in the output pixel. At the end of the pooling process, the aspect ratio of the image is reduced. In this process, there are losses in pixel values, but these losses create less computational load for subsequent layers.

![Figure 7](image-url) Fundamental architecture of the CNN
Dropout layer: The dropout layer is the most commonly used regularization layer in deep learning methods. Dropout is the removal of node from input and hidden layers according to certain rules. Thus, it is aimed to reduce the parameter and computation without loss of performance and it is aimed to prevent overfitting. Dropout layer is generally used after fully connected layers.\textsuperscript{37} Thus, there is less information link between nodes and the nodes are least affected by each other’s weight changes. As a result, more consistent network models are created.

Batch normalization: Batch normalization is a method used to make the CNN more regular, like the dropout layer. Besides the regulation of the neural network, it also allows the network to resist the vanishing gradient during training. This can reduce training time and provide better performance. For the batch normalization computation, the input is defined as $\{x_{i, n}\}$ and the training parameters are shown as $\gamma, \beta$. The operation in Equation (14) is the average of the mini batch. The average is calculated by summing all the input in the mini-batch and dividing them by the total number of input. Equation (15) is variance of the mini-batch. As a result of these calculations, the general intervals in which the data in the mini-batch are computed. Normalization of the data is calculated with Equation (16). And, normalized data is processed with $\gamma$ (scaling parameter) and $\beta$ (shifting parameter) in Equation (17). As a result of these operations, a learnable linear equation is obtained. $\gamma$ and $\beta$ are used as hyperparameters of the designed network.\textsuperscript{5} These parameters determine how much the batch normalization layer will do on the data.

\begin{align}
\mu_{\beta} &= \frac{1}{m} \sum_{i=1}^{m} x_{i} \\
\sigma_{\beta}^2 &= \frac{1}{m} \sum_{i=1}^{m} (x_{i} - \mu_{\beta})^2 \tag{14} \\
\tilde{x}_{i} &= \frac{x_{i} - \mu_{\beta}}{\sqrt{\sigma_{\beta}^2 + \epsilon}} \tag{15} \\
y_{i} &= \gamma \tilde{x}_{i} + \beta = BN_{\gamma, \beta}(x_{i}) \tag{16}
\end{align}

Fully connected layer: The fully connected layer is commonly located at the end of the CNN and is used in classification processes. This layer is often used towards the end of the CNN architecture and to optimize class scores. In addition, the number of these layers may vary in architectures based on deep learning.

Classification layer: Classification layer is the last layer of the CNN model that classification process is performed. The output values of this layer are equal to the number of classes, depending on the number of objects to be recognized. Activation functions are used to learn and predict any continuous and complex relationship between network variables. In this framework, the ReLU activation function exists to avoid linearity in the network. Since multi-class classification is needed in the fully connected layer in the CNN architecture, softmax function is used as in Equation (18). Softmax classifier is commonly used in this layer based on deep learning architectures. This classifier produces probabilistic values between 0 and 1 for each class.\textsuperscript{5} As a result, the highest probability value gives the class predicted by the model.

\begin{equation}
\text{softmax}(z_{j}) = \frac{e^{z_{j}}}{\sum_{k=1}^{K} e^{z_{k}}}; \quad j = 1, \ldots, K \tag{18}
\end{equation}

As shown in Figure 8, the user is able to build a sample deep CNN model consisting of four set convolution-pooling layers, a flattened layer, three fully connected layers through the HANDEFU framework. Moreover, the framework contains popular deep learning techniques as shallow-wholly connected network model, deep-fully connected network model, shallow-convolutional network model, deep-convolutional network model, AlexNet, VGG16, VGG19, ResNet50, InceptionV3, Xception, and DenseNet121 model as shown in Figure 9. Thus the user is able to build shallow neural network and deep neural network models through the HANDEFU framework. The framework has a dynamic structure. Therefore, other popular deep learning techniques could be added to the library later.

### 3.4 Fusion-based feature extraction

Ensemble learning is the creation of a new model by using multiple modeler algorithms together instead of using a single basic learning model. At this stage, it is aimed to obtain high performance by creating a robust model by combining handcrafted features and deep learning features. Feature fusion is achieved by combining multiple feature vectors together.\textsuperscript{37} In this context, previously obtained features are combined into a single vector. Here, a feature fusion is obtained by integrating the maximum feature vector. The fusion of features with entropy determines the projection of the method. For instance, deep feature extraction is done by InceptionV3 (Equation 19) and handcrafted feature extraction is done by LBP (Equation 20). Then, feature extraction is combined as a single feature vector as in Equation (21). In order to obtain better features depends on score, entropy is
In the classification stage, it is necessary to classify the handcrafted or deep learning features obtained in the previous stage with the help of a classifier. In this way, a supervised learning or pretrained model is built for prediction. Thus, prediction can be made from test data with an established model. At this stage, the user is given the opportunity to classify the features with other supervised learning techniques through framework. The classifier options as k-nearest neighbors (kNN), Naive Bayes (NB), and neural network (NN), MLPClaassifier exist in the framework. These options can be further increased by dynamically adding them to the framework library. At this stage, for instance, the user can classify and predict with SVM instead of softmax after extracting features with deep learning (e.g., ResNet50 + SVM). Similarly, the user can choose any classifier after extracting the handcrafted feature. Therefore, user can build different models such as LBP+SVM, LBP+kNN, LBP+NB, LBP+NN, HOG+SVM, HOG+kNN, HOD+NB, HOD+NN through the framework.
4 | EXPERIMENTAL RESULTS

4.1 | Data set

In this study, an open-access dataset is utilized for classification of COVID-19 chest x-ray images. This dataset consists of chest x-ray images for COVID-19 positive cases along with normal and viral pneumonia images. A team of researchers composes this database from the university Qatar, Doha, Bangladesh, Dhaka, and Qatar with their collaborators and doctors. As shown in Table 1, a total of 3886 chest x-ray images exist as 1200 COVID-19 (positive cases), 1345 viral pneumonia, and 1341 normal (negative cases) images. Figure 10 shows image samples of normal (negative), viral pneumonia, and COVID-19 from this dataset. Regions with a diagnosis of viral pneumonia and COVID-19 are pointed by a radiologist as shown in Figure 10. In experimental results, this dataset is utilized for training and testing through the framework. All predictions and classification of COVID-19 chest x-ray images are performed on this dataset.

4.2 | Results

The experimental studies in this study are conducted on a desktop computer with Intel (R) Core-i7 8700U CPU @ 3.20 GHz, 4GB NVIDIA GeForce GTX 1050 Ti graphics card and 16 GB primary memory. The software was implemented using Python 3.7 and its related libraries (e.g., Tensorflow, Keras, Numpy, Pandas, SciKit Learn, Matplotlib, and Tkinter GUI).

In this section, the performance evaluations of the methods are performed over four performance criteria (metrics) such as Precision (PR), Recall (RE), F1-Score (F1-SC), and Accuracy (ACC). These values are calculated over confusion matrices for each class. Confusion matrices are obtained for each model and experiment as shown in Figure 11. In this figure; 0, 1, and 2 for row and column name represents (0=COVID-19), (1=viral pneumonia) and (2=normal) class respectively. These performance metrics are calculated from confusion matrix and mathematically defined as follows.

\[
PR = \frac{TP}{TP + FP} \tag{22}
\]

\[
RE = \frac{TP}{TP + FN} \tag{23}
\]

| Table 1 | Categorical distribution of the x-ray images in dataset |
|---------|----------------------------------------------------------|
| Category name | Number of images |
| Normal       | 1341           |
| Viral pneumonia | 1345         |
| COVID-19     | 1200           |
| Total:       | 3886           |

Figure 10 shows sample chest x-ray images from dataset as the case of (A) normal, (B) viral pneumonia, and (C) COVID-19.
In these equations, True Positive (TP) means that the patient actual with COVID-19 positive case and system recognized as positive. True Negative (TN) means that the patient actually without COVID-19 case and the system recognized as the negative. False Positive (FP) means that the patient actually without COVID-19 case and the system recognized as the positive. False Negative (FN) means that the patient actual with COVID-19 case and system recognized as the negative.

In this section, performance comparisons of different feature extraction methods and models are performed through the HANDEFU framework as shown in Table 2. Performance comparisons of different feature extraction methods and models (top 5 models are marked in bold according to Acc) are given in Table 2. The user is able to split data for training and testing as 70% is reserved for training and 20% for testing or 80% is for training 20% is for testing in advance. In this section, we performed the performance evaluations for 80%–20%. The methods are categorized as hand-crafted, deep, and fusion-based feature extraction. For instance, LBP, HOG, and Gabor-based texture descriptors are used as hand-crafted feature extraction techniques with kNN, NB, NN, and SVM classifiers. In this context, 12 different models exist such as LBP+SVM, HOG+SVM, Gabor+SVM and so forth. The user can select any model and run the model on loaded dataset as train and test data through the framework. Similarly, the user is able to build shallow neural network and deep neural network models through the HANDEFU framework.

The framework has a dynamic structure. Therefore, other popular deep learning techniques could be added to the framework later. Nowadays, the most popular deep-learning techniques exist in frameworks as AlexNet, VGG16, VGG19, ResNet50, InceptionV3, Xception, and DenseNet121 as shown in Table 2. Hereby, performance comparisons of different deep feature extraction models are performed with obtained performance metrics. Furthermore, In fusion-based category, user can select and combine different deep and handcrafted feature extraction methods as InceptionV3+LBP, VGG19+GLCM, and VGG19+HOG through the framework. All of the performance evaluation results of different models on this dataset are given in Table 2. Thus, we can observe the superior models thanks to comparisons of their performances. The top 5 models among different feature extraction methods and their performance comparisons are presented in Figure 12. Various feature extraction methods and models in this framework are built and executed on this dataset. Each of the model’s performance and execution times are recorded. Performance evaluations and execution times of all models are recorded with this software. The top 10 models and their performance comparisons about accuracy and execution time are studied as shown in Table 3. Furthermore, the user can obtain training/test accuracy and training/test loss graphs according to iterations as shown in Figure 13. The graphs of each model are saved by this software.

\[
F1 - SC = \frac{2 \times TP}{2 \times TP + FN + FP} 
\]

\[
ACC = \frac{TP + TN}{TP + FN + FP + TN} 
\]
| Method                  | Model                  | PR (%) | RE (%) | F1-SC (%) | ACC (%) |
|------------------------|------------------------|--------|--------|-----------|---------|
| Hand-crafted feature   | LBP+kNN                | 92.94  | 89.95  | 94.41     | 94.76   |
| extraction             | LBP+NB                 | 88.13  | 100    | 95.68     | 95.80   |
|                        | LBP+NN                 | 94.70  | 99.20  | 96.88     | 98.04   |
|                        | LBP+SVM                | 98.52  | 99.42  | 98.97     | 99.36   |
|                        | HOG+kNN                | 75.24  | 96.81  | 84.66     | 89.04   |
|                        | HOG+NB                 | 85.58  | 96.53  | 90.70     | 93.96   |
|                        | HOG+NN                 | 91.70  | 96.81  | 94.19     | 96.14   |
|                        | HOG+SVM                | 92.67  | 98.46  | 95.47     | 97.23   |
|                        | Gabor+kNN              | 88.06  | 98.33  | 92.91     | 95.37   |
|                        | Gabor+NB               | 88.55  | 99.56  | 92.02     | 94.98   |
|                        | Gabor+NN               | 90.64  | 98.58  | 94.43     | 96.43   |
|                        | Gabor+SVM              | 93.22  | 98.82  | 95.94     | 97.49   |
| Deep feature extraction| Shallow FC Neural Net.  | 90.72  | 90.82  | 90.67     | 90.62   |
|                        | Deep FC neural net     | 92.03  | 92.20  | 92.08     | 92.03   |
|                        | Shallow CNN            | 94.85  | 94.78  | 94.78     | 94.73   |
|                        | Deep CNN               | 96.12  | 96.02  | 96.07     | 96.02   |
|                        | AlexNet                | 95.78  | 95.88  | 95.81     | 95.76   |
|                        | VGG16                  | 95.90  | 95.90  | 95.84     | 95.76   |
|                        | VGG19                  | 95.67  | 95.62  | 95.59     | 95.50   |
|                        | ResNet50               | 93.50  | 93.50  | 93.41     | 93.32   |
|                        | ResNet50+SVM           | 94.82  | 94.75  | 94.75     | 94.70   |
|                        | InceptionV3            | 93.70  | 93.88  | 93.75     | 93.70   |
|                        | Xception               | 95.88  | 95.57  | 95.58     | 95.50   |
|                        | DenseNet121            | 97.31  | 97.23  | 97.26     | 97.17   |
| Fusion-based feature   | InceptionV3+LBP+MLP    | 96.11  | 96.65  | 96.36     | 96.53   |
| extraction             | VGG19+GLCM+MLP         | 95.96  | 95.96  | 95.90     | 95.85   |
|                        | VGG19+HOG+MLP          | 94.17  | 97.04  | 95.53     | 96.37   |
### Table 3: Top 10 models and their performance comparison about accuracy and execution time

| Model           | Accuracy (%) | Execution time (t) |
|-----------------|--------------|--------------------|
| LBP+SVM         | 99.36        | 5 min 13 s         |
| LBP+NN          | 98.04        | 5 min 7 s          |
| Gabor+SVM       | 97.49        | 50 min 1 s         |
| HOG+SVM         | 97.23        | 6 min 33 s         |
| DenseNet121     | 97.17        | 2 h 57 min 43 s    |
| InceptionV3+LBP | 96.53        | 1 h 52 min 04 s    |
| VGG19+HOG       | 96.37        | 6h 29 min 16 s     |
| HOG+NN          | 96.14        | 11 min 27 s        |
| Deep CNN        | 96.02        | 7 h 52 min 52 s    |
| AlexNet         | 95.76        | 1h 59 min 27 s     |

**Figure 13** Training/test accuracy and training/test loss graphs of the DenseNet121 model

## Discussion

In experimental studies, it has been observed that LBP-based hand-crafted feature extraction method is more successful in diagnosing COVID-19 cases for this dataset. As shown in Table 1, LBP+SVM model gives the highest accuracy value with 99.36%. Besides, this model has the shortest execution time (approximately 5 min) according to other methods. Because, regional histograms in LBP give more efficient results. X-ray image is divided into $8 \times 8 = 64$ square regions and histograms are obtained by calculating the LBP codes separately for each region. These histograms are concatenated and then they have given to the classifier as feature vector. A 3776 size feature vector is obtained for each image in the data set. Similarly, HOG which is another hand-crafted feature extraction method is studied in experimental results. It is seen from the Table 1 that HOG+SVM model gives better performance with accuracy rate of 97.23%. The model gives very good performance according to accuracy and execution time. The execution time of the model is approximately recorded as 6 minutes as shown in Table 3. The feature vector of any image is obtained as 34,596 with HOG method for 256×256 size of image. Gabor which is the other hand-crafted feature extraction method, is studied in experimental results. In this study, Gabor feature extraction is performed by applying 40 Gabor filters obtained from 5 scales and 8 orientations. By using 40 Gabor filters, the dimension of the feature vector for 256×256 size of one image is 163,840. Gabor+SVM model gives better performance with accuracy rate of 97.49% as shown in Table 2. Since the dimension of feature vector obtained with the Gabor model is very large, the execution time of this method takes long time according to other hand-crafted methods.

In experimental studies, deep feature extraction method and different models are studied on this dataset. Thus, first the simple artificial neural network (shallow-fully connected network) model was established. Then deep CNNs, which are popular today, are used in COVID-19 diagnosing. As shown in Table 2, different popular pre-trained learning techniques such as Deep CNN, AlexNet, VGG16, VGG19, ResNet50, InceptionV3, Xception, and DenseNet121 have been compared on this dataset. Different models are run on the same data, and the performance of each is recorded.
as shown in Table 2. In the shallow-fully connected network model, which is the simplest neural network model according to other methods, four hidden layers and 128, 256, 128, and 64 neurons exist in each layer respectively. In the experimental study, the lowest accuracy value was obtained from this model with 90.62%. More hidden layers have been added to the network using convolution neural networks to further improve accuracy performance. For instance, the Deep CNN model consists of four sets of convolution pooling layers, a flattened layer, three fully connected layers, and a classifier. The model has 39,361,539 parameters. In the experimental study, accuracy value of 96.02% is obtained with the Deep CNN model. The DenseNet121 model, which is popular today and is one of the advanced versions of CNN, achieved the highest accuracy value with 97.17%. It was observed that the DenseNet121 model in this study gives superior results than other deep learning models. The InceptionV3+LBP model, which is fusion-based technique ensemble deep and hand-crafted features, achieved the highest accuracy value with 96.53%.

In Figure 13, training/test accuracy and training/test loss graphs for 50 iterations of the DenseNet121 are presented. All of the models’ evaluation results, confusion matrices, training/test accuracy, and training/test loss graphs could be taken from this software. Both training and test accuracy curves show a rising curve as the number of iterations increases. The loss curve shows the decrease in error rate. It shows that the training process and learning of the network are at a good learning rate. As shown in Figure 13, while the loss value decreases in each iteration, the accuracy rate increases with the given training set and learning occurs.

Performance comparison of the current study with other studies is given as shown in Table 4. In this table, the state-of-art/relevant studies with data of various sizes on the chest x-ray dataset are discussed. The accuracy results of the existing studies in the literature are examined as

| Study                  | Year | Model                  | Chest x-ray dataset                                                                 | Accuracy          |
|------------------------|------|------------------------|-------------------------------------------------------------------------------------|-------------------|
| Ahmmed et al.⁹         | 2020 | CNN                    | 2971 images (285 COVID-19, 1345 viral pneumonia, and 1341 normal)                   | Multiple: 94.03%  |
| Apostolopoulos et al.⁴⁰| 2020 | VGG19                  | 1428 images (224 COVID-19, 700 viral pneumonia, and 504 normal)                    | Binary: 98.75%    | Multiple: 93.48% |
| Chowdhury et al.⁵      | 2020 | SqueezeNet             | 487 images (423 COVID-19, 1485 viral pneumonia, and 1579 normal)                   | Multiple: 98.3%   |
| Duran-Lopez et al.⁴¹   | 2020 | COVID-XNet             | 6926 images (2589 COVID-19 and 4337 normal)                                        | Binary: 94.43%   |
| Elaziz et al.⁴²        | 2020 | MRFODE+kNN             | 1560 images (219 COVID-19 and 1341 non-COVID-19 images)                             | Binary: 98.09%   |
| Khan et al.⁴³          | 2020 | CoroNet                | 251 images (284 COVID-19, 330 pneumonia bacterial, 327 Pneumonia viral, and 310 normal) | Binary: 89.60%   |
| Ouchicha et al.⁵       | 2020 | CVDNet                 | 2905 images (219 COVID-19, 1345 Viral pneumonia, and 1341 normal)                  | Binary: 97.20%    | Multiple: 96.69% |
| Ozturk et al.⁴⁴        | 2020 | DarkCovidNet           | 1127 images (127 COVID-19, 500 viral pneumonia, and 500 normal)                    | Binary: 98.08%    | Multiple: 87.02% |
| Toraman et al.⁴⁵       | 2020 | Convolutional CapsNet  | 2331 images (231 COVID-19, 1050 viral pneumonia, and 1050 normal)                  | Binary: 97.24%    | Multiple: 84.22% |
| Ahmad et al.⁴⁶         | 2021 | Ensemble deep learning | 4200 images (1050 COVID-19, 1050 bacterial pneumonia, 1050 viral pneumonia, and 1050 normal) | Multiple: 96.49%  |
| Alam et al.⁴⁷          | 2021 | CNN+HOG+VGG19          | 5090 images (1979 COVID-19 and 3111 normal)                                        | Binary: 99.49%   |
| Aslan et al.¹⁰         | 2021 | CNN-BiLSTM             | 2905 images (219 COVID-19, 1345 viral pneumonia, and 1341 normal)                  | Multiple: 98.7%   |
| Bhardwaj and Kaur⁴⁸    | 2021 | Ensemble deep learning | 10,000 images (2161 COVID-19, 2022 viral pneumonia, and 5563 normal)               | Binary: 98.33%    | Multiple: 92.36% |
| Gupta et al.¹¹         | 2021 | InstaCovNet-19         | 3047 images (361 COVID-19, 1345 viral pneumonia, and 1341 normal)                  | Binary: 99.52%    | Multiple: 99.08% |
| Jain et al.¹⁹          | 2021 | Xception               | 6432 images (576 COVID-19, 4273 viral pneumonia, and 1583 normal)                  | Multiple: 97.97%  |
| Luz et al.²³           | 2021 | EfficientNet-B3        | 13,770 images (183 COVID-19, 5521 viral pneumonia, and 8066 normal)                | Multiple: 93.5%   |
| Rahman et al.²²        | 2021 | DenseNet201            | 18,479 images (3616 COVID-19, 6012 non-COVID, and 8851 normal)                     | Multiple: 95.11%  |
| Current study          | 2021 | LBP+SVM                | 3886 images (1200 COVID-19, 1345 viral pneumonia, and 1341 normal)                  | Multiple: 99.36%  |
6 | CONCLUSION

From past to present, infectious diseases are one of the most important threats to human health. COVID-19 is an infectious disease caused by severe acute respiratory syndrome coronavirus. The rapid spread of the disease and the increase in mortality rates in many countries reveal that an effective treatment method should be developed. At this point, artificial intelligence and ML can contribute to this problem. Especially, there is a serious shortage of experts and traditional pneumonia and COVID-19 have great similarities. ML aided diagnosing method could be an important milestone towards significantly reducing test time. This technique offers both a lower cost and a more accurate diagnosis and treatment for COVID-19 and similar diseases. Lung x-ray radiography is an effective tool for triaging non-COVID-19 patients with pneumonia to efficiently allocate hospital resources. However, there are many common features between x-ray images of COVID-19 and pneumonia caused by other viral infections, like common influenza. This similarity makes it difficult for radiologists to diagnose COVID-19 cases.

Feature engineering is the process of extracting relevant features from data for a ML model. The count of features is as important as the features themselves. Well-selected features facilitate subsequent modeling steps and increase the resulting model’s ability to complete the desired task (e.g., detection, classification). Thus, we study on performances of feature extraction techniques and feature vectors of chest x-ray images. In this study, we develop an original framework (HANDEFU) that supports handcrafted, deep, and fusion-based feature extraction techniques for feature engineering. The novelty of this study is that image preprocessing and diverse feature extraction and classification techniques are assembled under an original framework. This framework could be used in different computer vision and ML problems. The framework has a dynamic structure. The user is able to build any model with handcrafted, deep, and fusion-based feature extraction techniques and various classifiers through the HANDEFU framework. At present, the framework contains popular pretrained and deep learning techniques. Any feature extraction technique and model could then be added dynamically to the library of software at a later time upon request. Thus, the user can build a feature extraction method
and classification to train the model. All performance evaluations on test data are performed with this software. This framework is utilized for diagnosing COVID-19 from chest x-ray images. In experimental studies, COVID-19 prediction is performed by 27 different models through software. The superior performance with accuracy of 99.36% is obtained by LBP+SVM model. This model has the shortest execution time according to other methods. Because regional histograms in LBP give more efficient results for feature extraction.

In further studies, more handcrafted features-based and deep learning-based techniques, and more data will be studied and comparisons will be made. Furthermore, by using this framework in the background, it is planned to develop a mobile and web-based system aimed at supporting health professionals in their efforts to detect cases of COVID-19 and other diseases.

**DATA AVAILABILITY STATEMENT**

The data that support the findings of this study are openly available in [COVID-19 Chest x-ray Database] at [https://www.kaggle.com/tawsifurrahman/covid19-radiography-database], reference number.7

**ORCID**

Ferhat Bozkurt DOI: https://orcid.org/0000-0003-0088-5825

**REFERENCES**

1. WHO. Pneumonia of unknown cause–China. emergencies preparedness, response. disease outbreak news. World Health Organization (WHO); 2020.

2. Jiao Z, Choi JW, Halsey K, et al. Prognostication of patients with COVID-19 using artificial intelligence based on chest x-rays and clinical data: a retrospective study. *Lancet Digit Health*. 2021.

3. Shi H, Han X, Jiang N, et al. Radiological findings from 81 patients with COVID-19 pneumonia in Wuhan, China: a descriptive study. *Lancet Infect Dis*. 2020;20(4):425-434.

4. Khuzani AZ, Heidari M, Shariati SA. COVID-Classifier: An automated machine learning model to assist in the diagnosis of COVID-19 infection in chest x-ray images. *medRxiv*. 2020.

5. Ouchicha C, Ammor O, Meknassi M. CVDNet: a novel deep learning architecture for detection of coronavirus (Covid-19) from chest x-ray images. *Chaos Soliton Fract*. 2020;140:109245.

6. Zheng A, Casari A. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O’Reilly Media, Inc; 2018.

7. Rahman T, Chowdhury M, Khandakar A. COVID-19 chest X-ray database. Accessed September 15, 2021. [https://www.kaggle.com/tawsifurrahman/covid19-radiography-database], reference number.7

8. Rahman T, Chowdhury M, Khandakar A, et al. Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access*. 2020;8:132665-132676.

9. Ahmed S, Hossain MF, Noor MBT. Convid-Net: an enhanced convolutional neural network framework for COVID-19 detection from chest x-ray images. *Soft Comput*. 2021;98:106912.

10.  Asilf S, Wenhu Y, Jin H, Tao Y, Jinhai S. Classification of covid-19 from chest x-ray images using deep convolutional neural networks. *MedRxiv*. 2020.

11.  Cavallo A, Troisi J, Forcina M, et al. Texture analysis in the evaluation of Covid-19 pneumonia in chest X-ray images: a proof of concept study. *Current Med Imaging*. 2021.

12.  Rasheed J, Hameed AA, Djidji C, Jamil A, Al-Turjman F. A machine learning-based framework for diagnosis of COVID-19 from chest X-ray images. *Interdiscip Sci Comput Life Sci*. 2021;13(1):103-117.

13.  Ahmed S, Hossain MF, Noor MBT. *Convid-Net*: an enhanced convolutional neural network framework for COVID-19 detection from X-ray images. *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*, 2021:671-681; *Springer*, New York, NY.

14.  Sharifrazi D, Alizadehsani R, Roshanzamir M, et al. Fusion of convolution neural network, support vector machine and Sobel filter for accurate detection of COVID-19 patients using X-ray images. *Biomed Signal Process Control*. 2021;102622.

15.  Shankar K, Perumal E. A novel hand-crafted with deep learning features based fusion model for COVID-19 diagnosis and classification using chest x-ray images. *Complex Intell Syst*. 2020;1-17.

16.  Joshi RC, Yadav S, Pathak VK, et al. A deep learning-based COVID-19 automatic diagnostic framework using chest X-ray images. *Biocybern Biomed Eng*. 2021;41(1):239-254.

17.  Jain R, Gupta M, Tanuja S, Hemanth DJ. Deep learning based detection and analysis of COVID-19 on chest X-ray images. *Appl Intell*. 2021;51(3):1690-1700.

18.  Sarker L, Islam MM, Hannan T, Ahmed Z. Covid-densenet: a deep learning architecture to detect covid-19 from chest radiology images; 2020.

19.  Turkoglu M. *COVIDetectioNet*: COVID-19 diagnosis system based on X-ray images using features selected from pre-learned deep features ensemble. *Appl Intell*. 2021;51(3):1213-1226.

20.  Rahman T, Khandakar A, Qiblawey Y, et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Comput Biol Med*. 2021;132:104319.

21.  Luz E, Silva P, Silva R, et al. Towards an effective and efficient deep learning model for COVID-19 patterns detection in X-ray images. *Res Biomed Eng*. 2021;1:1-14.

22.  Bozkurt F, Yaganoglu M, Gunay FB. Effective Gaussian blurring process on graphics processing unit with CUDA. *Int J Mach Learn Comput*. 2015;5(1):57.

23.  Nanni L, Ghidoni S, Brahnam S. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recogn*. 2017;71:158-172.

24.  Bozkurt F, Yaganoglu M, Guven FB. Effective Gaussian blurring process on graphics processing unit with CUDA. *Int J Mach Learn Comput*. 2015;5(1):57.

25.  Nanni L, Ghidoni S, Brahnam S. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recogn*. 2017;71:158-172.

26.  Khaleefah SH, Mostafa SA, Mustapha A, Samsudin NA, Nasrudin MF, Baz A. A survey on local binary pattern and gabor filter as texture descriptors of smart profiling systems. *Indones J Electr Eng Comput Sci*. 2020;20(3):1379-1387.
27. Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell. 2002;24(7):971-987.
28. Li X, Niu H. Feature extraction based on deep-convolutional neural network for face recognition. Concurr Pract Exper. 2020;32(22):1-1.
29. Hosny KM, Darwish MM, Li K, Salah A. COVID-19 diagnosis from CT scans and chest X-ray images using low-cost raspberry Pi. PLoS One. 2021;16(5):e0250688.
30. Dalal N, Triggs B. Histograms of oriented gradients for human detection. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05); Vol. 1. 2005:886-893; IEEE.
31. Kadota R, Sugano H, Hiromoto M, Miyamoto R, Nakamura Y. Hardware architecture for HOG feature extraction. Proceedings of the 2009 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing; 2009:1330-1333; IEEE.
32. Deng HB, Jin LW, Zhen LX, Huang JC. A new facial expression recognition method based on local Gabor filter bank and PCA plus LDA. Int J Inf Technol. 2005;11(11):86-96.
33. Haifez SF, Selim MM, Zayed HH. 2D face recognition system based on selected Gabor filters and linear discriminant analysis LDA; 2015. arXiv preprint arXiv:1503.03741.
34. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436-444.
35. O'Shea K, Nash R. An introduction to convolutional neural networks; 2015. arXiv preprint arXiv:1511.08458.
36. Meunier LCV, Chandy DA. Design of convolution neural network for facial emotion recognition. Proceedings of the 2019 2nd International Conference on Signal Processing and Communication (ICSPC); 2019:376-379; IEEE.
37. Umer M, Ashraf I, Ullah S, Mehmoon A, Choi GS. COVINet: a convolutional neural network approach for predicting COVID-19 from chest X-ray images. J Ambient Intell Humaniz Comput. 2021;1-13.
38. Bozkurt F, Köse C, Sarı A. A texture-based 3D region growing approach for segmentation of ICA through the skull base in CTA. Multimed Tools Appl. 2020;79(43):33253-33278.
39. Yañganolu M. Real time wearable speech recognition system for deaf persons. Comput Electr Eng. 2021;91:107026.
40. Apostolopoulos ID, Mpesiana TA. Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med. 2020;43(2):635-640.
41. Duran-Lopez L, Domínguez-Morales JP, Corral-Jaime J, Vicente-Díaz S, Linares-Barranco A. COVID-XNet: a custom deep learning system to diagnose and locate COVID-19 in chest X-ray images. Appl Sci. 2020;10(16):5683.
42. Elaziz MA, Hosny KM, Salah A, Darwish MM, Lu S, Sahlol AT. New machine learning method for image-based diagnosis of COVID-19. PLoS One. 2020;15(6):e0235187.
43. Khan AI, Shah JL, Bhat MM. CoroNet: a deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. Comput Methods Prog Biomed. 2020;196:105581.
44. Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med. 2020;121:103792.
45. Toraman S, Alakus TB, Turkolgu I. Convolutional capsnet: a novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks. Chaos Soliton Fract. 2020;140:110122.
46. Ahmad F, Farooq A, Ghani MU. Deep ensemble model for classification of novel coronavirus in chest X-ray images. Comput Intell Neurosci. 2021:2021.
47. Alam NA, Ahsan MM, Based MA, Haider J, Kowalski M. COVID-19 detection from chest X-ray images using feature fusion and deep learning. Sensors. 2021;21(4).
48. Bhardwaj P, Kaur A. A novel and efficient deep learning approach for COVID-19 detection using X-ray imaging modality. Int J Imaging Syst Technol. 2021.

How to cite this article: Bozkurt F. A deep and handcrafted features-based framework for diagnosis of COVID-19 from chest x-ray images. Concurrency Computat Pract Exper. 2022;34(5):e6725. doi: 10.1002/cpe.6725