AIMS Mathematics

AIMS Mathematics, 9(8): 10539–10571.
DOI: 10.3934/math.2024998
Received: 22 February 2024
Revised: 02 May 2024
Accepted: 20 May 2024
Published: 25 June 2024

https://www.aimspress.com/journal/Math

Review

Deep convolutional neural network (CNN) model optimization techniques—Review for medical imaging

Ghazanfar Latif[1,*], Jaafar Alghazo[2], Majid Ali Khan[1], Ghassen Ben Brahim[1], Khaled Fawagreh[1] and Nazeeruddin Mohammad[1]

[1] Department of Computer Science, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia
[2] Department of Software Engineering and Information Technology Management, University of Minnesota Crookston, Crookston, MN 56716, USA

* **Correspondence:** Email: glatif@pmu.edu.sa; Tel: +966-501-057-422.

Abstract: The field of artificial intelligence (AI) and machine learning (ML) has been expanding and is explored by researchers in various fields. In medical diagnosis, for instance, the field of AI/ML is being explored because if medical diagnostic devices are built and designed with a backend of AI/ML, then the benefits would be unprecedented. Automated diagnostic tools would result in reduced health care costs, diagnosis without human intervention, overcoming human errors, and providing adequate and affordable medical care to a wider portion of the population with portions of the actual cost. One domain where AI/ML can make an immediate impact is medical imaging diagnosis (MID), namely the classification of medical images, where researchers have applied optimization techniques aiming to improve image classification accuracy. In this paper, we provide the research community with a comprehensive review of the most relevant studies to date on the use of deep CNN architecture optimization techniques for MID. As a case study, the application of these techniques to COVID-19 medical images were made. The impacts of the related variables, including datasets and AI/ML techniques, were investigated in detail. Additionally, the significant shortcomings and challenges of the techniques were touched upon. We concluded our work by affirming that the application of AI/ML techniques for MID will continue for many years to come, and the performance of the AI/ML classification techniques will continue to increase.

Keywords: convolutional neural network; CNN architecture optimization; deep learning; medical imaging; AI model optimization; medical diagnosis
Mathematics Subject Classification: 62M45, 68T45

1. Introduction

Artificial intelligence has played a significant role in nearly every aspect we encounter in our day-to-day lives, including healthcare, communications, manufacturing, transportation, and robotics, to name a few. It has been widely applied in numerous ways and it is growing rapidly to give more promising results and possibilities. As the world is working on recovering from the coronavirus pandemic by mostly working on a more effective set of vaccines as well as techniques for fast disease diagnosis, AI was on the front lines playing a prominent role in combatting the ill effects of this deadly contagion. AI plays a crucial role in this situation due to its important applications in the areas of medical image processing, data analysis, text processing, etc. Since the advent of this pandemic, a lot of active research has been performed by AI researchers on a war footing for early detection of this disease so that preventive measures can be taken and precious human lives can be saved. In general, the early and reliable diagnosis of any medical disease can help save many lives [1–3].

Deep learning (DL) has been proven to be very effective in processing large amounts of data, extracting features, and eventually making decisions. With the help of DL techniques, many applications related to medical imaging can be applied [4,5]. Medical imaging is the visual representation of human organs, which will be very helpful in the early detection and diagnosis of diseases. Medical images are known to generate a large amount of data, and experts may need to analyze every image to make an accurate diagnosis for the patient. Given the large number of medical images that need to be analyzed, it can be time-consuming for the medical staff to study each image and give a correct diagnosis. When a large amount of medical imaging data is given to a DL machine, the machine can assist experts in making decisions and learning about features in an image without human intervention [6]. Sometimes, it can be a challenge for experts to distinguish similar patterns associated with a disease, and it can lead to a false diagnosis. However, with advanced technology, these patterns can be discriminated against easily when the machine extracts the features on an image, learn the patterns, and classify them. As a result, the machine can then make predictions with high accuracy, leading to an accurate diagnosis. Convolutional neural network (CNN) is one of the many examples of DL techniques that can achieve high accuracy on classification and image recognition problems. The CNN-based model would need to be provided with a large dataset consisting of labeled medical images. This dataset is usually split into two parts, where the first is used for training and the second is used for testing. An advantage of such a model is its speed in processing large data and making decisions way beyond human ability. In the medical field, the speed of analysis and, eventually, early prediction is a very crucial factor in saving patients' lives.

DL can play a significant role in many medical imaging applications. One of the many examples of the use of DL in medical imaging is COVID-19 detection from X-ray and CT scans. With DL techniques, COVID-19 can be detected at its earliest stage to save a patient's life. Furthermore, DL can be used to detect the patterns of the lungs, and it can give a prediction of the criticality of COVID-19 patients [7,8]. Without using DL techniques, using a device that captures the visualization of the human body, such as the costly MRI, requires a large amount of computation power, and it needs to process millions of voxels. However, with DL techniques, the algorithm can train on a huge dataset and can predict the disease at a much faster speed and, better yet, with a low probability of being prone to errors. DL can help experts extract features and create new ones [9]. DL works well with many data types and image types. For example, it has been proven successful in the identification of skin lesions using dermoscopic images and the identification of human gastrointestinal tract abnormalities using endoscopic images [10,11]. DL in medical imaging is so promising to the medical sector in terms of its performance and speed. Using AI, a patient's life can be saved by making the correct diagnosis, and accordingly, patients are given the proper medication on time, which will increase the chances of overcoming the disease.

Similarly, researchers have been using this data to learn predictive models using ML and DL approaches, which aid in classifying the images of the lungs of the subjects as either coronavirus-infected or not. As we already have state-of-the-art DL classification algorithms like VGG, Inception, etc., AI researchers were able to apply these algorithms to existing datasets and further fine-tune the existing algorithms to improve their performance in terms of classification accuracy [12,13]. The results were promising in many cases, and custom-crafted algorithms such as CheXNet and COVNet also yielded better results along with those existing algorithms such as ResNet50 [14–16]. Statistical analysis, in addition to the ML experiments, was also conducted on these datasets, but the results were not as promising as AI approaches [17]. Algorithms like DarkCovidNet resulted in better performance in terms of classification accuracy, as was the case when using YOLO, which is a real-time object detection algorithm [18]. Such performance improvement is a good indication that, when integrated with an X-ray imaging machine output system, these algorithms would detect the Coronavirus disease with more than 95% accuracy while the X-ray images are being projected. The advantage of such an approach is that it saves time by avoiding the waiting time for the X-ray images to be available for training the model and then using the model for the classification task. The machine learning community has made remarkable progress in a very short span of time for a quite reliable and early diagnosis of diseases to initiate preventive steps to avoid severe health issues potentially leading to death.

2. Materials and methods

CNN is a class of neural networks that is designed using Multilayer Perceptron. Figure 1 shows a basic CNN architecture. The architecture consists of a number of layers, including convolution, pooling, dropout, and dense layers. The convolutional layer identifies key distinguishable features in the input images. The pooling layer is used to select only the salient features from the features identified by the convolutional layer. This also helps reduce the spatial size of the representation. The dropout layer is used to reduce overfitting, and finally, the dense layer is used to learn the model based on input features from previous layers. Such networks can be used to solve various image-processing problems related to classification and segmentation. Developed in the 1980s, CNNs were used for pattern recognition [19]. Hubel and Wiesel [20] proposed that the sensitivity of small cells in the visual cortex system generates a receptive field with edge-like patterns that are proportional to the position of the images. The neurons use the respective field of the image to extract the fundamental features, which are coupled in the subsequent layers to extract the complex features of the image. LeCun et al. [21] proposed the LeNet in 1998 to classify numbers in an application for handwritten cheques to be used in banks. The proposed LeNet model consists of five layers in addition to the input and output layers. The inputs of the model are 32×32 grayscale images and used the Sigmoid activation function [22]. In addition, LeNet utilized regularization to account for overfitting.

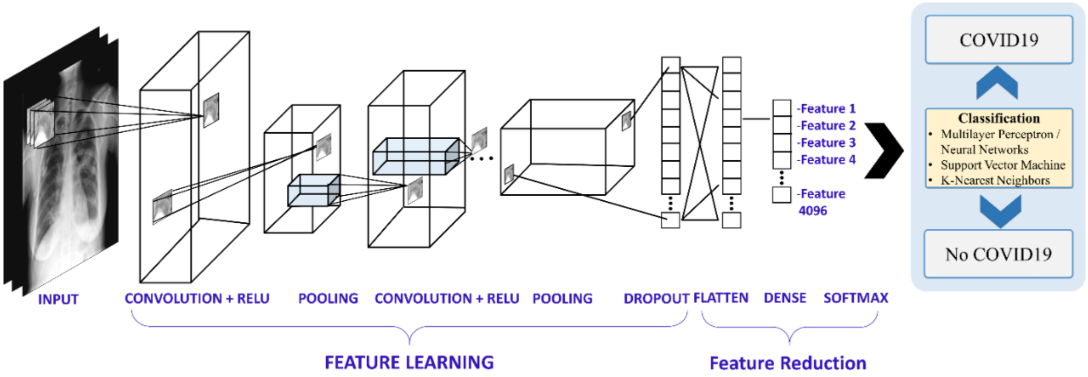

Figure 1. A basic convolutional neural network architecture.

Krizhevsky et al. [23] proposed a modified version of the LeNet named AlexNet, which was a deeper network. The AlexNet contained stacked convolutional layers with more filters per layer. The inputs to the AlexNet are 227×227 RGB images. AlexNet utilized Dropout to account for overfitting and utilized the ReLU activation function. Krizhevsky's network was able to surpass other networks in terms of performance in the ImageNet2012 challenge. The system used 1.3 million images for training purposes to segregate 1000 image classes. Google proposed a further modification to LeNet, which was named GoogleNet [24]. To drastically reduce the parameters, GoogleNet utilized small convolutions. They reduced the parameters to 4 million from the 60 million parameters in AlexNet. The inputs to GoogleNet are 224×224 RGB images. After ImageNet2012, CNN has been used in various image processing fields for classification and pattern recognition. CNN can be directly given raw images instead of feature vectors as input [25]. The principal requirement in designing CNN is establishing the architecture that can help develop a learning algorithm that minimizes the number of parameters and the computation complexity of the system [26,27]. The steps involved in creating the architecture for CNN are a local connection, weight sharing, pooling of data, and multilayer implementation [28]. The layers' implementation of CNN consists of convolutional layers, pooling layers, and fully connected dense layers [29]. The convolution layer extracts many low-level features from the input images. The convolution layer has equal weight and bias that make the images shift-invariant. The image's neighboring features, like the orientation of corners, edges, or endpoints, can be obtained by the neurons using the respective local field. Lecun et al. [21] and Cires et al. [30] utilize the complex features in the hidden layers for image classification. The sparse connectivity between the intermittent layers and the weight of the parameters between the neighboring pixels are used to develop a system that can classify images effectively. The convolutional layer utilizes filters (also called kernels) to extract distinct features from the input image. This process generates k feature maps that are sized $M - N + 1$, as illustrated in Figure 2. Afterward, each feature map undergoes down-sampling (also known as pooling) using either mean or max pooling over a $q \times q$ region. Typically, q ranges between 2 to 5 for larger inputs, as shown in Figure 3. The convolutional and pooling stages are then succeeded by a series of fully connected layers.

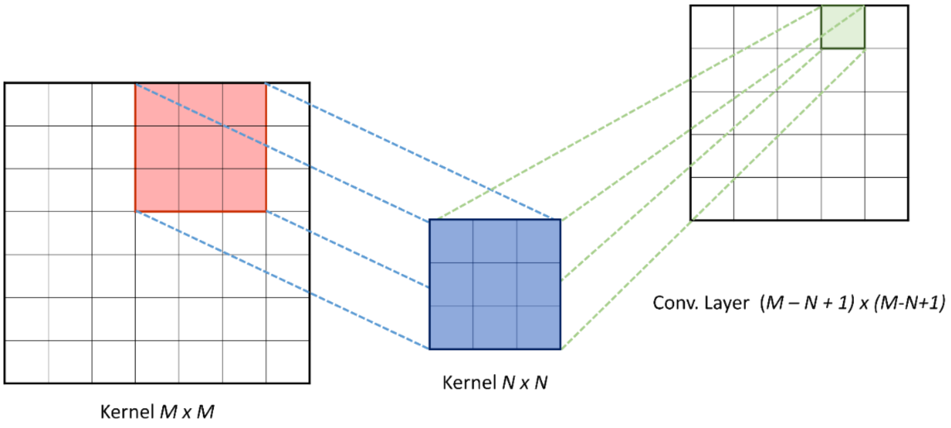

Figure 2. Mapping of the convolutional kernel filter to the input image.

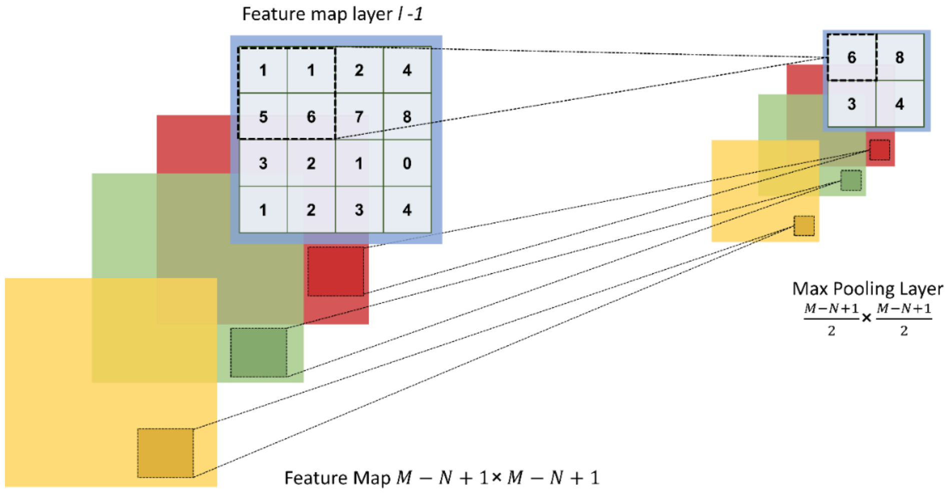
Feature map layer l -1
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
6 8
3 4
Max Pooling Layer
$\frac{M-N+1}{2} \times \frac{M-N+1}{2}$
Feature Map $M - N + 1 \times M - N + 1$

Figure 3. Sample of the max pooled layer mapping to the feature map.

Studies have shown that the automation of hyperparameters improves the accuracy of computer vision problems, such as face verification, object recognition, and pattern recognition [31–33]. The time complexity of the learning process can be substantial when dealing with large datasets that deal with a vast number of hyperparameters. The learning time in the large data set can be optimized using a graphics processing unit (GPU), but the hyperparameters optimization plays a vital role in the system's time complexity. The optimization leads to higher overall accuracies.

2.1. Approximation, depth, width, and invariance properties

The approximation of temporary functions of the input data in the design of CNN architecture plays an important role. This approximation is connected with the selection of parameters of the network like depth and width. The CNN architecture with numerous hidden layers can better approximate the properties of the input data than networks with a small number of hidden layers. A scattering network is a type of deep-learning network in which the values of convolutional filters are calculated by complex wavelet families [34]. This makes the scattering networks more stable and invariant to input data variations. This helps uncover the fundamental property of geometry and balance of the input data that undermines the generalization performance of CNN.

2.2. Generalization and regularization properties

An essential property of DL networks is their capability to generalize the system from a short input training dataset. From a statistical learning theory, it has been proven that the number of training datasets grows in a polynomial manner relative to the size of the network to achieve good generalization [35]. In practical implementations of deep networks, the system is trained with a small dataset compared to the number of parameters and hidden layers. Overfitting in such cases can be avoided using regularization techniques [36]. A dropout image regularization is one of the possible methods to prevent overfitting the data by the random selection of constant subset parameters at each repetition.

3. Deep CNN architecture optimization techniques

Various techniques have been considered in designing effective CNN-based models. These include early stopping, training with more data, regularization, cross-validation, and dropout. Under these models, convolutional layer is followed by ReLU activation function. Furthermore, downsampling is applied in the pooling layers, and a cross-entropy loss function is used to calculate the system's accuracy. The mini-batch gradient descent method using the Adam optimizer algorithm is employed for backpropagation. Table 1 summarizes the CNN architecture parameters used to optimize the CNN models for imaging. The table also describes the details of their effects on performance and causes.

3.1. Data augmentation

The role of data augmentation is important to improve the performance of the algorithm and also to tackle the problem of smaller and imbalanced datasets in DL-based models. In data augmentation, the size of the training dataset is increased, without increasing computational complexity, by adding transformations of the training data. In image classification and visual imagery applications, the most commonly used techniques for data augmentation include color jittering, cropping, flipping the image vertically or horizontally, rotation and scaling, etc. Krizhevsky et al. [23], proposed a method to change the RGB color channel intensity on training ALEXNET by PCA, which captures other notable image properties as well. Bengio et al. [44] proposed that data augmentation techniques are more beneficial for deep architecture as compared to shallow networks. He et al. implemented data augmentation along with regularization techniques such as dropouts and weight decay [45].

Transfer learning approach using well-known CNN models (GoogleNet, AlexNet, VGG16, VGG19, DenseNet, etc.) along with data augmentation techniques can be used to accelerate the training and testing process while yielding good results and performance. Figure 4 shows the utilization of preprocessing techniques along with the well-known CNN models for COVID-19 and Lungs Pneumonia detection using transfer learning.

Table 1. Summary of Deep CNN Architecture Improvement Parameters along with the details of their effects on performance and causes.

Parameter	Definition	Recommendations Values	Effect on Performance	Cause of the Effect
Batch Size [37]	The number of samples that will be propagated through the network.	A suitable batch size for medical image classification is between 16 to 64.	Batch size has an effect on accuracy, time to convergence, and regularization.	Smaller batch sizes can result in faster convergence and can also have a significant regularization effect but might not be able to reach optimum minima. A lower learning rate should be used for a smaller batch size for better accuracy.
Activation Function [38]	Defines the output of the node for the given input. This output is used as input to the next layer nodes.	ReLU is the commonly used as activation function for hidden layers, SoftMax (multiclass) and Sigmoid (binary class) are used for output layer.	Activation functions provide Non–linearity, Differentiability and Range Monotonic.	ReLU and SoftMax functions achieve excellent experimental performance.
Pooling Layer Size [39]	Minimizes the number of parameters required to describe the layers in DL.	2×2 Max pooling layer size is most commonly used, which will reduce the number of features to one-fourth.	Max pooling picks up the most salient features and minimizes the number of trainable parameters.	Max pooling is a discretization process based on the sample, causing less spatial information and fewer parameters.
Dropout Layer [40]	Neurons are “dropped out” randomly during training.	The dropout percentage can vary from 20% to 50%.	Increases performance while preventing overfitting.	The dropped-out neurons cause other neurons to handle the model representation, increasing network generalization.
Dense Layer [41]	Linear operation in which every input is connected to every output by weight, thus reducing the output feature size.	Generally, three levels of dense layers.	These layers perform the bulk of model learning based on features from previous layers that are available.	ReLU, Sigmoid, and SoftMax are the most common activation functions that can be used for dense layers.

Continued on next page

Parameter	Definition	Recommendations Values	Effect on Performance	Cause of the Effect
Kernel Size [42]	The kernel is the convolutional filter.	Generally, a 3 × 3 filter size is sufficient.	The filter size determines the details of the detected features.	In medical imaging, a model is generally looking for features at a low level, so a smaller filter size is more appropriate.
Number of Hidden Layers [43]	The layers between the input and the output layers.	Generally, 8 to 11 hidden layers can provide good results.	An upper bound exists for the number of hidden layers that can be defined without causing overfitting.	Computation time increases with an increase in the number of hidden layers as well as the possibility of overfitting.

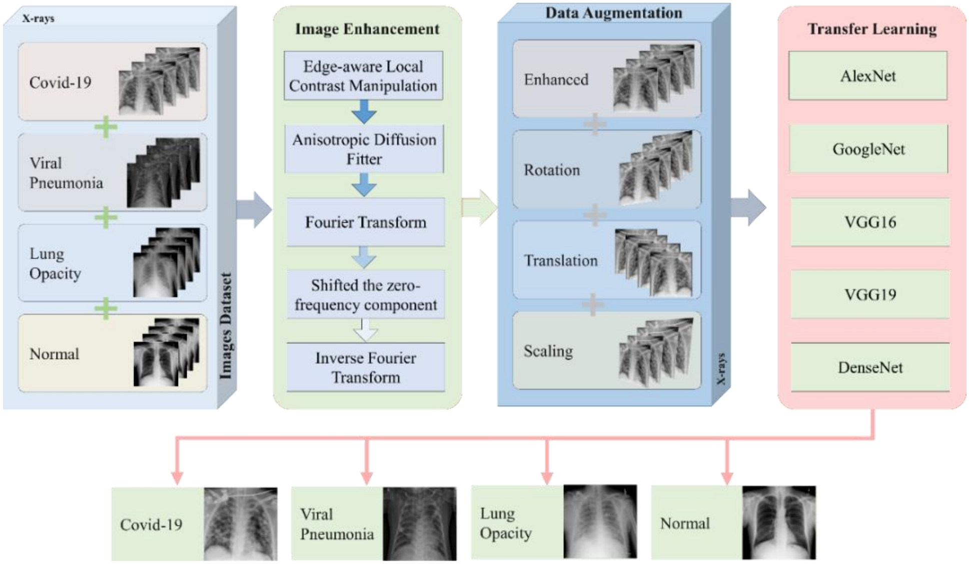

Figure 4. Utilization of preprocessing techniques along with the well-known CNN models for the COVID-19 and Lungs Pneumonia detection using transfer learning.

3.2. L1 and L2 regularization

L_1 and L_2 are the most commonly used regularization techniques and are used in multiple applications. For the parameters in L_1 regularization, the sum of absolute value is reduced by adding a regularization term to the objective function. In L_2 regularization, the addition in the objective function is done to minimize the sum of the squares of parameters. The vectors for parameters in L_1 regularization also known as Lasso regularization encourages sparsity in the model parameters pushing coefficient of less important features to zero, thus performing feature selection. For this reason, it will be implemented in the feature space domain. In the feature space setting, many features will be ignored. A modified version of the L_2 regularization method is most commonly used in machine learning techniques in which L_2 constraints are squared and then imposed on the weights. This method is named Tikhonov regularization or weight decay. The reason for labeling this method as weight decay is because the overall effect of weight reduction by factor is related to the gradient descent magnitude at every iteration [46]. The cost function in the standardized form is given in Eq (1).

$$\delta = arg\ min_{\delta} \frac{1}{R} \sum_{i=1}^{R} (L(\hat{y}_i,\ y_i) + \rho N(w)). \tag{1}$$

Equation (1) defines the total cost function that a learning model aims to minimize. It combines the loss (how well the model fits the data) with a penalty on the size of the model parameters (to avoid overfitting). In Eq (1), $N(w)$ is the regularization factor, which is defined in Eq (2).

$$N_{L2}(w) \triangleq ||W||_2^2. \tag{2}$$

In L_1 regularization, the magnitude of absolute weights is penalized. The regularization factor is shown in Eq (3).

$$N_{L1}(w) \triangleq \sum_{i=1}^{q} ||W||_1. \tag{3}$$

The L_1 regularization factor is not differentiable at zero. To avoid this problem, a small constant is added, which will make the factor close to zero. To solve the non-convex L_1 regularization problem, first-order differentiation is applied to weights, which is a common practice in neural network problems [47]. The L_1 norm with an approximation is shown in Eq (4).

$$||W||_1 = \sum_{i=1}^{q} \sqrt{w_i^2 + \vartheta}. \tag{4}$$

Elastic net regularization is another method defined in [48], where a mixture of L_1 and L_2 regularization terms are considered. L1 regularization, as part of this function, specifically aims to make some of the model's weights zero, thus encouraging a simpler, sparser model that is less likely to overfit.

3.3. Dropout

If neurons are dropped temporarily during their connections in neural networks, it is called Dropout. With the help of random dropouts, many architectures of the network can be combined efficiently, hence, reducing the overfitting problem. The probability of dropping the neurons is $1 - p$, and it will also reduce the effect of co-adaptability among neurons. Usually, hidden units are implemented at such stages with a neuron dropout probability of 0.5. For 2^n neurons, the dropout sample average is computed by the full working network. Here, each node weight will be factored by p because the larger value of n will increase the computations. Besides reducing the overfitting effect, dropout influences the training dynamics by randomly ignoring nodes during the training phase. While this does not inherently increase the learning speed, it introduces noise into the training, which can help the model generalize better to unseen data. A similar kind of improvement with respect to error is observed with experimentation on the SVH data set [23]. The parameters in convolution-shared filters are reduced while the dropouts are applied in fully connected layers, and it also increases the ability of the convolution layers to reduce overfitting. Data augmentation is used with regularization techniques for the dropout in AlexNet to reduce the overfitting effect in which 2200 images were classified by handling almost 50 million parameters [23].

3.4. *Early stopping*

In over-expressive training models, early stopping is used to avoid poor generalization. Deciding about the number of iterations, also known as the number of passes, is critical in such problems, as fewer passes will make the algorithm underfit. In this case, bias is reduced at the cost of higher variance. With a higher number of passes, the algorithm tends to overfit, causing higher bias with lower variance. Early stopping will eliminate this problem of manually assigning values to the parameters and number of passes. The principle of early stopping is to halt the training process when the validation performance no longer improves, preventing overfitting. It typically requires dividing the data into a training set and a validation set, not necessarily three subsets. The first one will be called the training set, the second is validation, and the third is the test set. The training data set will be used to compute the bias and weights with the help of a gradient descent optimization method. The validation set will be used to monitor the training process. If the number of passes is increased for both validation and training datasets, then computational error is reduced. However, the validation error starts to increase if the algorithm starts overfitting due to the higher number of passes. The early stopping parameter will ensure to stop the iterations as soon as it observes the deviation in validation error after that bias and weight values are returned. Early stopping is used to boost the optimization for non-convex loss functions and generalize the boosting algorithms [43]. This optimization leads to generalization improvement and an increase in validation and test accuracy.

3.5. *Hyper-parameter optimization*

Bayesian optimization is a powerful and most commonly used optimization technique in which the extreme values of any objective function are estimated within a bounded set of data. This method is especially useful to determine the extreme values for non-convex functions or black-box functions where expressions or any derivatives do not exist. For the objective function, a probabilistic model is generated using this optimization technique. This function will exploit the decision regarding the expected location of the function while keeping into account the uncertainty. Acquisition functions are used to find the next location and select functions on a priority basis. Acquisition functions are essential as they incorporate the prior belief of objective function and manage the trade-off between exploitation and exploration.

According to the Bayesian theorem, if a model has posterior probability E with O observations, this probability will be directly proportional to the likelihood of O given E times the prior probability of E as shown in Eq (5).

$$P\left({}^{E}/_{O}\right) \propto P\left({}^{O}/_{E}\right)P(E). \tag{5}$$

In Bayesian optimization, this prior probability accounts for the possible space of the objective function [49]. For ith sample s_l of the objective function, the observation is represented by $f(s_l)$ with an accumulated observation $O_{1:t} = \{s_{1:t}, f(s_{1:t})\}$. The likelihood function $P\left({}^{O_{1:t}}/_{f(s_{1:t})}\right)$ will be combined with the prior distribution to secure the posterior distribution, as shown in Eq (6).

$$P\left({}^{f(s_{1:t})}/_{O_{1:t}}\right) \propto P\left({}^{O_{1:t}}/_{f(s_{1:t})}\right)P(f(s_{1:t})). \tag{6}$$

The updated beliefs of the said objective function will be updated by this posterior distribution. The next expected location of the sample s_{t+1} will be secured through the acquisition function through Bayesian optimization. An automatic trade-off between exploitation and exploration is achieved, which allows only a minimum number of samples for the objective function to be evaluated. This technique is helpful even with the objective functions having local multiple maxima and minima. In the next step, maxima are computed through the acquisition function to evaluate the next sample, and this can be achieved through the calculations of covariance and mean of the predictions. Later, this objective function will be sampled at extreme values of the function, and then, all the parameters will be updated, and the whole process will repeat until termination criteria are reached.

Well-known CNN-trained models' weights can be used to initialize the training process by freezing a few initial layers and unfreezing the remaining layers to retrain the model for fine-tuning the hyper-parameters and weights. Figure 5 shows an example of transfer learning using VGG16, where the initial ten layers of the model are frozen to use the weights of the existing model and retrain it with the new COVID-19 data to fine-tune hyper-parameters and weights.

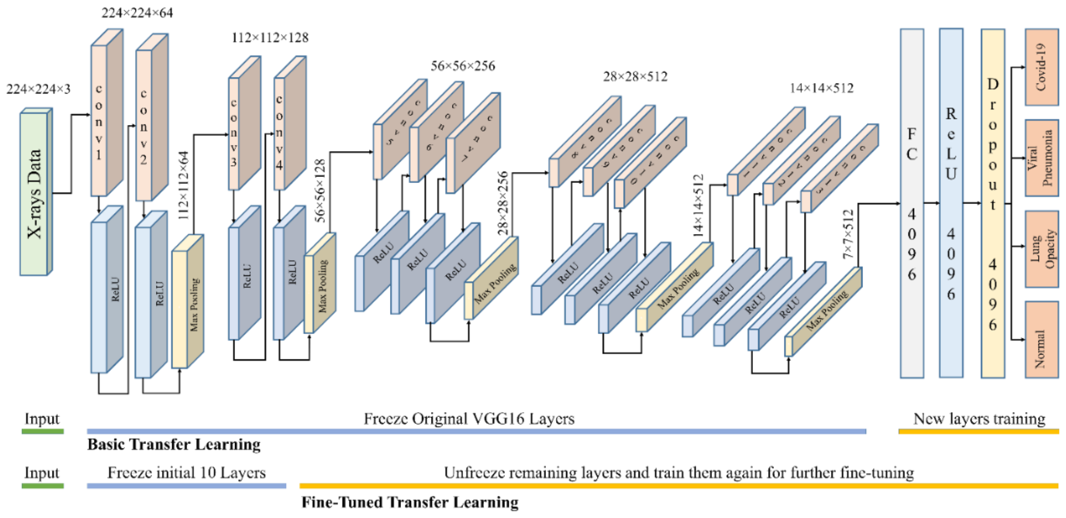

Figure 5. An example of transfer learning using VGG16 to fine-tune hyper-parameters and weights.

3.6. Loss function

A well-defined loss function will generate an actual number associated with any event and that event can include one or more variables. This same loss function will evaluate the model performance concerning its ability to predict the value $\hat{y}_i^{L+1}$ corresponding to the actual value y_i [50]. A decreased value of the loss function will correspond to better performance of the model.

If the output vector y_i can have possible values such as $y_i \in \{0,1\}$ and input event vector x having possible vectors $x = (x_1, x_2, x_3 \ldots x_h)$, then the loss function G will be used to map the input x to output y_i, as shown in Eq (7).

$$G(\hat{y}_i^{L+1}, y_i) = \frac{1}{h}\sum_{i=1}^{h}\big(y_i, (\sigma(x), w, b)\big). \tag{7}$$

Cross-entropy is the most widely used loss function, and most of the applications use cross-entropy loss function [51]. For example, if there is a training set label $\hat{y}_i^{L+1}$, then the probability that y_i output belonging to this training label is $P(y_i/u^{l-1}) = \hat{\imath}_h^{L+1} = 1$, and the probability that y_i output does not belong to training label $\hat{y}_i^{L+1}$ is $P(y_i/p^{l-1}) = \hat{y}_i^{L+1} = 0$. A cost function is given in the following equations in which y is the expected label.

$$P(y_i/p^{l-1}) = (\hat{y}_i^{L+1})^{y_i}\,(1 - \hat{y}_i^{L+1})^{1-y_i}, \tag{8}$$

$$\log\,[P(y_i/p^{l-1})] = \log\,[(\hat{y}_i^{L+1})^{y_i}(1 - \hat{y}_i^{L+1})^{1-y_i}], \tag{9}$$

$$log[P(y_i/p^{l-1})] = y_i \log(\hat{y}_i^{L+1}) + (1 - y_i)\log(1 - \hat{y}_i^{L+1}). \tag{10}$$

To minimize the following cost function.

$$\log\,[P(y_i/p^{l-1})] = -\log\,[(\hat{y}_i^{L+1})^{y_i}\,(1 - \hat{y}_i^{L+1})^{1-y_i}]. \tag{11}$$

If the number of training samples is i ,then the cost function is shown in Eq (12).

$$G(\hat{y}_i^{L+1}, y_i) = -\frac{1}{h}\sum_{i=1}^{h}(y_i \log(\hat{y}_i^{L+1}) + (1 - y_i)\log(1 - \hat{y}_i^{L+1})). \tag{12}$$

For simplicity, the predicted value p_j^L can be written as $\hat{y}_i$, where h is a vector that will store y_i representing the actual value. The cost function can then be written in a simplified form as shown in Eq (13).

$$G(\hat{y}_i, y_i) = \frac{1}{h}\sum_{i=1}^{h} f\left(y_i, \left(\sigma\left(w_{ij}p_i + b_i\right)\right)\right). \tag{13}$$

A parameter δ_i is defined such that $\delta_i : \{w_i, b_i\}$ in order to minimize the distance between the actual value y_i and predicted value $\widehat{y}_i$, then the objective function can be written as in Eq (14).

$$G(\delta_i) = \frac{1}{h}\sum_{i=1}^{h} f(y_i, \hat{y}_i). \tag{14}$$

3.7. Mini-batch gradient descent

The Mini-Batch Gradient Descent is an iterative optimization method, known as the steepest descent algorithm, in which the minima of any complex function are calculated by iterative negative derivatives of the function points. The basic definition of derivative is the calculation of the change in the rate of change in the slope of a given function that is similar to the change of any function at some points if the function is differentiable and continuous. In the batch gradient descent method, the gradient of any function-related parameters is computed; this method is also known as vanilla gradient descent. If output values are $\hat{y}_i = f(p^L, \delta) \in \mathbb{R}$, and the input value is $p \in \mathbb{R}$, then the output value will be predicted by processing the input value through hidden and intermediate layers. From Eqs (12)–(14), the layer-associated cost function is given in Eq (15).

$$\delta_i = \delta_i - \beta \frac{\partial}{\partial \delta_i} G(\delta_i : (y_i, \hat{y}_i)). \tag{15}$$

Where the learning rate is represented by β. The cost function G updates the parameters for all the respective values $0,1,2,3 \ldots i$ in the direction of the steepest descent method [52,53]. This batch gradient descent method completes the computation of the gradient points for the whole training set, which is a single update. If the data set is considerably large, then this process of update would be extremely inefficient with respect to computational time. Although the gradient is zero for the convex function at the optimum value, for the non-convex function; this method will tend to fall in a local minimum value. Due to these drawbacks in batch gradient, some modifications are proposed in this method in which training samples are divided into multiple batches, and every single batch will carry multiple samples at every iteration. Such a modified method is referred to as Mini-batch descent. These mini-batches are used to update the parameters and compute the errors. The variance of the mini-batches will reduce if the sum or average is applied and such techniques are applied to DL methods. If the batch size is assumed to be $i: i + n,$ then Equation 16 represents the mini-batch gradient.

$$\delta_i = \delta_i - \beta \frac{\partial}{\partial \delta_i} G\big(\delta_i :\ (y_i^{i:i+n}{}_{i}, \hat{y}_i^{i:i+n})\big). \tag{16}$$

The mini-batch gradient descent allows for faster convergence, better generalization, memory efficiency, and context specific accuracy increase.

3.8. Adaptive moment estimation (Adam)

Adaptive Moment Estimation is a technique in which the learning rates for all the parameters are computed adaptively. This optimization method stores the decaying average of the previous gradient square that is k_i. It will also store the same average for previous historic gradients that is v_i, with both averages being exponential. A similar method is followed in root mean squared propagation (RMSprop) and the moment [54]. Eqs (17) and (18) represent the update process of k_i and v_i.

$$v_{i+1} = \alpha_1 v_i + (1 - \alpha_1) g_i, \tag{17}$$

$$k_{i+1} = \alpha_2 k_i + (1 - \alpha_2) g_i. \tag{18}$$

Usually, the first and second moments are biased to zero during the process of initialization. The corrected moments are given in Eqs (19) and (20).

$$\hat{v}_i = \frac{v_i}{1 - \alpha_1^i}, \tag{19}$$

$$\hat{k}_i = \frac{k_i}{1-\alpha_2^i}. \tag{20}$$

After solving the above equations, Adam's update equation is obtained, as shown in Eq (21).

$$\delta_{i+1} = \delta_i - \frac{\beta}{\sqrt{\hat{k}_i+\epsilon}}\hat{v}_i. \tag{21}$$

In common practice, the values of ϵ, α_1 and α_2 are set to 10^{-8}, 0.9, and 0.999, respectively. Adam also leads to faster convergence, context dependent efficacy, and more robust to hyperparameter choice.

3.9. Model performance evaluation metrics

In addition to the optimization techniques described above, it is also important to understand how the resultant ML models should be evaluated to assess their performance. In the following, we describe some of the most commonly used performance metrics.

Confusion matrix is a commonly used method to describe a classification model's performance. For binary classification, the table consists of four quadrants: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). For example, Table 2 shows the confusion matrix for a binary classification model that predicts sick patients (positive) or healthy patients (negative) of 1000 patients. TP indicates that out of 130 sick patients, 100 were correctly identified as sick. FN shows that out of 130 sick patients, 30 were falsely identified as healthy patients. TN indicates that out of 870 healthy patients, 800 were correctly identified as healthy. FP shows that out of 870 healthy patients, 70 were incorrectly identified as sick patients.

Table 2. An example of a confusion matrix.

	Predicted Sick (Positive)	**Predicted Healthy (Negative)**
Sick Patients (Positive)	100 (TP)	30 (FN)
Healthy Patients (Negative)	70 (FP)	800 (TN)

Accuracy is the most commonly used performance metric. It just describes the total number of errors made by the model over the total predictions made (see Eq (22)). For example, based on the confusion matrix in Table 2, model accuracy is 90%. However, imbalanced dataset accuracy can be misleading. There are other performance metrics to handle such issues.

$$accuracy = \frac{FP+FN}{TP+TN+FP+FN} \tag{22}$$

Precision is the accuracy of positive predictions. It indicates how good is the model when it predicts the positive cases as can be calculated using Eq (23). For example, based on the confusion matrix in Table 2, model precision is 58.8%. It clearly shows that even though the model's overall accuracy is 90%, it is not a good model to identify sick patients (i.e. out of 170 sick patients predicted by the model, almost half (70) were not sick).

$$precision = \frac{TP}{TP+FP}. \tag{23}$$

Recall is another popular performance measure. It describes how well the model identifies all positive cases properly, as can be calculated using Eq (24). It is also known as a *sensitivity* measure. For example, based on confusion matrix in the Table 2, model's recall is 76.9%.

$$recall = \frac{TP}{TP+FN}. \tag{24}$$

Specificity does the same job as recall but for negative cases. It describes how well the model identifies all negative cases properly, as can be calculated using Eq (25). For example, based on the confusion matrix in Table 2, model's specificity is 91.9%.

$$Specificity = \frac{TN}{TN+FP}. \tag{25}$$

F1 Score is the *harmonic mean* of precision and recall. Whereas the regular mean treats all values equally, the harmonic mean gives much more weight to low values as can be calculated using Eq (26). As a result, the classifier will only get a high F1 score if both recall and precision are high.

$$F1\,Score = 2 \times \frac{precision \times recall}{precision+recall}. \tag{26}$$

4. Review of CNN optimization techniques: Case study of COVID-19 diagnosis

Medical image classification using CNNs can help doctors and radiologists accurately diagnose medical conditions by analyzing images such as X-rays, CT scans, and MRI scans. This section provides an extensive review of the CNN-based model optimization techniques being used in the context of COVID-19 diagnosis in the most recent literature work in MID. Though we have used COVID-19 Diagnosis as a case study within a larger MID field to limit the scope of the study, the findings remain relevant in other MID fields as well. Table 3 provides a summary of the detailed review showing the various optimization techniques applied of COVID-19 medical images. In the table, the dataset is specified as well as the technique, optimization techniques, accuracy results, and limitations. COVID-19 is a highly infectious and potentially deadly respiratory illness caused by the SARS-CoV-2 virus. Several techniques have been proposed for COVID detection, such as PCR, RT-PCR or medical imaging (X-ray, CT Scan, etc.) based approaches. In medical imaging, X-ray is considered one of the most commonly used approaches due to its ease of availability and affordability [55,56]. In a recent literature review [57], about 77% of COVID-19 detection approaches were X-ray based imaging technique. The importance of COVID X-ray image classification using CNN lies in its ability to accurately and promptly diagnose patients with COVID-19. With this type of disease, early diagnosis is critical in preventing its spread, improving patient conditions, and reducing the burden on healthcare systems. CNN-based COVID-19 detection models have the potential of assisting healthcare providers in making accurate and efficient diagnoses, particularly in situations where diagnostic testing resources are limited or unavailable. By analyzing X-ray images of the lungs, these models can extract and eventually detect the unique features of COVID-19 and distinguish them from other lung abnormalities.

In [58], a COVIDX-Net framework has been developed to run 2D X-ray images from 7 DL classifiers and analyze their performance. The framework runs images through 3 steps namely, pre-processing, training, and classification. The balanced X-ray images dataset [59] has been used in this framework which contains around 50 2D images of chest X-rays with 25 confirmed COVID-19 cases. The framework uses VGG19, DenseNet201, ResNetV2, InceptionV3, InceptionResNetV2, Xception, and MobileNetV2 networks for classification and comparison. The best performance was obtained when considering the VGG19 and DenseNet201 models. These models achieved 0.89 and 0.91 F1 score, respectively for the given dataset while performing the best in all of the aforementioned methods. They did not use any data augmentation and used learning rate of 0.001. The batch size was 7 and models were trained for 50 epochs. Though it was not clearly mentioned in the paper, most likely, the authors retrained the entire networks (without any frozen layers). VGG19 performed better, probably due to its smaller kernel size, which helps capture minor features present within X-ray images. Based on their findings, out of seven evaluated models, the authors recommend to use VGG19 and DenseNet201 models to identify the health status of patients against the COVID-19 in X-ray images.

In [60], the authors have considered ResNet50, ResNet101, ResNet152, InceptionV3, and InceptionResNetV2 for classifying and detecting COVID-19-positive patients. The dataset used in this paper contained 341 COVID-19 patients, 2800 normal (healthy) chest X-ray images [61], and 2772 bacterial and 1493 viral pneumonia chest X-ray images used from the Kaggle repository [62]. The learning rate was set to 0.00001, the batch size of 3 and all models were trained for 30 epochs. The data augmentation method was also used with scaling factor = 1/255, shear range = 0.1, zoom range = 0.1, and horizontal flipping enabled in the training dataset. The pre-trained models were used with two additional dense layers, and training was performed on these additional layers only. ROC analysis and confusion matrices were generated with 5-fold cross-validation. The transfer learning approach was also used for these models to accelerate the training and testing process while yielding good results and performance. The authors conclude that the pretrained ResNet50 model provided the best performance results with 98% accuracy. InceptionV3 and InceptionResNetV2 were ranked second and third in terms of achieved performance with 97% and 87% accuracy, respectively. Please note that these evaluations provided an optimal model than the one recommended in [58]. This can be attributed to a variety of factors including the availability of a larger dataset, application of data augmentation for increasing the sample size and cross-validation techniques for model evaluation.

In [63], the authors have conducted a study with 2 datasets collected from public medical repositories. The first dataset contained 1427 X-ray images with around 224 COVID-19 cases and other common pneumonia and normal conditions. The second dataset contained 224 COVID-19 case images and 1218 images of other normal and infection cases. CNNs such as VGG19, MobileNetV2, Inception, Xception, and InceptionResNetV2 were analyzed with different hyperparameters. During this study, the best performance was achieved when considering VGG19, with an accuracy level of 96.78%. The MobileNet v2 outperforms VGG19 in terms of reducing false negatives. During this experiment, all models were trained at different cut-off layers (i.e., model was re-trained on layers beyond the cut-off point only) as follows: VGG 19 = 18, MobileNetV2 = 10, Inception = 19, Xception = 120, and InceptionResNetV2 = 730. A dense neural network (one or two layers) was placed on top of these existing models with a ReLU activation function and a dropout layer in between each dense layer. The training was performed using the Adam optimizer, with a batch size of 64 for ten epochs. The results suggest that deep learning with X-ray imaging can help in extracting significant biomarkers related to the Covid-19 disease. Out of five evaluated models, VGG19 achieved overall better accuracy. However, MobileNet v2 model outperforms VGG19 in terms of specificity and is proven to be more effective model for reducing false negatives for this specific classification task with specific data sample.

In [64], deep features extraction from 2 datasets was carried out, where 50 X-ray images dataset [65] and a Kaggle dataset were utilized. The DL models AlexNet, VGG16, VGG19, GoogleNet, ResNet18, ResNet50, ResNet101, InceptionV3, InceptionResNetV2, DenseNet201, and XceptionNet were used for feature extraction. After the feature extraction, the Support Vector Machine was used to classify the COVID-19 cases. This combination of networks yielded 95.38% accuracy on the given datasets using ResNet50 model. Out of eight evaluated models, ResNet50 model provided the best features that can were used to classify using SVM for this dataset. Please note that data set used in this study was quite limited and imbalanced.

Binary classification-based anomaly detection approach, namely confidence-aware anomaly detection (CAAD), was introduced with 3 modules [66]. Shared feature extraction, anomaly detection, and confidence prediction modules were applied to the clinical X-VIRAL dataset. This dataset contains 43,369 pneumonia, non-viral pneumonia, and normal cases. Moreover, it was directly tested on 106 confirmed COVID-19 cases as well. The feature extraction was performed using 18-layer ResNet and EfficientNet- B0 models. A comparison was made between a binary classifier and an anomaly detection method. The proposed anomaly detection method using the EfficientNet- B0 model provided the highest accuracy, sensitivity, specificity, and AUC metrics. In this experiment, a learning rate of 0.0005 and batch size of 40 was used during training, leading to an 83.61% accuracy (AUC) on the given datasets. The authors conclude that the proposed anomaly detection approach works well in term of viral pneumonia screening on chest X-ray images and is superior to binary classification methods. The learning model confidence can greatly reduce the false negatives. The proposed CAAD model achieves an AUC of 83.61% and sensitivity of 71.70% on the unseen dataset (X-COVID), which is comparable to the performance of medical professionals. In [67], COVID-CAPS was introduced, which can perform reasonably well on small datasets, unlike CNNs, which require larger datasets. The main dataset used were [59] and [62]. This technique achieved 95.7% accuracy with less trainable parameters to configure (compared to CNN-based model). The comparison was made with [64] which had provided 95.4% accuracy using ResNet50. The NIH Chest X-ray dataset was used with over 112 thousand X-ray samples with around 14 anomalies for the pre-trained model. The pre-trained model COVID-CAPS resulted in 98.3% accuracy. The models used an Adam optimizer with an initial learning rate of 0.001 and a batch size of 16, and the model was trained for 100 epochs. The key contribution of this work is to provide an alternative to traditional CNN models with Capsule Networks (CapsNets) which had been previously used for other medical diagnosis such as brain and lung tumor detection. The main drawback of CNN based approach is that they are not able to capture spatial relations between image instances. As a result of this inability, CNNs cannot recognize the same object when it is rotated or subject to another type of transformation. This requires a large dataset with use of data augmentation techniques. Capsule Networks (CapsNets) are capable of capturing spatial information without the availability of a huge dataset.

Table 3. Summary of the latest articles on COVID 19 detection using DL-based models.

Ref	Method Name	Dataset Name	Results and Key Findings	Limitations
[58]	COVIDX-Net (7 CNNs)	COVID-19 X-ray images	91% (Out of seven evaluated models, VGG19 and DenseNet201 models are recommended to be used to identify COVID-19.)	• Very small dataset. • Only 50 images for training. • Only comparison between the existing models. • The novelty is not obvious.
[60]	ResNet50 with Transfer Learning	COVID-19 X-ray images	98% (The key conclusion is that the pretrained ResNet50 model provided the best performance results out of five evaluated models.)	• Datasets 1, 2 and 3 are small and imbalanced due to absence of images for bacterial and viral pneumonia. • The performance comparison is based on existing CNN models and the uniqueness of the proposed approach is questionable.
[63]	VGG19 with Transfer Learning	Public Medical Repo collections for X-ray images	96.78% (Out of five evaluated models, VGG19 achieved overall better accuracy while MobileNet v2 model outperforms VGG19 in terms of specificity and thus reducing the false negatives.)	• Limited number of images in the dataset of the patients tested positive for COVID-19. • Rather than just classifying the patient as COVID-19 positive or negative, a severity measure of infection can be calculated based on observed symptoms.
[64]	SVM + ResNet50	COVID-19 X-ray images and Kaggle dataset	95.38% (Out of eight evaluated models, ResNet50 model provided the best features which were used to classify using SVM.)	• Imbalanced small dataset. • Used existing CNN pre-trained models for detection and classification using SVM. • Novelty in terms of contribution is not mentioned clearly.
[66]	Confidence aware anomaly detection (CAAD)	COVID-19 X-ray images and XVIRAL dataset	AUC: 83.61% (The proposed CAAD model achieves an AUC of 83.61% and sensitivity of 71.70% on the unseen X-COVID dataset.)	• Low classification accuracy. • Feature extraction is done manually which is subjective as the number of features chosen will affect the classification accuracy.

Continued on next page

Ref	Method Name	Dataset Name	Results and Key Findings	Limitations
[67]	COVID-CAPS	NIH Chest X-ray images and COVID-19 X-ray images	98.3% (Did experiments with Capsule Networks (CapsNets) which had been previously used for other medical diagnosis such as brain and lung tumor detection. CapsNets are capable of capturing spatial information without the availability of a huge dataset.)	• CNN models are unable to capture spatial relations between images. • small dataset of COVID-19 images. • The capsule layers incorporated in this approach are subjective in nature and further add to the complexity and time factor during the training process.
[68]	DenseNet201	423 COVID-19 images public dataset	99.7% (DenseNet201 outperformed other CNN networks when image augmentation was used and CheXNet outperformed on actual dataset.)	• Used augmentation for enlarging the dataset which is questionable. • Novelty is not reflected as the study does performance comparison for already existing pre-trained CNN models.
[69]	Resnet50	135 COVID-19 images and 320 Other cases dataset	89.2% (102 COVID-19 chest X-rays images are used in training indicates it may, with more data, be possible to build deep neural network models for accurate diagnosis.)	• Imbalanced small dataset. • The resolution of COVID-19 images is low which affects the classification accuracy. • Lack of information on types of chest X-rays.
[71]	Early fusion technique with MLP	RYDLS-20	F1 score: 0.89 (COVID-19 identification in an unbalanced environment with more than three classes with novel hierarchical classification approach.)	• Limited sized dataset used for experiments. • Manual feature extraction process is subjective and time-consuming.
[73]	Fine-tuned ResNet50	COVID-19 X-ray images + CT images	96.61% (Comparison of CNN models; Resnet50 MobileNet_V2 and Inception_Resnet_V2.)	• Lacks novelty in the proposed approach, as it is merely comparing the classification accuracy of already existing CNN models.

Continued on next page

Ref	Method Name	Dataset Name	Results and Key Findings	• Limitations
[75]	CheXNet	QaTa-Cov19 dataset	95.9% (Proposed a novel convolution support estimation network (CSEN) architecture that can be seen as a bridge between deep learning models and representation- based methods. It also provides a benchmark X-ray data set, namely QaTa-Cov19.)	• Imbalanced dataset of covi19 and non-COVID-19 X-ray images.
[76]	Deep feature fusion and ranking + CNN	150 COVID-19 images-based CT Images dataset	98.27% (The main contribution of this work is to evaluate an approach where features were obtained using multiple pre-trained CNN networks and then these features were fused and ranked. The resultant features were used to train on SVM.)	• Novelty is not reflected as the study does performance comparison for already existing pre-trained CNN models.
[77]	ResNet50	500 images based COVID-19 Chest X-ray images dataset	95% (Our main finding of this study is that the CNN based systems (even with a limited number of cases: 250 of them COVID-19 positive and 250 COVID-19 negative), showed an interesting performance.)	• Limited number of COVID-19 positive cases in the dataset. • Geographically restricted results that may not be generalized (from Lombardy, Italy). • Clinical conditions were not considered (e.g., symptoms and pulse oximeter data).
[79]	COVNet	19 patients CT Images dataset	93%	• Provides opinion on the suitability of CT scans as a diagnostic tool for COVID-19 detection.

A study on multiple CNN architectures was conducted, including DenseNet variants, CheXNet, MobileNet, SqueezeNet, etc. [68]. The DenseNet201 outperformed other networks even when networks such as CheXNet were already trained on X-ray images. The networks were trained and evaluated on personally collected datasets by authors, which contained 439 images of COVID-19 and other cases. The models were trained with and without data augmentation. The augmentation included rotation and scaling of images. They used Gradient Descent (SGD) with momentum optimizer with a learning rate of 0.001, momentum update of 0.9, and mini-batch size of 16 images with 20 backpropagation epochs. For binary classification (COVID-19 Or Normal), CheXNet and ResNet18 provided the highest accuracy of 99.41% without any augmentation, while DenseNet201 provided the highest accuracy of 99.7% with data augmentation. For three class problems (COVID-19, Viral Pneumonia, and Normal), ChexNet provided the highest accuracy (97.74%) without data augmentation, and DensetNet201 provided the highest accuracy (97.94%) with data augmentation. The experimentation results were obtained using 5-fold cross-validation. The authors observed that DenseNet201 outperformed other deep CNN networks when image augmentation was used and CheXNet, that is a variant of DenseNet, outperformed other networks when image augmentation was not used. This is because CheXNet was pre-trained on a large X-ray database and it showed better performance on a small non-augmented image dataset. However, a deeper version of DenseNet, when trained on a large augmented dataset, Dense201 outperforms CheXNet.

In [69], ResNet50, VGG16, and a small crafted CNN were adopted for being trained on a small dataset of 135 COVID-19 images and 320 other pneumonia cases. The last dense layers of VGG16 and ResNet were removed and replaced by the trainable part, which consisted of global average pooling. This was followed by a fully connected layer of 64 units with dropout and finally, a classification layer with sigmoid output. The cyclic learning rate was used with the base learning rate and max learning rate set to 0.0001 and 0.001, respectively. RMSprop was used for optimization and binary-cross-entropy as the loss function. Data was augmented using horizontal flip only. The models were evaluated using 10-fold cross-validation. ResNet50 gave better results on the collected dataset with 89.2% accuracy when trained with 10-fold cross-validation. This work reported a 6% false positive rate that may be reduced by biasing the training data to include more non COVID-19 cases. The authors conclude that this preliminary work with just 102 COVID-19 chest X-rays used in training indicates it may, with more data, be possible to build deep neural network models for accurate diagnosis.

In [70], the authors used DarkCovidNet as a CNN classifier, which was inspired by the DarkNet-19 model used in YOLO for multi-class real-time object detection. The COVID-19 X-ray images dataset [59] was used in this experiment for binary (COVID-19 vs. non-COVID-19) classification, and the image dataset [62] was used for three class classifications (COVID-19, pneumonia, and normal). The dataset was split into 80% training and 20% validation with 5-fold cross-validation. The model was trained for 100 epochs. The number of filters in the proposed DarkCovidNet model was increased gradually from 8, 16, 32, 64, 128, and 256. The learning rate was set to 0.003. The model yielded 98.08% accuracy for binary classes and 87% for 3-class classification. This model can be used in real-time classification as well. The main contribution of this paper is to evaluate the performance of DarkCovidNet model for COVID-19 diagnosis. It provides accuracy results which are quite similar to other CNN based networks.

In [71], the authors explored COVID-19 and other pneumonia (MERS, SARS, Varicella, Streptococcus, Pneumocystis) in a hierarchical classification using traditional machine learning approaches (KNN, SVM, MLP, Decision Tree and Random Forests). A chest X-ray images-based dataset, namely RYDLS-20, was collected by authors consisting of images of healthy lungs and with pathogens. The Dataset had huge data imbalances (reflecting real-world situations). Combinations of feature extraction methods were used in the experiment under the name of the early fusion technique. The data was then resampled after classification. The multi-class learners (MLPs) performed better, with a reported 0.65 F1 score, and COVID-19 identification in hierarchical classification, with a reported 0.89 F1 score. The top identification rate obtained for COVID-19 identification in the same paper was the best nominal rate obtained in an unbalanced environment with more than three classes. The authors in [72] used multi-level thresholding along with an SVM classifier for COVID-19 detection on the COVID-19 chest X-ray images dataset. The dataset consisted of 40 contrast-enhanced lung X-rays with 15 normal and 25 with COVID-19 infection. This resulted in promising performance, reaching 97.48% accuracy on 512 × 512 same-sized images. This work reported an F1-Score of 0.8889, which was the best nominal rate (at the time of article publication) obtained for the task of COVID-19 identification in an unbalanced environment with more than three classes. The other main contribution of this work is to provide a novel hierarchical classification approach (considering types of pneumonia).

The authors in [73] focused on a binary classification problem consisting of the identification of pneumonia (which includes COVID-19) using a well-known Kaggle dataset [62]. The dataset was balanced using image augmentation (the normal class was doubled using image augmentation techniques). The study was conducted for fine-tuned versions of VGG16, VGG19, Xception, InceptionV3, Resnet50, MobileNetV2, InceptionResNetV2, and DenseNet201. Around 5856 Chest X-ray and CT images were used in this study. The models were trained for 300 epochs with a batch size of 32 and a learning rate of 0.00001. On this large dataset, ResNet50 outperformed other networks by achieving 96.61% accuracy. InceptionResNetV2, on the other hand, yielded 96.09% accuracy. Most of the other networks were below 90% in accuracy on the given dataset. The key finding of this work was that Resnet50, MobileNet_V2 and Inception_Resnet_V2 provided better accuracy (more than 96%) on this particular dataset and outperformed other architectures.

In [74], a dataset of CT-based images was used to classify COVID-19 cases. MODE-based CNN was proposed with 20-fold cross-validation to avoid overfitting. Comparisons were conducted against traditional CNN and Artificial Neural Network models. This model outperformed those networks by 2.01% and achieved an accuracy of 97.89%. The key contribution of this work is to evaluate the performance of MODE-based CNN on chest CT based images (rather than X-ray images).

In [75], the authors propose a novel convolution support estimation network (CSEN) to identify COVID-19 using a relatively larger dataset QaTa-Cov19 which contained over 6200 X-ray images with 462 COVID-19 patient X-rays. Feature extraction was performed using CheXNet. This approach resulted in 95.9% accuracy, 98% sensitivity and 95% specificity for COVID-19 detection. The main contribution of this work is to propose and evaluate a novel convolution support estimation network (CSEN) architecture that can be seen as a bridge between deep learning models and representation-based methods. It also provides a benchmark X-ray data set, namely QaTa-Cov19.

In [76], computerized tomography (CT) images dataset was used with deep fusion and ranking based method. Dataset contained over 150 images. The images were labeled as COVID-19 and had no label. Feature fusion and ranking methods were applied during the data preprocessing phase, and then SVM was applied. The CNN was later applied with transfer learning, which resulted in 98.27% accuracy, 97.63% precision, 98.28% F1 score, 98.93% sensitivity, and 97.60% specificity. The main contribution of this work is to evaluate an approach where features were obtained using multiple pre-trained CNN networks and then these features were fused and ranked. The resultant features were used to train a SVM based classifier.

In a different study [77], the ResNet50 model was used with 250 COVID-19 and 250 non-COVID-19 chest X-ray images dataset. 10-fold cross-validation was applied to avoid overfitting. The training was conducted for 30 epochs with a batch size of 8. Results were 80% in terms of sensitivity, 81% in terms of specificity, and around 95% accuracy. The main finding of this study is that the CNN based system (even with a limited number of cases: 250 of them COVID-19 positive and 250 COVID-19 negative), showed an interesting performance. [78] provides a short survey of DL methods used for COVID-19 detection using radiology images and highlights a few available data sources and the need for future AI-based applications to handle COVID-19. The survey is not detailed enough; it only covers about 12 research publications and does not provide the details of the machine learning methods used.

In the early days of COVID-19 discovery, a dataset of 19 patients' CT scans was collected [79]. This data was insufficient for training an AI model, so over 4356 chest CT exam images were collected later to be used with COVNet. Researchers reported 90% sensitivity and 96% specificity on these early tests [79]. COVNet was built on ResNet50 with a few additions, such as a new Max Pooling layer and Fully Connected Layer. This model was provided with CT images for both training and testing. CT images dataset resulted in 0.96 AUC, which is quite equivalent to 95.5% accuracy [80].

4.1. Discussion

Parameter optimization is a crucial step in the training of CNNs. CNNs are complex models that typically have millions of parameters, and optimizing these parameters is necessary for the network to learn the features of the input data and achieve high performance during classification.

A comparative analysis of the research work shows that researchers have attempted to use a variety of optimization techniques to improve overall accuracy. Some of them have worked on binary classification while others have attempted on a more difficult multi-class and hierarchical classification problems. The role of available dataset is also critical and we have noticed continuous efforts to improve the size and quality of existing available datasets (sometimes based on integration of multiple datasets). Due to scarcity of large datasets, most of the existing work has attempted to use pre-trained models (VGG, ResNet, CheXNet, DenseNet, DarkCovidNet, etc.) to improve overall accuracy. Some have attempted to use pre-trained models for feature extraction and then use traditional classifiers like SVM. Researchers have used different batch sizes (3, 7, 8, 16, 32, 40, and 64), number of filters (8 to 256), and learning rates (from 0.00001 to 0.003) to best fit their proposed DL models.

The reviewed studies in Section 4 showed that the selection of specific CNN architectures and hyperparameters is influenced by the unique characteristics of the medical image datasets and the specific objectives of each study. For instance, architectures like VGG19 and DenseNet were chosen for their deep layered structures, which are beneficial for capturing the intricate patterns in medical images, while models like MobileNetV2 and InceptionResNetV2 were selected for their efficiency and depth, which are crucial for handling the computational constraints of medical image analysis. Hyperparameters such as learning rate, batch size, and epochs were adjusted based on the dataset's size and complexity, with smaller batch sizes and appropriate learning rates chosen to improve the model's ability to generalize from the medical images, particularly when dealing with limited and imbalanced datasets. The decision to employ transfer learning, as seen with ResNet50 and InceptionV3, is influenced by the need to leverage pre-trained networks to enhance feature extraction capabilities, especially when the dataset is not large enough to train a model from scratch effectively. A carefully designed CNN architectures by incorporating domain knowledge can capture complex features from the images and produce higher accuracies. However, designing optimal CNN architectures can be time-consuming and requires expertise, and sometime leads to overfitting specially in the case of limited training dataset. Compared to the architecture designing, parameters optimization methods can automate parameter tuning and can reduce manual effort which can find good features efficiently. Fine tuning parameters requires experimentation and can also get stuck in local minima which leads to suboptimal solutions. The best approach often involves a combination of both. Complex tasks may benefit from well-designed architectures, while simpler tasks might work well with pre-designed architectures and parameter optimization.

These optimization techniques can also be used in other medical domains. When leveraging AI techniques for optimizing medical images related to tumors and cancers across diverse organs, several challenges arise. These include variations in image quality and variability due to imaging modalities and patient factors, such as anatomy and pathology. Complex anatomical structures in organs like the brain, lungs, and liver add another layer of challenge, requiring AI algorithms to adapt to diverse features for accurate analysis. Detecting and segmenting lesions, especially in organs with overlapping structures or irregularly shaped lesions, poses significant hurdles that need to be addressed.

Further, data imbalance and rarity are also critical considerations, particularly for diseases with low prevalence rates or limited datasets available for training AI models. Ensuring model generalizability and preventing biases become crucial tasks in such scenarios. Moreover, the interpretability and explainability of AI results are essential for aiding clinicians in understanding diagnostic decisions, especially in complex cases that heavily rely on detailed image analysis.

CNN optimization-based classification techniques have also shown their effectiveness in different medical domains in terms of its capability in accurately performing diagnosis of various types of disease. For instance, in [81], A Bayesian Optimization-based efficient hyperparameter optimization technique for CNN is proposed to classify CE-MRI images into three types of brain tumors: Glioma, Meningioma, and Pituitary. The optimized CNN method was able to achieve a 98.70% accuracy. To demonstrate the feasibility of automating hyperparameter optimization, the proposed model was able to outperform state-of-the-art methods on the CE-MRI dataset. Also, in [82], the authors have presented a region growing-fuzzy c-means clustering model optimized with Harris Hawks CNN-based brain tumor recognition process. The proposed methodology was evaluated on the Kaggle Brain MRI for Brain Tumor Detection dataset, and achieved an overall tumor recognition accuracy of 98%. In [83], the authors have presented a multi-classification of brain tumors for early diagnosis using 3 CNN models whose almost all hyper-parameters are automatically tuned using grid search. The first model achieved a 99.33% accuracy for brain tumor detection. The second model achieved an accuracy of 92.66% when classifying the brain tumors into three grades as Grade II, Grade III and Grade IV. With an accuracy of 98.14%, the third model was able to classify the brain tumors into three grades as Grade II, Grade III and Grade IV. In a different type of disease; namely lung cancer diagnosis, CNN based models have also performed well. In [84], A new method, utilizing the Otsu thresholding-based cuckoo search optimization and CNN classifier approach, was used for recognizing lung cancer in CT images. When tested on lung cancer CT images collected from a private hospital (Satyam diagnostic center, Anantapur), experimental results of the proposed method achieved an accuracy of 96.97%. Also, in [85], to optimize hyper-parameters of CNN, an optimized convolution neural network model based on gray wolf optimization algorithm is developed in this research. The optimized CNN performance was evaluated on the NIH/NCI Lung Image Database Consortium dataset, and achieved an accuracy of 98.21%, which is higher than accuracy of the conventional CNN. Finally, in [86], a 2D convolutional neural network (2D CNN) with Taguchi parametric optimization method is proposed for automatically recognizing lung cancer from CT images. When the proposed method is evaluated on the LIDC-IDRI and SPIE-AAPM datasets, experimental results showed that is it 6.86% and 5.29% more accurate than the conventional 2D CNN method.

Based on existing research studies, it is also noticed that there is a scarcity of large medical datasets. Creating standardized labeled datasets for medical conditions is crucial for advancing AI research in healthcare. The following are few recommendations that can be taken to address the challenge of scarcity and encourage the creation of large medical datasets:

- **Collaborative Approach:** Foster collaboration among healthcare institutions, research groups, and AI developers to consolidate private datasets and create large, high-quality datasets. This involves sharing anonymized data under strict privacy and security protocols to ensure patient confidentiality.
- **Merge Public Datasets:** Another possibility is to leverage publicly available datasets such as those provided by government agencies, research institutions, or healthcare organizations. These datasets can be standardized and consolidated for larger datasets.
- **Data Augmentation and Synthesis:** The third possibility is to apply advanced data augmentation techniques, including transformations like rotation, flipping, and noise addition, to expand existing datasets. Additionally, synthetic data generation methods, such as GANs, to create realistic medical images can be explored. These approaches can be combined with expert labeling to ensure and enhance dataset quality.

Among the above suggested three approaches, collaborative approach is key as the datasets produced under collaborations will be the basis for data augmentation and synthesis. Building dedicated platforms or consortiums that bring together healthcare institutions and AI researchers. These platforms can serve as hubs for sharing expertise, resources, and data, fostering a collaborative ecosystem. These platforms also implement robust quality control measures, which include defining standardized labeling protocols and guidelines to ensure consistency and accuracy across medical datasets. A common practice to build these platforms is on a public cloud. Research grants can also be offered for joint research projects that involve both medical experts and AI researchers. These projects can focus on specific medical conditions or imaging modalities, where medical practitioners provide domain knowledge, and AI researchers contribute technical expertise in healthcare data analysis and model development. Another approach can be developing and standardizing data sharing agreements and frameworks that facilitate the ethical and secure exchange of medical data between healthcare institutions and AI researchers. Methods need to developed for data anonymization, patient consent, and compliance with regulatory standards to ensure data privacy and security.

By implementing these strategies and initiatives, stakeholders can cultivate a collaborative environment where medical practitioners and AI researchers work together seamlessly to create standardized labeled datasets for various medical conditions. This collaboration not only accelerates AI-driven innovations in healthcare but also ensures the ethical and responsible use of medical data for improving patient care.

5. Conclusion and future directions

The use of AI and ML for medical diagnosis has been extensively researched as a potential tool for the automatic diagnosis of medical conditions. The successful integration of AI/ML techniques into medical devices for use in hospitals would represent a significant accomplishment, as it offers numerous benefits to both patients and medical institutions. These benefits include reduced costs, increased precision, and improved access to medical services for a wider population. Early diagnosis/detection of many diseases is also possible, leading to early intervention and, ultimately, saving lives and further reducing costs. The use of AI/ML in medical diagnosis would have far-reaching implications, including improved medical services in remote areas and developing countries.

This paper provides a comprehensive review of the most pertinent studies to date on the application of deep CNN architecture optimization techniques for MID. The optimization techniques examined have the advantage of being generic and, hence, can be applied to diagnose any disease. As a case study, the optimization techniques are explored in the context of COVID-19 diagnosis. The impact of the various related variables, including datasets and AI/ML techniques, is investigated in detail as well. Furthermore, the paper highlights noteworthy shortcomings and challenges associated with existing optimization techniques. The optimization techniques applied to CNNs for COVID-19 diagnosis, such as specific architecture choices and hyperparameter tuning, have demonstrated applicability in other medical imaging domains, including cancer detection and neuroimaging.

A detailed analysis reveals that the accuracy of binary and multiclass classification is improving; however, a lack of collaboration between medical practitioners and computer scientists/engineers is evident, which is manifested in the scarcity of reasonably large labeled datasets. Medical professionals can add value and expedite the development of AI/ML learning algorithms by providing a sizable standard labeled dataset for all medical conditions. The datasets can comprise medical images or relevant symptom and diagnosis data depending on the medical conditions and diseases. Many research medical institutions are hiring data engineers and computer scientists to utilize the power of artificial intelligence in medical research. This kind of collaboration could lead to huge strides in the medical field, which include the development of tools for medical diagnosis that are AI/ML based.

Annually, computational power continues to increase, yet the practical application of AI/ML techniques remains limited in medical disease diagnosis. Theoretical studies on small-scale datasets exist, with only a few patented and approved devices. The authors posit that this lack of practical use of AI/ML applications is attributed to the insufficient collaboration between computer scientists and engineers with medical institutions or professionals in the medical field. Such lack of collaboration may arise for various reasons, including legal considerations such as patient privacy. Greater efforts ought to be undertaken to facilitate this partnership, which would assuredly hasten the process of integrating AI/ML-based devices in the medical diagnosis field for the benefit of all.

As part of future work, these optimization techniques can be investigated for other medical images, such as tumors and cancers in diverse organs. Moreover, the scope of this study can be extended to encompass non-medical images. It is interesting to see the contribution of each optimization technique on larger datasets of diverse medical and non-medical images. Similarly, the influence of these optimization techniques can be studied for large images such as whole slide imaging (WSI) and high-resolution computed tomography (HRCT). Exploring optimization techniques for both medical and non-medical image datasets allow for the knowledge transfer of methodologies, enhancing algorithmic robustness and adaptability across various contexts. However, the key limitation is that optimizations effective in non-medical contexts may not directly apply to medical imaging due to the latter's unique requirements for precision, interpretability, and compliance with clinical standards. Challenges that might arise in this case would be similar to those found in this study, which include lack of standard datasets, lack of large datasets, quality of dataset issues, etc. however, the integration of AI into medical field and applied to various organs and diseases is expected to make life better for the patient through early diagnosis and early intervention and life better for the providers in terms of reducing cost and expediting the process.

Author contributions

Ghazanfar Latif: conceptualization, methodology, software, supervision, validation; Jaafar Alghazo: writing—original draft preparation, formal analysis, writing—review; Majid Ali Khan: conceptualization, writing—review and editing; Ghassen Ben Brahim: validation, visualization, resources; Khaled Fawagreh: writing—original draft preparation, visualization, Nazeeruddin Mohammad: conceptualization, writing—review and editing, supervision.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. V. Sharma, M. G. Dastidar, S. Sutradhar, V. Raj, K. De Silva, S. Roy, A step toward better sample management of COVID-19: On-spot detection by biometric technology and artificial intelligence, *COVID-19 Sustain, Develop. Goals*, **2022** (2022), 349–380. https://doi.org/10.1016/B978-0-323-91307-2.00017-1
2. G. Latif, H. Morsy, A. Hassan, J. Alghazo, Novel coronavirus and common pneumonia detection from CT scans using deep learning-based extracted features, *Viruses*, **14** (2022), 1667. https://doi.org/10.3390/v14081667
3. A. Islam, T. Rahim, M. Masuduzzaman, S. Y. Shin, A blockchain-based artificial intelligence-empowered contagious pandemic situation supervision scheme using internet of drone things, *IEEE Wirel. Commun.*, **28** (2021), 166–173. https://doi.org/10.1109/MWC.001.2000429
4. T. Rahim, M. A. Usman, S. Y. Shin, A survey on contemporary computer-aided tumor, polyp, and ulcer detection methods in wireless capsule endoscopy imaging, *Comput. Med. Imag. Grap.*, **85** (2020), 101767. https://doi.org/10.1016/j.compmedimag.2020.101767
5. G. Latif, DeepTumor: Framework for brain MR image classification, segmentation and tumor detection, *Diagnostics*, **12** (2022), 2888. https://doi.org/10.3390/diagnostics12112888

6. T. Rahim, S. A. Hassan, S. Y. Shin, A deep convolutional neural network for the detection of polyps in colonoscopy images, *Biomed. Signal Proces.*, **68** (2021), 102654. https://doi.org/10.1016/j.bspc.2021.102654
7. A. Bashar, G. Latif, G. Ben Brahim, N. Mohammad, J. Alghazo, COVID-19 pneumonia detection using optimized deep learning techniques, *Diagnostics*, **11** (2021), 1972. https://doi.org/10.3390/diagnostics11111972
8. E. Hussain, M. Hasan, M. A. Rahman, I. Lee, T. Tamanna, M. Z. Parvez, CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images, *Chaos Soliton. Fract.*, **142** (2021), 110495. https://doi.org/10.1016/j.chaos.2020.110495
9. G. Latif, G. Ben Brahim, D. N. F. A. Iskandar, A. Bashar, J. Alghazo, Glioma tumors' classification using deep-neural-network-based features with SVM classifier, *Diagnostics*, **12** (2022), 1018. https://doi.org/10.3390/diagnostics12041018
10. I. Iqbal, M. Younus, K. Walayat, M. U. Kakar, J. Ma, Automated multi-class classification of skin lesions through deep convolutional neural network with dermoscopic images, *Comput. Med. Imag. grap*, **88** (2021), 101843. https://doi.org/10.1016/j.compmedimag.2020.101843
11. I. Iqbal, K. Walayat, M. U. Kakar, J. Ma, Automated identification of human gastrointestinal tract abnormalities based on deep convolutional neural network with endoscopic images, *Intell. Syst. Appl.*, **16** (2022), 200149. https://doi.org/10.1016/j.iswa.2022.200149
12. V. Shah, R. Keniya, A. Shridharani, M. Punjabi, J. Shah, N. Mehendale, Diagnosis of COVID-19 using CT scan images and deep learning techniques, *Emerg. Radiol.*, **28** (2021), 497–505. https://doi.org/10.1007/s10140-020-01886-y
13. M. M. Rahaman, C. Li, Y. Yao, K. Frank, M. A. Rahman, Q. Wang, et al., Identification of COVID-19 samples from chest X-Ray images using deep learning: A comparison of transfer learning approaches, *J. X-Ray Sci. Technol.*, **28** (2020), 821–839. https://doi.org/10.3233/XST-200715
14. A. S. Al-Waisy, S. Al-Fahdawi, M. A. Mohammed, K. H. Abdulkareem, S. A. Mostafa, M. S. Maashi, et al., COVID-CheXNet: Hybrid deep learning framework for identifying COVID-19 virus in chest X-rays images, *Soft Comput.*, **27** (2020), 2657–2672. https://doi.org/10.1007/s00500-020-05424-3
15. Y. Chang, X. Jing, Z. Ren, B. Schuller, CovNet: A transfer learning framework for automatic COVID-19 detection from crowd-sourced cough sounds, *Front. Digit. Health*, **3** (2022), 799067. https://doi.org/10.3389/fdgth.2021.799067
16. M. Elpeltagy, H. Sallam, Automatic prediction of COVID− 19 from chest images using modified ResNet50, *Multimed. Tools Appl.*, **80** (2021), 26451–26463. https://doi.org/10.1007/s11042-021-10783-6
17. R. K. Patel, M. Kashyap, Automated diagnosis of COVID stages from lung CT images using statistical features in 2-dimensional flexible analytic wavelet transform, *Biocybern. Biomed. Eng.*, **42** (2022), 829–841. https://doi.org/10.1016/j.bbe.2022.06.005
18. D. K. Redie, A. E. Sirko, T. M. Demissie, S. S. Teferi, V. K. Shrivastava, O. P. Verma, et al., Diagnosis of COVID-19 using chest X-ray images based on modified DarkCovidNet model, *Evol Intell.*, **16** (2022), 729–738. https://doi.org/10.1007/s12065-021-00679-7
19. F. Özyurt, Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures, *J. Supercomput*, **76** (2020), 8413–8431. https://doi.org/10.1007/s11227-019-03106-y

20. D. H. Hubel, T. N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, *J. Physiol.*, **160** (1962), 106–154. https://doi.org/10.1113/jphysiol.1962.sp006837
21. Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, In: *The handbook of brain theory and neural networks*, 1995.
22. G. Latif, J. Alghazo, L. Alzubaidi, M. N. Nasser, Y. Alghazo, Deep convolutional neural network for recognition of unified multi-language handwritten numerals, In: *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, 2018. https://doi.org/10.1109/ASAR.2018.8480289
23. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks. *Commun. ACM*, **60** (2017), 84–90. https://doi.org/10.1145/3065386
24. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, 2014, arXiv:1409.4842.
25. S. Alghamdi, M. Alabkari, F. Aljishi, G. Latif, A. Bashar, Lung cancer detection from LDCT images using deep convolutional neural networks, In: *International Conference on Communication, Computing and Electronics Systems*, Singapore: Springer, **733** (2021), 363–374. https://doi.org/10.1007/978-981-33-4909-4_27
26. D. A. Alghmgham, G. Latif, J. Alghazo, L. Alzubaidi, Autonomous traffic sign (ATSR) detection and recognition using deep CNN, *Procedia Comput. Sci.*, **163** (2019), 266–274. https://doi.org/10.1016/j.procs.2019.12.108
27. G. Latif, N. Mohammad, R. AlKhalaf, R. AlKhalaf, J. Alghazo, M. Khan, An automatic arabic sign language recognition system based on deep CNN: An assistive system for the deaf and hard of hearing, *Int. J. Comput. Digital Syst.*, **9** (2020), 715–724. http://doi.org/10.12785/ijcds/090418
28. B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, In: *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, **1** (2014), 487–495.
29. M. M. Butt, G. Latif, D. N. F. A. Iskandar, J. Alghazo, A. H. Khan, Multi-channel convolutions neural network based diabetic retinopathy detection from fundus images, *Procedia Comput. Sci.*, **163** (2019), 283–291. https://doi.org/10.1016/j.procs.2019.12.110
30. D. C. Cireşan, U. Meier, J. Masci, L. Gambardella, J. Schmidhuber, Flexible, high performance convolutional neural networks for image classification, In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2011, 1237–1242. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-210
31. G. Lokku, G. H. Reddy, M. N. G. Prasad, OPFaceNet: Optimized face recognition network for noise and occlusion affected face images using hyperparameters tuned convolutional neural network, *Appl. Soft Comput.*, **117** (2022), 108365. https://doi.org/10.1016/j.asoc.2021.108365
32. S. Y. Kim, Z. W. Geem, G. Han, Hyperparameter optimization method based on harmony search algorithm to improve performance of 1D CNN human respiration pattern recognition system, *Sensors*, **20** (2020), 3697. https://doi.org/10.3390/s20133697
33. G. Latif, K. Bouchard, J. Maitre, A. Back, L. P. Bedard, Deep-learning-based automatic mineral grain segmentation and recognition, *Minerals*, **12** (2022), 455. https://doi.org/10.3390/min12040455
34. J. Bruna, S. Mallat, Invariant scattering convolution networks, *IEEE T. Pattern Anal.*, **35** (2013), 1872–1886. https://doi.org/10.1109/TPAMI.2012.230

35. S. Lawrence, C. L. Giles, A. C. Tsoi, What size neural network gives optimal generalization? Convergence properties of backpropagation, In: *Digital Repository at the University of Maryland*, 1998.
36. L. Wan, M. Zeiler, S. Zhang, Y. Cun, R. Fergus, Regularization of neural networks using dropconnect, In: *ICML'13: Proceedings of the 30th International Conference on International Conference on Machine Learning*, **28** (2013), 1058–1066.
37. Q. Xu, M. Zhang, Z. Gu, G. Pan, Overfitting remedy by sparsifying regularization on fully-connected layers of CNNs, *Neurocomputing*, **328** (2019), 69–74. https://doi.org/10.1016/j.neucom.2018.03.080
38. S. R. Dubey, S. K. Singh, B. B. Chaudhuri, Activation functions in deep learning: A comprehensive survey and benchmark, *Neurocomputing*, **503** (2022), 92–108. https://doi.org/10.1016/j.neucom.2022.06.111
39. S. Akbar, M. Peikari, S. Salama, S. Nofech-Mozes, A. Martel, The transition module: A method for preventing overfitting in convolutional neural networks, *Comput. Methods Biomech. Biomed. Eng.: Imaging Vis.*, **7** (2019), 260–265. https://doi.org/10.1080/21681163.2018.1427148
40. H. Wu, X. Gu, Towards dropout training for convolutional neural networks, *Neural Networks*, **71** (2015), 1–10. https://doi.org/10.1016/j.neunet.2015.07.007
41. M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, S. Mougiakakou, Lung pattern classification for interstitial lung diseases using a deep convolutional neural network, *IEEE T. Med. Imaging*, **35** (2016), 1207–1216. https://doi.org/10.1109/TMI.2016.2535865
42. J. Chen, Y. Shen, The effect of kernel size of CNNs for lung nodule classification, In: *2017 9th International Conference on Advanced Infocomm Technology (ICAIT)*, 2017, 340–344. https://doi.org/10.1109/ICAIT.2017.8388942
43. B. Chen, W. Deng, J. Du, Noisy softmax: Improving the generalization ability of DCNN via postponing the early softmax saturation, In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, 4021–4030. https://doi.org/10.1109/CVPR.2017.428
44. Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, In: *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, 2006.
45. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2016, 770–778. https://doi.org/10.1109/CVPR.2016.90
46. S. Han, J. Pool, J. Tran, W. J. Dally, Learning both weights and connections for efficient neural network, In: *NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems*, **1** (2015), 1135–1143.
47. P. Ochs, A. Dosovitskiy, T. Brox, T. Pock, On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision, *SIAM J. Imaging Sci.*, **8** (2015), 331–372. https://doi.org/10.1137/140971518
48. P. Murugan, S. Durairaj, Regularization and optimization strategies in deep convolutional neural network, 2017, arXiv:1712.04711.
49. J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, et al., Scalable Bayesian optimization using deep neural networks, In: *ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning*, **37** (2015), 2171–2180.

50. D. Cheng, Y. Gong, S. Zhou, J. Wang, N. Zheng, Person re-identification by multi-channel parts-based CNN with improved triplet loss function, In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, 1335–1344. https://doi.org/10.1109/CVPR.2016.149

51. Y. S. Aurelio, G. M. de Almeida, C. L. de Castro, A. P. Braga, Learning from imbalanced data sets with weighted cross-entropy function, *Neural Process Lett.*, **50** (2019), 1937–1949. https://doi.org/10.1007/s11063-018-09977-1

52. M. Bouten, J. Schietse, C. Van. den Broeck, Gradient descent learning in perceptrons: A review of its possibilities, *Phys. Rev. E*, **52** (1995), 1958–1967. https://doi.org/10.1103/PhysRevE.52.1958

53. A. El-Sawy, M. Loey, H. El-Bakry, Arabic handwritten characters recognition using convolutional neural network, *WSEAS Trans. Comput. Res.*, **5** (2017), 11–19.

54. Y. Sun, W. Zhang, H. Gu, C. Liu, S. Hong, W. Xu, et al., Convolutional neural network based models for improving super-resolution imaging, *IEEE Access*, **7** (2019), 43042–43051. https://doi.org/10.1109/ACCESS.2019.2908501

55. G. D. Rubin, C. J. Ryerson, L. B. Haramati, N. Sverzellati, J. P. Kanne, S. Raoof, et al., The role of chest imaging in patient management during the COVID-19 pandemic: A multinational consensus statement from the Fleischner society, *Radiology*, **296** (2020), 172–180. https://doi.org/10.1148/radiol.2020201365

56. B. Xu, Y. Xing, J. Peng, Z. Zheng, W. Tang, Y. Sun, et al., Chest CT for detecting COVID-19: A systematic review and meta-analysis of diagnostic accuracy, *Eur Radiol*, **30** (2020), 5720–5727. https://doi.org/10.1007/s00330-020-06934-2

57. A. M. Rahmani, E. Azhir, M. Naserbakht, M. Mohammadi, A. H. M. Aldalwie, M. K. Majeed, et al., Automatic COVID-19 detection mechanisms and approaches from medical images: A systematic review, *Multimed. Tools Appl.*, **81** (2022), 28779–28798. https://doi.org/10.1007/s11042-022-12952-7

58. E. E. Hemdan, M. A. Shouman, M. E. Karar, COVIDX-Net: A framework of deep learning classifiers to diagnose COVID-19 in X-ray images, 2020, arXiv:2003.11055.

59. M. Polsinelli, L. Cinque, G. Placidi, A light CNN for detecting COVID-19 from CT scans of the chest, *Pattern Recogn. Lett.*, **140** (2020), 95–100. https://doi.org/10.1016/j.patrec.2020.10.001

60. A. Narin, C. Kaya, Z. Pamuk, Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks, *Pattern Anal, Appl.*, **24** (2021), 1207–1220. https://doi.org/10.1007/s10044-021-00984-y

61. P. Mooney, Chest X-ray images (Pneumonia), 2018. Available from: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia.

62. T. Rahman, COVID-19 radiography database, 2020. Available from: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database.

63. I. D. Apostolopoulos, T. A. Mpesiana, Covid-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks, *Phys. Eng. Sci. Med.*, **43** (2020), 635–640. https://doi.org/10.1007/s13246-020-00865-4

64. P. K. Sethy, S. K. Behera, P. K. Ratha, P. Biswas, Detection of coronavirus disease (COVID-19) based on deep features and support vector machine, *Int. J. Math. Eng. Manage. Sci.*, **5** (2020), 643–651. https://doi.org/10.33889/IJMEMS.2020.5.4.052

65. Y. Wang, M. Hu, Q. Li, X. Zhang, G. Zhai, N. Yao, Abnormal respiratory patterns classifier may contribute to large-scale screening of people infected with COVID-19 in an accurate and unobtrusive manner, 2020, arXiv: 2002.05534.
66. J. Zhang, Y. Xie, G. Pang, Z. Liao, J. Verjans, W. Li, et al., Viral pneumonia screening on chest X-rays using confidence-aware anomaly detection, *IEEE T. Med. Imaging*, **40** (2021), 879–890. https://doi.org/10.1109/TMI.2020.3040950
67. P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, A. Mohammadi, COVID-CAPS: A capsule network-based framework for identification of COVID-19 cases from X-ray images, *Pattern Recogn. Lett.*, **138** (2020), 638–643. https://doi.org/10.1016/j.patrec.2020.09.010
68. M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, et al., Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access*, **8** (2020), 132665–132676. https://doi.org/10.1109/ACCESS.2020.3010287
69. L. O. Hall, R. Paul, D. B. Goldgof, G. M. Goldgof, Finding Covid-19 from chest X-rays using deep learning on a small dataset, 2020, arXiv: 2004.02060.
70. T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, U. R. Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, *Comput. Biol. Med.*, **121** (2020), 103792. https://doi.org/10.1016/j.compbiomed.2020.103792
71. R. M. Pereira, D. Bertolini, L. O. Teixeira, C. N. Silla, Y. M. G. Costa, COVID-19 identification in chest X-ray images on flat and hierarchical classification scenarios, *Comput. Meth. Prog. Bio.*, **194** (2020), 105532. https://doi.org/10.1016/j.cmpb.2020.105532
72. L. Mahdy, K. Ezzat, H. Elmousalami, H. Ella, A. Hassanien, Automatic X-ray COVID-19 lung image classification system based on multi-level thresholding and support vector machine, 2020, medRxiv preprint. https://doi.org/10.1101/2020.03.30.20047787
73. K. El Asnaoui, Y. Chawki, A. Idri, Automated methods for detection and classification pneumonia based on X-ray images using deep learning, In: *Artificial intelligence and blockchain for future cybersecurity applications*, Springer, Cham, 2021, 257–284. https://doi.org/10.1007/978-3-030-74575-2_14
74. D. Singh, V. Kumar, V. Kaur, M. Kaur, Classification of COVID-19 patients from chest CT images using multi-objective differential evolution–based convolutional neural networks, *Eur. J. Clin. Microbiol. Infect. Dis.*, **39** (2020), 1379–1389. https://doi.org/10.1007/s10096-020-03901-z
75. M. Yamac, M. Ahishali, A. Degerli, S. Kiranyaz, M. E. H. Chowdhury, M. Gabbouj, Convolutional sparse support estimator-based COVID-19 recognition from X-ray images, *IEEE T. Neur. Net. Lear. Syst.*, **32** (2021), 1810–1820. https://doi.org/10.1109/TNNLS.2021.3070467
76. U. Özkaya, Ş. Öztürk, M. Barstugan, Coronavirus (COVID-19) classification using deep features fusion and ranking technique, In: *Big Data Analytics and Artificial Intelligence Against COVID-19: Innovation Vision and Approach*, Springer, Cham, 2020, 281–295. https://doi.org/10.1007/978-3-030-55258-9_17
77. C. Salvatore, M. Interlenghi, C. Monti, D. Ippolito, D. Capra, A. Cozzi, et al., Artificial intelligence applied to chest X-ray for differential diagnosis of COVID-19 pneumonia, *Diagnostics*, **11** (2021), 530. https://doi.org/10.3390/diagnostics11030530
78. T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, S. Yang, P. W. Eklund, T. Huynh-The, et al., Artificial intelligence in the battle against coronavirus (COVID-19): A survey and future research directions, 2020, arXiv: 2008.07343.

79. E. Neri, V. Miele, F. Coppola, R. Grassi, Use of CT and artificial intelligence in suspected or COVID-19 positive patients: Statement of the Italian society of medical and interventional radiology, *La radiologia medica*, **125** (2020), 505–508. https://doi.org/10.1007/s11547-020-01197-9
80. L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, et al., Using artificial intelligence to detect COVID-19 and community-acquired pneumonia based on pulmonary CT: Evaluation of the diagnostic accuracy, *Radiology*, **296** (2020), E65–E71. https://doi.org/10.1148/radiol.2020200905
81. M. A. Amou, K. Xia, S. Kamhi, M. Mouhafid, A novel MRI diagnosis method for brain tumor classification based on CNN and Bayesian optimization, *Healthcare,* **10** (2022), 494. https://doi.org/10.3390/healthcare10030494
82. S. Z. Kurdi, M. H. Ali, M. M. Jaber, T. Saba, A. Rehman, R. Damaševičius, Brain tumor classification using meta-heuristic optimized convolutional neural networks, *J. Pers. Med*, **13** (2023), 181. https://doi.org/10.3390/jpm13020181
83. E. Irmak, Multi-classification of brain tumor MRI images using deep convolutional neural network with fully optimized framework, *Iran. J. Sci. Technol. Trans. Electr. Eng.*, **45** (2021), 1015–1036. https://doi.org/10.1007/s40998-021-00426-9
84. C. Venkatesh, K. Ramana, S. Y. Lakkisetty, S. S. Band, S. Agarwal, A. Mosavi, A neural network and optimization based lung cancer detection system in CT images, *Front. Public Health*, **10** (2022), 769692. https://doi.org/10.3389/fpubh.2022.769692
85. D. Paikaray, A. K. Mehta, D. A. Khan, Optimized convolutional neural network for the classification of lung cancer, *J. Supercomput.*, **80** (2024), 1973–1989. https://doi.org/10.1007/s11227-023-05550-3
86. C. Lin, S. Jeng, M. Chen, Using 2D CNN with Taguchi parametric optimization for lung cancer recognition from CT images, *Appl. Sci.*, **10** (2020), 2591. https://doi.org/10.3390/app10072591