Available at www.sciencedirect.com
ScienceDirect
journal homepage: www.elsevier.com/locate/bbe

Original Research Article

Detection of pneumonia using convolutional neural networks and deep learning

Patrik Szepesi [a], *László Szilágyi* [b,c,d,*]

[a] Corvinus University of Budapest, Budapest, Hungary
[b] Computational Intelligence Research Group (CIRG), Sapientia University of Transylvania, Tîrgu Mureş, Romania
[c] Physiological Controls Research Center, Óbuda University, Budapest, Hungary
[d] Biomatics and Applied Artificial Intelligence Institution, Óbuda University, Budapest, Hungary

ARTICLE INFO

Article history:
Received 6 February 2022
Received in revised form
25 July 2022
Accepted 7 August 2022
Available online 19 August 2022

Keywords:
Convolutional neural network
Deep learning
Transfer learning
Pneumonia detection
Medical imaging

ABSTRACT

The objective and automated detection of pneumonia represents a serious challenge in medical imaging, because the signs of the illness are not obvious in CT or X-ray scans. Further on, it is also an important task, since millions of people die of pneumonia every year. The main goal of this paper is to propose a solution for the above mentioned problem, using a novel deep neural network architecture. The proposed novelty consists in the use of dropout in the convolutional part of the network. The proposed method was trained and tested on a set of 5856 labeled images available at one of Kaggle's many medical imaging challenges. The chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients, aged between one and five years, from Guangzhou Women and Children's Medical Center, Guangzhou, China. Results achieved by our network would have placed first in the Kaggle competition with the following metrics: 97.2% accuracy, 97.3% recall, 97.4% precision and AUC = 0.982, and they are competitive with current state-of-the-art solutions.

1. Introduction

Computer vision is a relevant application field of artificial intelligence (AI), which comes as no surprise, mainly because it offers solutions to a vast amount of problems that we humans face in the 21st century. One of the fields of computer vision that has proved to be fruitful again and again is medical image diagnosis with the aid of AI.

Pneumonia represents a highly ranked cause of death with over 4 million fatalities yearly [1]. Pneumonia has many variants, it can be viral, bacterial, or fungal, and each of these variants may have various classes. Pneumonia is often diagnosed via chest X-ray (CXR) or computed tomography (CT) scans, the former being the most commonly used technique due to its reduced costs and availability in all countries. The computer can help the human expert in establishing the

* Corresponding author at: Computational Intelligence Research Group (CIRG), Sapientia University of Transylvania, Şoseaua Sighişoarei 1/C, 540485 Tîrgu Mureş, Romania.
E-mail addresses: patrikszepesi1@gmail.com (P. Szepesi), lalo@ms.sapientia.ro, szilagyi.laszlo@uni-obuda.hu (L. Szilágyi).
https://doi.org/10.1016/j.bbe.2022.08.001
0168-8227/

diagnosis of pneumonia due to its high speed and objective reproducible judgement, as the signs of pneumonia in X-ray images are not always visible or legible for the human eye. Several computer aided solutions have been provided for the detection and categorization of pneumonia. The COVID-19 pandemic caused a new wave in the development of such diagnosis procedures. In the following, we are presenting some of the recent successful solutions, mainly based on convolutional neural networks (CNN) and deep learning. Jain et al. [2] deployed VGG-16, VGG-19, Resnet50, and Inception-V3 network architectures and transfer learning for the detection of viral pneumonia, obtaining 71–88% accuracy on 624 test images. Dey et al. [3] combined the VGG-19 architecture with various classifiers like SVM or random forest, and based on a set of 1650 X-ray images they achieved accuracy, precision, and F_1 score values ranging up to 98%, 95%, and 96%, respectively. Brunese et al. [4] established a three-class pneumonia detection framework based on VGG-16 architecture and the so-called GradCAM algorithm for visual debugging, and achieved a 96.2% accuracy on a set of 6523 CXR images. Panwar et al. [5] combined the VGG-19 architecture with the GradCAM algorithm, and achieved 95.6% accuracy in a three-class pneumonia classification framework. Similarly, Ibrahim et al. [6] proposed a solution for three-class pneumonia detection, using VGG and ResNet152V2 architectures, and obtained recall values of 93–97% based on an own collection of X-ray images. Jin et al. [7] proposed a three-stage hybrid model consisting of feature extractor, feature selector, and SVM classifier stage, which achieved a remarkable 98.62% accuracy on a private set of 1743 X-ray images. Karthik et al. [8] employed the so-called Channel-Shuffled Dual-Branched (CSDB) CNN architecture to distinguish various types of pneumonia from several publicly available datasets, achieving F_1 scores of 94–98%. Quan et al. [9] combined the DenseNet and the CapsNet architecture and achieved 96% recall on a small sized COVID-19 dataset, but only 90.7% accuracy on larger set of pneumonia X-ray images. Alhudhaif et al. [10] deployed a DenseNet-201 based architecture and achieved accuracy and recall values up to 95% and almost 90% precision on Kaggle's dataset of over 6000 X-ray images. Sirazitdinov et al. [11] used the 26,684 CXR images of the Kaggle Pneumonia Detection Challenge to train an ensemble of two CNNs, namely a RetinaNet and a Mask R-CNN, but they achieved only a recall value of 80%. The Mask R-CNN architecture was also deployed by Jaiswal et al. [12] to localize regions of the lung affected by pneumonia. Mahmud et al. [13] proposed a so-called CovXNet architecture, a CNN that uses depthwise convolution with varying dilation rates to extract a wide range of features from X-ray images, which was trained and validated on 6161 CXR scans, achieving accuracy of 90.2% in a four-class and 97.4% in a two-class pneumonia classification problem. Ouchicha et al. [14] proposed a residual CNN architecture and trained it on 2905 X-ray images, reaching a 96.7% overall accuracy in three-class pneumonia classification. Wang et al. [15] designed a so-called prior-attention residual learning block and combined it with two 3D-ResNets and obtained 93.3% accuracy in the three-class pneumonia detection problem, based on 4697 X-ray images. Probably the largest available CXR image dataset, the one published by RSNA Pneumonia Detection Challenge consisting of over 120,000 items, was used by Wang et al. [16] to train a VGG architecture. They obtained an accuracy of 94.62% in the detection of pneumonia. Wang et al. [17] deployed a so-called deep stacked sparse autoencoder network model to provide a four-class classification of breast CT images based on two-dimensional fractional Fourier entropy features, and obtained a 92.3% averaged F_1 score. Wang et al. [18] deployed a deep rank-based average pooling network that combined an n-conv rank-based average pooling module with a CNN model inspired by the VGG network, reaching an average F_1 score of 95.5% in a study elaborated on 1164 CT scans of 521 subjects. Horry et al. [19] performed a comparative study of imaging modalities within the pneumonia detection framework and found that pneumonia can be best detected from ultrasound scans. Rajaraman et al. [20] presented a framework in which various CNN models deployed through transfer learning were involved in ensemble learning. Their ensembles were trained using 16,700 CXR images originating from 4 public databases, and achieved various accuracy rates between 94% and 99%. Singh and Yow [21] proposed an interpretable deep learning neural network approach based on two existing models, ProtoPNet and NP-ProtoPNet, and achieved an accuracy of 87.27%. For further details of the current state-of-the-art, the reader is referred to recent review papers, e.g. [1,22–25].

Most of the above presented solutions deploy transfer learning, meaning that the networks were previously trained on images not related to pneumonia. There are several machine vision applications that use CNNs built from scratch (e.g. [26]), which proved that a smaller architecture may obtain finer accuracy than several pretrained larger models deployed via transfer learning. In this study we propose to build a novel CNN architecture to provide an accurate solution to the problem of pneumonia detection. The main novelty consists in the fact that our CNN model uses dropout in the convolutional part of the network, unlike the majority of existing architectures, which use dropout exclusively in the fully connected part of the network where the main part of the parameters is trained. This paper proves that the proposed model can provide accurate classification despite the reduced number of trained parameters and can even benefit from this property in terms of efficiency.

The rest of this paper is structured as follows: Section 2 gives a detailed presentation of the proposed methodology. Section 3 presents the evaluation of the proposed solution. Section 4 discusses the achieved results in comparison with the state-of-the-art. Section 5 concludes this study.

2. Materials and Methods

2.1. Data

The dataset used for this study originates from one of Kaggle's many deep learning competitions.[1] The dataset depicts lung X-ray images of infants aged one through five. The lung

[1] https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

X-ray images taken from Guangzhou Women and Children's Medical Center were verified by medical experts. All chest X-ray imaging were performed as part of patients' routine clinical care. The dataset consisted of 5856 labeled images, of which 4273 showed pneumonia, while the remaining 1583 were negatives. Due to the imbalance in the dataset, a generative adversarial network was used to generate further images for the minority class (which were used exclusively during training). No generated image was used for the evaluation of the algorithm. All scans were single-channel intensity images and their dimensions varied from 1346×1044 to 2090×1858 pixels. All images were transformed to the $224 \times 224 \times 3$ format to comply with the expected input of most CNN network architectures. Fig. 1 exhibits some example images together with their ground truth.

2.2. *Foundations*

In this paper we propose to create a convolutional neural network that is constructed without the help of transfer learning or pretrained weights. Unlike the majority of CNN architectures described in the literature, out network uses a dropout layer carefully placed in the convolutional part of the network, which greatly diverges from the common placement of dropout layers only in the fully connected dense part of the network [27].

But what are dropout layers, and why are they mainly found in fully connected layers? Adding dropout layers to a convolutional neural network is one of many regularization techniques commonly used to prevent the network from overfitting. It is worth noting that dropout layers are only used during the training phase. Overfitting is a term used to describe a network's behavior that has learned the details of the training data too well and fails to generalize on the test set. It is a byproduct of the model, which learns the noise of the training data altogether with the useful information. Dropout layers prevent the model from relying on only a few important weights or features. This is achieved by randomly disregarding neurons in each training epoch, which ensures that the model does not become too dependent on one, or just a few single parameters. In case of employing dropout, each neuron will be used with a probability of $1 - p$, where p is the dropout rate, an input parameter to the dropout layer. The dropout regularization technique has enabled data scientists to create larger (with more hidden layers) and more accurate models [27]. The diagrams exhibited in Fig. 2 demonstrate how dropout works in a simple feed forward neural network: in this example we set the dropout rate for each node in its layer to 0.5. When this dropout rate is used, there is a 50% chance for each node in that layer to be ignored in that certain training epoch. The dropout rate is likely to be chosen between 0 and 50%.

Dropout layers are almost always found exclusively in dense layers, because dense layers have more parameters than convolutional layers. So dense layers are more prone to overfit. In this study we place the dropout layer in the convolutional part of the CNN model and investigate its behavior in comparison with the CNN model with no dropout and further existing CNN models deployed via transfer learning.

2.3. *The process of learning*

As described previously, the images fed to the network models are RGB images, previously resized to the following dimensions: $224 \times 224 \times 3$. Upon resizing the images, they are normalized by dividing each of the pixel values across the three channels by 255 in order to make the calculations faster and more effective. It is important to normalize images so that the data have a similar distribution which yields faster convergence during the training. Normalized images are then

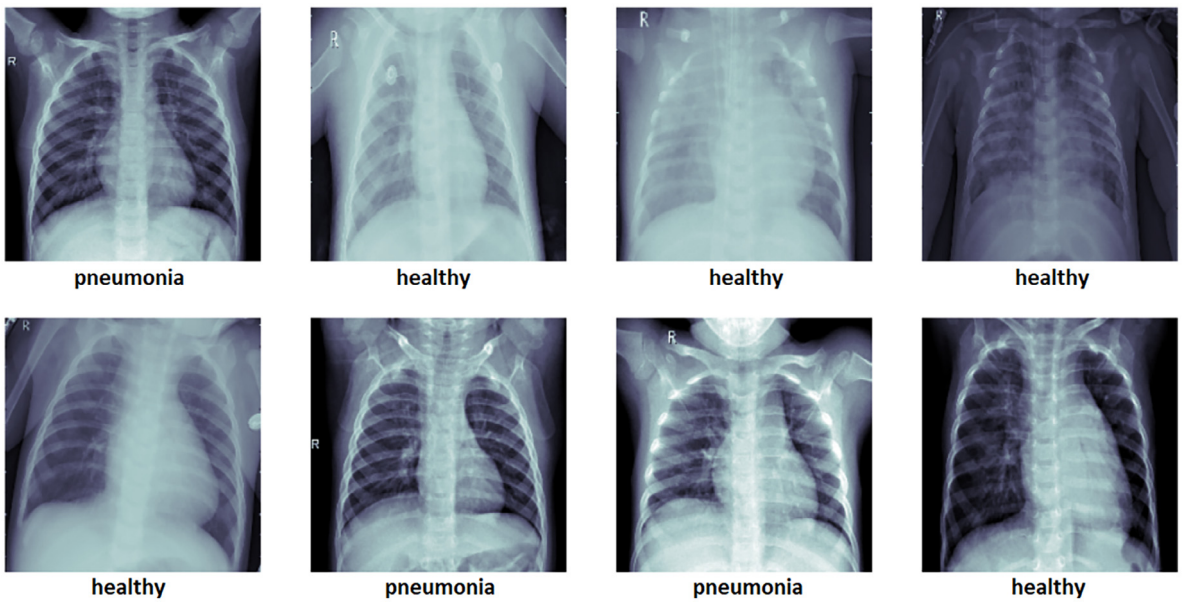

Fig. 1 – Some examples of input images with their ground truth. The eyes of a non-expert human can hardly distinguish positive and negative X-ray scans.

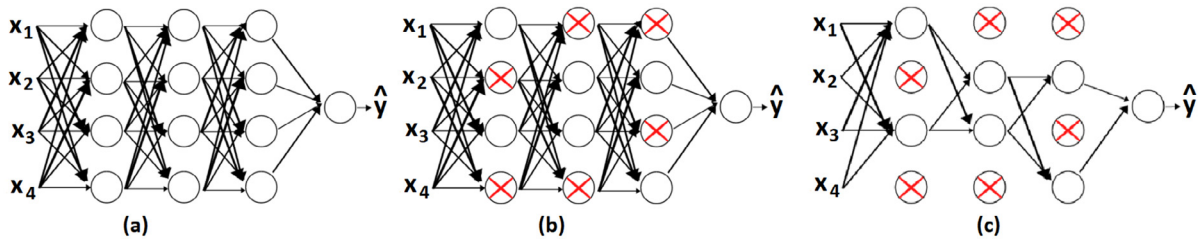

Fig. 2 – Effect of dropout: (a) normal feed forward network; (b) randomly selected nodes to be dropped at 50% dropout rate; (c) all connections of dropped nodes eliminated.

split into training and testing images; 80% of the images (4685) are training images and the remaining 20% (1171) images are placed into the testing set. The images were split in a way that retains the same proportion of classes in the train and test sets that are found in the entire original dataset, meaning that the proportion of healthy and unhealthy lungs in the training and testing dataset are identical. Before discussing the architecture of the network, it is important to have the right configurations for training, in order to make sure that the model converges to the global minimum (minimize loss function) and in the meantime it does not take an extensive time to train.

Two mechanisms were implemented in order to allow for the best possible convergence and lowest required training time. The first mechanism called *reduce learning rate on plateau* makes sure that when the learning begins to stagnate the learning rate is reduced by some arbitrary factor. This is done so that gradient descent would not overshoot the global minimum. Some of the parameters for this algorithm in our model included patience which was set to 5. This parameter tells the algorithm to decrease the learning rate after 5 epochs if no improvement was achieved in the given metric (accuracy in our case). Another key parameter is the factor by which the learning rate is multiplied for reduction, in our case this factor was set to 0.8. The final essential parameter to mention is cooldown, set to 5 for our algorithm, which represents the wait time expressed in epochs, after which the algorithm can start checking again if a learning rate reduction is needed. As earlier stated, the primary objective of the algorithm is to prevent the model from overshooting the global minimum.

The other mechanism, named *early stopping*, monitors the given metric, in our case the validation loss, and when that metric no longer improves over a specified interval, the training is stopped. The provided setting can immensely speed up the training process, because it is possible that a model is given 100 epochs to train but it has already converged and found the global minimum at epoch 30. Meaning that the remaining 70 epochs could actually cause negative side effects to the model and even lead to overfitting. This is why the early stopping mechanism is crucial. Our model was given 150 epochs to train, but it actually converged after 40 epochs. This saved us a lot of time and has also averted the model from overfitting.

The optimization algorithm, the batch size, the learning rate, as well as the loss function are all essential parts of a model. In our model we decided to use a learning rate of 10^{-4}, which was just the result of a hyperparameter tuning job. It is worth noting that a large learning rate can cause a convergence to a local minimum instead of the global one, and a small learning rate can result in slow convergence; neither is ideal. It was experimentally established that our algorithm performs the best when a mini batch of size 16 is used. Batch size is simply the number of samples processed before the model is updated by the chosen optimization algorithm. Some noise is good and is needed for a non-convex optimization, because that noise helps the algorithm overcome saddle points and local minima. This is why we ended up using mini batch gradient descent with a batch size of 16.

The optimization algorithm that fit our needs the best was the Adam optimizer (Adaptive Moment Estimation), which is an upgraded version of gradient descent, as it combines RMSProp with momentum to boost the performance of the optimizer. The Adam optimizer dynamically updates its learning rate for each parameter, combined with momentum, which helps the algorithm prevent getting stuck in local minima [28].

2.4. *Model architecture*

Our new model slightly resembles the VGG-16 architecture in that some of its early layers follow a similar structure of convolutional layer, batch normalization layer, activation layer, which is finally followed by a pooling layer. However our network could not be more different from the VGG's architecture which performed poorly on the given dataset. As previously described, the unique integration of the dropout is located in a convolutional layer. As depicted in Fig. 3 and Table 1, the model follows the pattern of convolutional layer followed by pooling layer. This pattern is repeated several times, because it yet again proves the power of the convolutional layers and its ability to learn specific features, which are often referred to as feature maps when many of them are stacked together [29]. Filters are essentially the elements that enable the neural network to pick up on specific patterns such as edges. Filters are often initialized in many ways but ultimately that can be omitted, because the purpose of the convolutional layer is to learn the specific features during the training; they are basically the mechanism needed for feature extraction. At last the matrices are flattened and are then connected to the dense fully connected layers, which are responsible for the classification. It is worth noting that within each convolutional block in the model's architecture there is a batch normalization layer followed by a ReLU activation function.

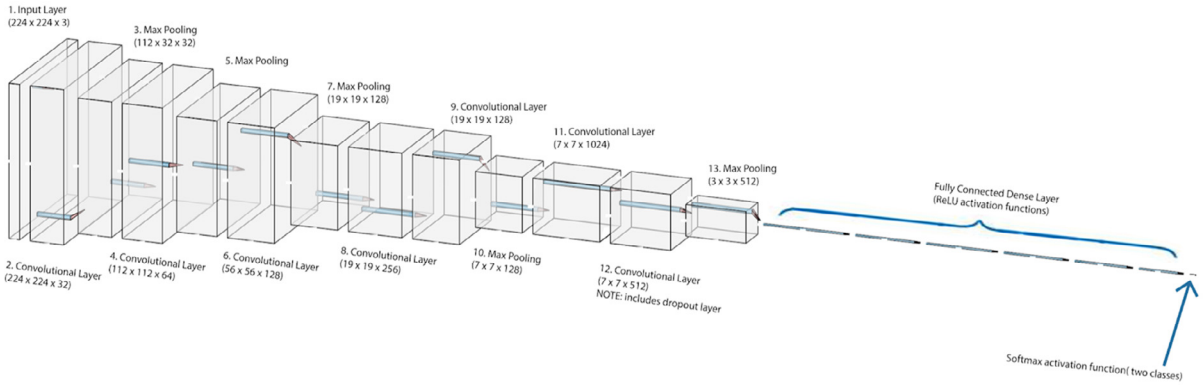

Fig. 3 – The structure of the proposed network model.

Batch normalization receives as input the output of the convolutional layer. First it applies a standard normalization to assure zero mean and unit variance, and then it performs a scaling by a term γ and shifting by another term β. This result is then passed to the activation function, which finally provides the input for the next convolutional layer [30]. These terms γ and β are learned during network training via back-propagation to find the best values for the given task.

Pooling layers usually follow convolutional layers in order to reduce the number of parameters and dimensions of the feature maps. This is performed so that the algorithm could learn faster and become more efficient by downsampling features it has detected. Overall the main concept behind pooling layers is that they summarize the key points of the feature map, by applying some kind of pooling to it. In our case max pooling was applied, with both the strides and filter size equaling to 3×3.

The output of the convolutional layers is data that represents the features learned from the layer. This data is stored in a matrix which at this point gets flattened out so neurons could have full connections to all activations in the previous layer. This allows the network to learn the combinations of these features (in a non-linear way) to make the best possible prediction.

The activation function in all of our layers was the well known ReLU, except for the last one. The last activation function, which ultimately decides whether the lung in the CXR image had pneumonia or not was the SoftMax activation function, which predicts probabilities in the last layer.

The implementation of the proposed network model is available at: https://github.com/patrikszepesi/pneumonia/blob/main/CMIG-paper.ipynb.

2.5. *Evaluation criteria*

Classification problems, like the one addressed in this paper are usually evaluated with statistical means. During the evaluation process, predictions are made for the testing data, and the results are collected in the confusion matrix, which reflects the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Based on these four counts, several statistical indicators can be extracted, which characterise the accuracy of the prediction. The statistical indicators involved in this study are:

1. Recall (or sensitivity or true positive rate, TPR) shows the rate of positive cases correctly predicted and it is computed as $\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$. This is a key metric in medical diagnosis because a high recall can be translated to few false negatives, which is a strict requirement in medical systems.
2. Precision (or positive predictive value, PPV) indicates the rate of positives in the set of cases declared positive by the classifier. It is computed as $\mathrm{PPV} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$.
3. F_1 score is an indicator that equally penalizes false positives and false negatives, and is defined as the harmonic mean of TPR and PPV: $F_1 = 2 \times \mathrm{TPR} \times \mathrm{PPV}/(\mathrm{TPR} + \mathrm{PPV})$.
4. Accuracy (ACC) represents the rate of correct decisions, namely $\mathrm{ACC} = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})$.

All the above statistical indicators are valued between 0 and 1, where the overall value of 1 representing perfect classification. Their values are usually expressed in percentage.

Further on, the receiver operator characteristic (ROC curve) of the investigated models was established by varying the threshold at the output layer of the networks. The area under ROC curve (AUC) was also extracted to provide more reliable comparison of the performance of the network models.

3. Results

As it was earlier stated, 80% of the available CXR images were used as training data (70% for the actual training and 10% for validation), and the remaining 20% as testing data. For each CNN architecture involved in this study, a tenfold cross-validation was performed. The obtained benchmarks are presented in the following.

Table 2 presents the average value and standard deviation of the main statistical indicators obtained by the proposed CNN architecture with and without dropout in the convolutional part, in comparison with three existing popular architectures deployed via transfer learning. Fig. 4 exhibits the average values of the main indicators separately for each network model. The proposed architecture with dropout in its

Table 1 – The full list of layers of our proposed network architecture.

Layer (type)	Output shape	Parameters
(InputLayer)	[(224, 224, 3)]	0
(Conv2D)	(224, 224, 32)	864
(Batch Normalization)	(224, 224, 32)	128
(Activation)	(224, 224, 32)	0
(Conv2D)	(224, 224, 32)	9216
(Batch Normalization)	(224, 224, 32)	128
(Activation)	(224, 224, 32)	0
(MaxPooling2D)	(112, 112, 32)	0
(Conv2D)	(112, 112, 64)	18432
(Batch Normalization)	(112, 112, 64)	256
(Activation)	(112, 112, 64)	0
(Conv2D)	(112, 112, 64)	36864
(Batch Normalization)	(112, 112, 64)	256
(Activation)	(112, 112, 64)	0
(MaxPooling2D)	(56, 56, 64)	0
(Conv2D)	(56, 56, 128)	73728
(Batch Normalization)	(56, 56, 128)	512
(Activation)	(56, 56, 128)	0
(Conv2D)	(56, 56, 128)	147456
(Batch Normalization)	(56, 56, 128)	512
(Activation)	(56, 56, 128)	0
(MaxPooling2D)	(19, 19, 128)	0
(Conv2D)	(19, 19, 256)	294912
(Batch Normalization)	(19, 19, 256)	1024
(Activation)	(19, 19, 256)	0
(Conv2D)	(19, 19, 128)	294912
(Batch Normalization)	(19, 19, 128)	512
(Activation)	(19, 19, 128)	0
(MaxPooling2D)	(7, 7, 128)	0
(Conv2D)	(7, 7, 1024)	1179648
(Batch Normalization)	(7, 7, 1024)	4096
(Activation)	(7, 7, 1024)	0
(Conv2D)	(7, 7, 512)	4718592
(Dropout)	(7, 7, 512)	0
(Batch Normalization)	(7, 7, 512)	2048
(Activation)	(7, 7, 512)	0
(MaxPooling2D)	(3, 3, 512)	0
(Flatten)	(1, 4608)	0
(Dense)	(1, 512)	2359808
(Dense)	(1, 1024)	525312
(Dense)	(1, 512)	524800
(Dense)	(1, 512)	262656
(Dense)	(1, 256)	131328
(Dense)	(1, 64)	16448
predictions (Dense)	(1, 2)	130

convolutional part obviously outperforms all other tested models in terms of accuracy, according to all four statistical indicators, achieving recall, precision, F_1 score and accuracy well above 97%, while the counter candidates obtain values up to 95.5%. There is almost 2% difference in all statistical indicators if we compare the two proposed architectures, in the favor of the one that uses dropout in the convolutional part. The standard deviation values of accuracy indicators suggest that the proposed model is stable, all the tests led to accurate classification. Table 2 also contains a row that presents the AUC values obtained by the investigated network models during the ROC analysis, in average ± standard deviation format. Fig. 5 exhibits the ROC curves obtained by the network models which correspond to the median AUC value for each network. Also here it is clearly visible, that the proposed network model with dropout in the convolutional part leads to more accurate classification of CXR images. Further on, mainly because of its lower number of trainable parameters, the proposed model is more efficient than its counter candidates, needing only 122 ms in average to produce a prediction. The inference time is same for the models with and without dropout, because dropout is only applied during training. The architectures deployed via transfer learning require up to 3 times more computations for an average inference. It is necessary to remark that the VGG-19 architecture was omitted in Fig. 4 to emphasize the differences among better performing models.

A series of tests were dedicated to establish the dropout rate that leads to the best performance of the proposed network model. Results are exhibited in Table 3 and Fig. 6, which show the accuracy indicator values obtained by the proposed CNN architecture at dropout rates varying between 0% and 50%. The best performance was obtained when the dropout rate was set to 40%, in terms of all statistical indicators. Dropout rates up to 20% did not significantly improve the performance of the proposed network in comparison to the model that uses no dropout, while dropout rate approaching 50% proved to be an obstacle of accurate classification. Also here it is visible that a well tuned dropout in the convolutional part of the network can improve the accuracy indicators by up to 2%. The AUC values obtained via ROC analysis also reveal the best performance of the proposed network used at dropout rate of 40%.

Fig. 7 depicts the evolution of the classification accuracy measured on the validation data, during the epochs of the training process. Here the comparison is made between the architectures with and without dropout in the convolutional part of the network. The version with dropout reaches an accuracy that is higher by 2%, achieves it in less epochs, and it keeps on being more stable if the training is not stopped. Table 4 shows the local maxima of the validation accuracy achieved by the two proposed network models during a typical run of the training process. The use of dropout in the convolutional part of the network leads to better accuracy and quicker convergence.

The proposed pneumonia detection model was implemented using Python and the Tensorflow(2.x) framework. Tensorflow provides a vast amount of pre-built functions that makes it possible for researchers to experiment with various model architectures without having to spend too much time on negligible details. Tensorflow enables researchers to quickly deploy models with ease of use, thus making the production implementation of such models facile. The training of the model was conducted with an Nvidia Tesla K80 GPU with 24 GB of memory. Data cleansing, analysis and training was performed in Jupyter notebook environment provided by AWS SageMaker.

Table 2 – Comparison of network architectures involved in this study, and their obtained benchmark values during testing. Accuracy, recall, precision, F_1 score, and AUC are presented as average value ± standard deviation.

Network model	Proposed model		Transfer learning		
	without dropout	with dropout	InceptionV3	ResNet50	VGG-19
Parameters	10.6 M	10.6 M	26.2 M	24.8 M	145.2 M
Accuracy (%)	95.67 ± 1.50	97.21 ± 1.13	90.94 ± 1.72	89.06 ± 1.64	61.19 ± 1.13
Recall (%)	95.54 ± 1.95	97.34 ± 1.56	89.10 ± 1.55	86.23 ± 2.31	61.88 ± 1.61
Precision (%)	95.50 ± 1.22	97.40 ± 1.21	91.89 ± 1.04	91.43 ± 1.59	62.33 ± 0.88
F_1 score (%)	95.52 ± 1.46	97.37 ± 1.32	90.47 ± 1.24	88.75 ± 1.88	62.10 ± 1.14
AUC	0.970 ± 0.005	0.982 ± 0.006	0.936 ± 0.010	0.921 ± 0.008	0.682 ± 0.016
Training time (50 epochs)	2304 s	2433 s	6247 s	5750 s	6577 s
Single inference time	122 ms	122 ms	307 ms	298 ms	313 ms

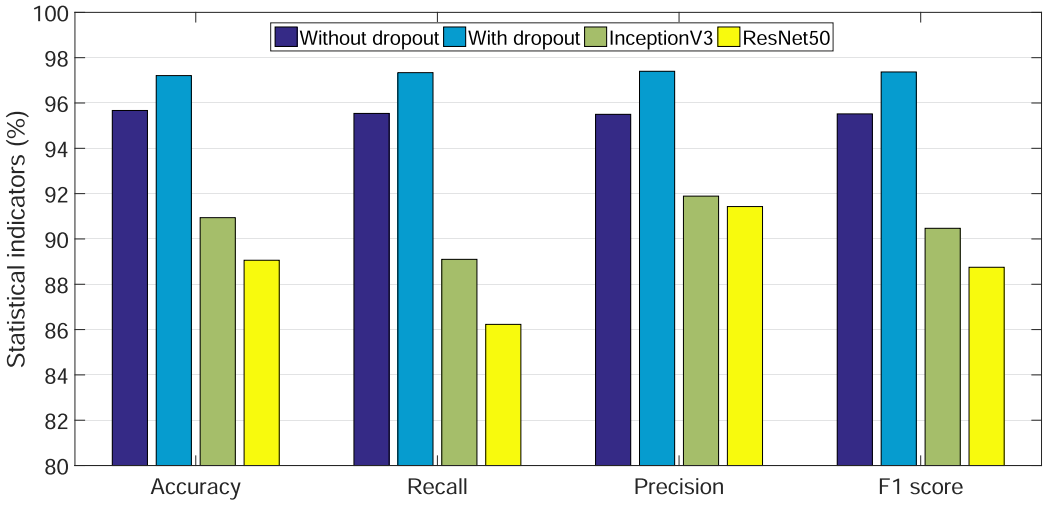

Fig. 4 – Comparison of average accuracy, recall, precision, and F1 score values achieved by various tested network models.

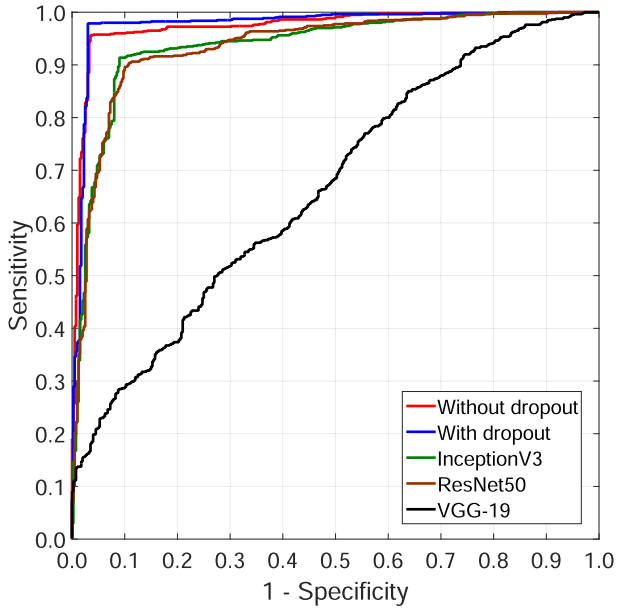

Fig. 5 – ROC curves obtained for various tested network models.

4. Discussion

The main finding of this study is the beneficial effect of the dropout layer placed in the convolutional part of a CNN architecture. Beside providing quicker convergence at training, it produces better accuracy at validation and prediction. The application of the proposed network model in infant pneumonia detection based on CXR images represents another proof that smaller CNN architectures built from scratch can easily outperform larger models deployed via transfer learning. Detailed explanation is given in the following paragraphs.

The tables and figures presented in Section 3 compared the performance of the proposed CNN model with some existing architectures under very same conditions. We found the newly introduced network superior in terms of both accuracy and efficiency. Table 5 gives us the opportunity to compare the performance of our architecture with several further recent solutions, which all belong to the current state-of-the-art. The network models listed here were tested by their authors under various circumstances, so simply comparing the reported accuracy percentages is not always objective. For example, larger datasets may contain a wider variety of patterns that is more difficult to handle with the same

Table 3 – Test performance of the proposed model at various dropout rates. All indicators are presented as average value ± standard deviation.

Dropout rate	No dropout	10%	20%	30%	40%	50%
Accuracy (%)	95.67 ± 1.50	95.88 ± 0.95	96.05 ± 1.44	96.54 ± 1.39	97.21 ± 1.13	95.52 ± 1.12
Recall (%)	95.54 ± 1.95	95.70 ± 1.21	95.55 ± 1.89	96.14 ± 1.16	97.34 ± 1.56	96.12 ± 0.96
Precision (%)	95.50 ± 1.22	95.48 ± 1.78	95.60 ± 1.52	96.43 ± 1.40	97.40 ± 1.21	95.64 ± 1.19
F_1 score (%)	95.52 ± 1.46	95.59 ± 1.44	95.57 ± 1.68	96.28 ± 1.27	97.37 ± 1.32	95.88 ± 1.06
AUC	0.970 ± 0.005	0.972 ± 0.003	0.974 ± 0.004	0.976 ± 0.004	0.982 ± 0.006	0.971 ± 0.005

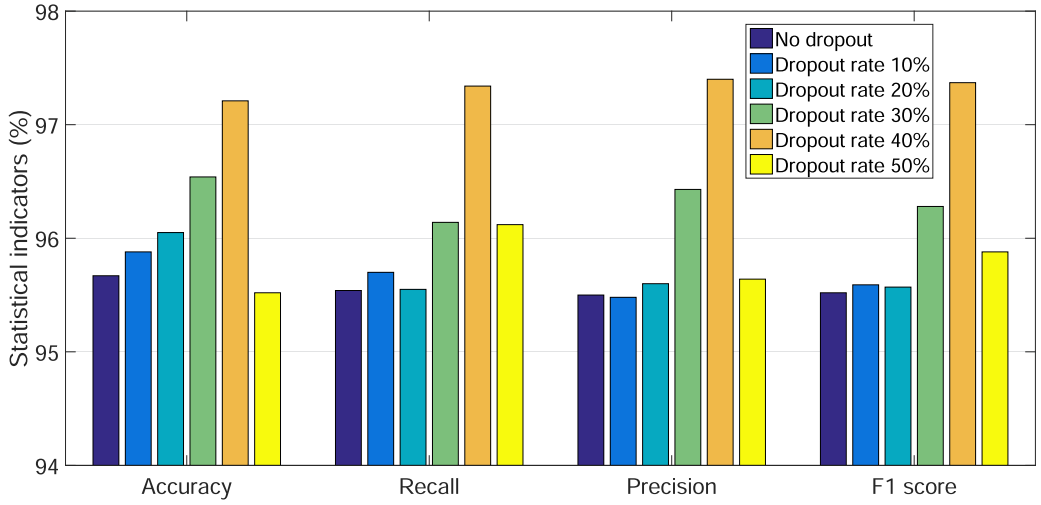

Fig. 6 – Comparison of the proposed network's performance at various dropout rates. Average values of the accuracy indictors are exhibited.

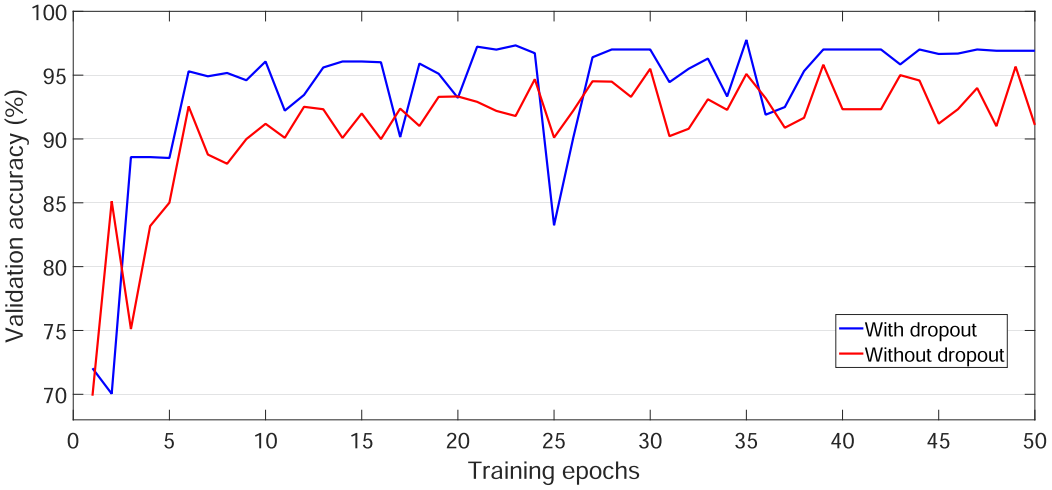

Fig. 7 – The evolution of validation accuracy during training, with and without dropout.

accuracy. According to the reported accuracy indicators, our CNN model that uses dropout in the convolutional layers, is very competitive among its peers. Further on, the decisions made by these methods are dedicated to distinguish a number of classes varying between two and four, depending on whether they only distinguish pneumonia and healthy lungs, or they attempt to categorize various types of pneumonia. The highest reported F_1 scores range between 97% and 98%, and this is the interval where the accuracy indicator of our proposed network model is situated.

Regarding the performance against the clock, it is necessary to remark that not too many of the listed methods report efficiency benchmarks, but those which do, almost all say that the duration of a single inference at testing lasts one or

Table 4 – Cardinal points of the validation accuracy, expressed in percentages, during the training of the proposed model with and without the use of dropout layers.

Epoch	1	2	3	6	10	19	21	23	24	30	35	39
With dropout	72.04%		88.58%	95.30%	96.07%		97.23%	97.33%			97.76%	
Without dropout	69.89%	85.12%		92.55%		93.30%			94.68%	95.50%		95.82%

Table 5 – Comparison with state-of-the-art methods from the literature.

Paper	Year	Method	Data	F_1 score	Runtime
Brunese et al. [4]	2020	VGG-16	6523 CXR	97%	2.5 s
Panwar et al. [5]	2020	VGG-19 + GradCAM	2482 CT + 6382 CXR	95.61%	2 s
Mahmud et al. [13]	2020	customized CNN (CovXNet)	6161 CXR	97.4%	N/A
Ouchicha et al. [14]	2020	customized CNN (CVDNet)	2905 CXR	96.7%	N/A
Wang et al. [15]	2020	3D-ResNet	4697 CXR	93.3%	N/A
Choudhury et al.[31]	2020	DenseNet201	3487 CXR	97.94%	N/A
Ren et al. [32]	2020	CNN + Bayesian Network	35,389 CXR	87%	N/A
Arias et al. [33]	2020	CNN	79,500 CXR	91.5%	N/A
Sakib et al. [34]	2020	customized CNN (DL-CRC)	5367 CXR	94%	N/A
Ozturk et al. [35]	2020	YOLO via DarkNet	1000 CXR	87–98%	< 1 sec
Alhudjaif et al. [10]	2021	DenseNet-201	1218 CXR	94.96%	"within seconds"
Nikolaou et al. [36]	2021	EfficientNet models	15,153 CXR	95%	N/A
Das et al. [37]	2021	CNN + transfer learning	1004 CXR	95%	"few seconds"
Munusamy et al. [38]	2021	FractalCovNet	473 CT + 11,934 CXR	92–98%	N/A
Joshi et al. [39]	2021	DarkNet-53	6884 CXR	97.11%	0.137 s
Singh and Tripathi [40]	2022	Quaternion CNN	5856 CXR	93.75%	N/A
Dash and Mohapatra [41]	2022	CNN + transfer learning	1272 CXR	97.12%	N/A
Gour and Jain [42]	2022	VGG-19, Xception	4645 CT + 3040 CXR	97.5%	0.029–3.66 s
Proposed method	2022	CNN + modified dropout	5856 CXR	97.4%	0.122 s

more seconds. Our network model, with its 122-ms inference time can be called highly efficient, it is competitive with the quickest counter candidate models [39,42].

Some existing CNN architectures were involved in the evaluation process to obtain objective comparison. In the final evaluation phase of our model, it was decided to compare the accuracy, recall and precision of our model to some of the most advanced models in the industry, namely the ResNet50, InceptionV3 model and VGG-19. In order to get reliable results all the models that were trained with transfer learning were given the same preprocessed input as our own model, and the training process was exactly the same, no parameter was changed. Our model was able to outperform every other model that we have tested, in all metrics: accuracy, precision, recall, F_1 score, and AUC.

Table 2 clearly depicts the efficient performance of our network compared to other models. All models were trained for 50 epochs. The total training time of our model was significantly less than all the other models. The small size of our proposed model greatly contributed to its fast training speed. The superiority of the proposed model is also reflected in the average duration of a single inference. So in conclusion our model not only produced finer decisions and better metrics, it also performed in notably less time.

When taking into consideration the size of the models with regards to the number of parameters, our model is significantly more effective than all other models involved in this study. While our model has a little over 10 million parameters, the ResNet50 has around 25 million parameters, the InceptionV3 has over 26 million parameters and the VGG-19 has 144 million parameters. When taking these into account, one quickly realizes why the time needed to train our model is remarkably less, and the duration of a single inference is considerably reduced.

The main limitation of this study is the fact that it relies on a single source of X-ray images. However, according to the number of images used for training and testing, our study would be at the middle of the ranking among the solutions listed in Table 5. Another limitation of the proposed method is that it only works with CXR data and does not use CT images. However, this can also be considered a useful feature, since X-ray imaging is much more widely available than computed tomography.

All pneumonia detection models listed in Table 5, and our proposed model as well, were trained in a retrospective manner using past cases taken from one or more publicly available databases or own collections of CXR images. Such datasets can hardly cover a representative population that would correspond to the mass of patients at any and every hospital of the world. Therefore, establishing a clinical application requires the possibility to periodically retrain the decision-making algorithm with current cases of the local population. In clinical practice, a key issue is the reliability of the decision, which means that the result should be reproducible upon random repetition of the test with the same patient. This means that multiple CXR images of the same

patient recorded by different X-ray imaging devices under various circumstances should be included in the train dataset, not only images altered via rotation and added noise, which is a common practice in AI algorithm development. Without such training, the AI algorithm can easily be fooled, not only by intentionally produced adversarial examples [43].

To choose the best performing algorithm for a certain medical decision or diagnosis, it is necessary to trial the candidates on the very same, well-constructed dataset that sufficiently covers the whole target population. Just because a certain AI model gives higher accuracy on publicly available datasets does not mean it will perform fine in clinical use. For example, there exists a reported case in automated pneumonia diagnosis, where the AI recognized the fact that an image was recorded by the mobile X-ray equipment in urgent regime and consequently predicted pneumonia with higher probability, causing an increased number of false positives [44].

A special attention should be dedicated to make sure that the AI models learn to make decision based on relevant features of the medical image instead of circumstantial issues. Accordingly, the explainable AI models [45] tend to appear in medical decision making and are likely to attract a quickly increasing attention.

5. Conclusion

In this paper we proposed a CNN model to provide an efficient and accurate solution for the pneumonia detection problem based on X-ray images. The main novelty consisted in the placement of a dropout layer among the convolutional layers of the network. Instead of deploying pretrained networks via transfer learning, we created a CNN model from scratch. Experiments have shown that the proposed model performs better than its counter candidates in terms of accuracy and efficiency, achieving recall and precision well above 97% with predictions produced in only 122 ms.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The work of L. Szilágyi was supported by the Sapientia Institute for Research Programs and the Consolidator Researcher Program of Óbuda University.

REFERENCES

[1] Li YY, Zhang ZY, Dai C, Dong Q, Badrigilan S. Accuracy of deep learning for automated detection of pneumonia using chest X-Ray images: A systematic review and meta-analysis. Comput Biol Med 2020;123 . https://doi.org/10.1016/j.compbiomed.2020.103898 103898.

[2] Jain R, Nagrath P, Kataria G, Kaushik VS, Hemanth DJ. Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning. Measurement 2020;165 . https://doi.org/10.1016/j.measurement.2020.108046 108046.

[3] Dey N, Zhang YD, Rajinikanth V, Pugalenthi R, Sri Madhava Raja N. Customized VGG19 Architecture for Pneumonia Detection in Chest X-Rays. Patt. Recogn Lett 2021;143(67–74). https://doi.org/10.1016/j.patrec.2020.12.010.

[4] Brunese L, Mercaldo F, Reginelli A, Santone A. Explainable Deep Learning for Pulmonary Disease and Coronavirus COVID-19 Detection from X-rays. Comput Meth Prog Biomed 2020;196 . https://doi.org/10.1016/j.cmpb.2020.105608 105608.

[5] Panwar H, Gupta PK, Siddiqui MK, Morales-Menendez R, Bhardwaj P, Singh V. A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. Chaos, Solitons & Fractals 2020;140 . https://doi.org/10.1016/j.chaos.2020.110190 110190.

[6] Ibrahim DM, Elshennawy NM, Sarhan AM. Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases. Comput Biol Med 2021;132 . https://doi.org/10.1016/j.compbiomed.2021.104348 104348.

[7] Jin WQ, Dong SQ, Dong CZ, Ye XD. Hybrid ensemble model for differential diagnosis between COVID-19 and common viral pneumonia by chest X-ray radiograph. Comput Biol Med 2021;131 . https://doi.org/10.1016/j.compbiomed.2021.104252 104252.

[8] Karthik R, Menaka R, Hariharan M. Learning distinctive filters for COVID-19 detection from chest X-ray using shuffled residual CNN. Appl Soft Comput 2021;99 . https://doi.org/10.1016/j.asoc.2020.106744 106744.

[9] Quan H, Xu XS, Zheng TT, Li Z, Zhao MF, Cui XY. DenseCapsNet: Detection of COVID-19 from X-ray images using a capsule neural network. Comput Biol Med 2021;133 . https://doi.org/10.1016/j.compbiomed.2021.104399 104399.

[10] Alhudhaif A, Polat K, Karaman O. Determination of COVID-19 pneumonia based on generalized convolutional neural network model from chest X-ray images. Expert Syst Appl 2021;180 . https://doi.org/10.1016/j.eswa.2021.115141 115141.

[11] Sirazitdinov I, Kholiavchenko M, Mustafaev T, Yixuan Y, Kuleev R, Ibragimov B. Deep neural network ensemble for pneumonia localization from a large-scale chest x-ray database. Comput Electr Eng 2019;78:388–99. https://doi.org/10.1016/j.compeleceng.2019.08.004.

[12] Jaiswal AK, Tiwari P, Kumar S, Gupta D, Khanna A, Rodrigues JJPC. Identifying pneumonia in chest X-rays: A deep learning approach. Measurement 2019;145:511–8. https://doi.org/10.1016/j.measurement.2019.05.076.

[13] Mahmud T, Rahman MA, Fattah SA. CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. Comput Biol Med 2020;122 . https://doi.org/10.1016/j.compbiomed.2020.103869 103869.

[14] Ouchicha C, Ammor O, Meknassi M. CVDNet: A novel deep learning architecture for detection of coronavirus (Covid-19) from chest x-ray images. Chaos, Solitons & Fractals 2020;140 . https://doi.org/10.1016/j.chaos.2020.110245 110245.

[15] Wang J, Bao YM, Wen YF, Lu HB, Luo H, Xiang YF, Li XM, Liu C, Qian DH. Prior-Attention Residual Learning for More Discriminative COVID-19 Screening in CT Images. IEEE Trans Med Imag 2020;39(8):2572–83. https://doi.org/10.1109/TMI.2020.2994908.

[16] Wang Z, Xiao Y, Li Y, Zhang J, Lu FG, Hou MZ, Liu CW. Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays. Patt

Recogn 2021;110 . https://doi.org/10.1016/j.patcog.2020.107613 107613.
[17] Wang SH, Zhang X, Zhang YD. Deep stacked sparse autoencoder analytical model for COVID-19 diagnosis by fractional Fourier entropy. ACM Trans. Manage. Inf Syst 2021;13(1):20. https://doi.org/10.1145/3451357.
[18] Wang SH, Khan MA, Govindaraj V, Fernandes SL, Zhu ZQ, Zhang YD. Deep rank-based average pooling network for Covid-19 recognition. Comput Mater Contin 2022;70 (2):2797–813. https://doi.org/10.32604/cmc.2022.020140.
[19] Horry MJ, Chakraborty S, Paul M, Ulhaq A, Pradhan B, Saha M, Shukla N. COVID-19 Detection Through Transfer Learning Using Multimodal Imaging Data. IEEE Access 2020;8:149808–24. https://doi.org/10.1109/ACCESS.2020.3016780.
[20] Rajaraman S, Siegelman J, Alderson PO, Folio LS, Les R, Antani SK. Iteratively Pruned Deep Learning Ensembles for COVID-19 Detection in Chest X-Rays. IEEE Access 2020;8:115041–50. https://doi.org/10.1109/ACCESS.2020.3003810.
[21] Singh G, Yow KC. An Interpretable Deep Learning Model for Covid-19 Detection With Chest X-Ray Images. IEEE Access 2021;9:85198–208. https://doi.org/10.1109/ACCESS.2021.3087583.
[22] Anis S, Lai KW, Chuah JH, Ali SM, Mohafez H, Hadizadeh M, Yan D, Ong ZC. An Overview of Deep Learning Approaches in Chest Radiograph. IEEE Access 2020;8:182347–54. https://doi.org/10.1109/ACCESS.2020.3028390.
[23] Khan W, Zaki N, Ali L. Intelligent Pneumonia Identification From Chest X-Rays: A Systematic Literature Review. IEEE Access 2021;9:51747–71. https://doi.org/10.1109/ACCESS.2021.3069937.
[24] Dong D, Tang ZC, Wang S, Hui H, Gong LX, Lu Y, Xue Z, Liao HE, Chen F, Yang F, Jin RH, Wang K, Liu ZY, Wei JW, Mu W, Zhang H, Jiang JY, Tian J, Li HJ. The Role of Imaging in the Detection and Management of COVID-19: A Review. IEEE Rev Biomed Eng 2021;14:16–29. https://doi.org/10.1109/RBME.2020.2990959.
[25] Kora P, Ooi CP, Faust O, Raghavendra U, Gudigar A, Chan WY, Meenakshi K, Swaraja K, Plawiak P, Rajendra Acharya U. Transfer learning techniques for medical image analysis: A review. Biocybern Biomed Eng 2022;42:79–107. https://doi.org/10.1016/j.bbe.2021.11.004.
[26] Lefkovits S, Lefkovits L, Szilágyi L. Applications of different CNN architectures for palm vein identification. In: Proc. Modeling Decisions for Artificial Intelligence (MDAI). Lect. Notes Comput. Sci. 2019; 11676:295–306. doi: 10.1007/978-3-030-26773-5_26.
[27] Wager S, Wang S, Liang P. Dropout training as adaptive regularization. In: 27th Annual Conference on Neural Information 2013; pp. 351–359. doi: 10.5555/2999611.2999651.
[28] Kingma DP, Ba J. Adam: a method for stochastic optimization. ArXiv: 1412.6980, 2014. https://arxiv.org/abs/1412.6980
[29] Albelwi S, Mahmood A. A Framework for Designing the Architectures of Deep Convolutional Neural Networks. Entropy 2017;19:242. https://doi.org/10.3390/e19060242.
[30] Thakkar V, Tewary S, Chakraborty C. Batch Normalization in Convolutional Neural Networks – A comparative study with CIFAR-10 data. In: 5th International Conference on Emerging Applications of Information Technology. p. 1–5. https://doi.org/10.1109/EAIT.2018.8470438.
[31] Chowdhury MEH, Rahman T, Khandakar A, Mazhar R, Kadir MA, Mahbub ZB, Islam KR, Khan MS, Iqbal A, Emadi NA, Reaz MBI, Islam MT. Can AI Help in Screening Viral and COVID-19 Pneumonia? IEEE Access 2020;8:132665–76. https://doi.org/10.1109/ACCESS.2020.3010287.
[32] Ren H, Wong AB, Lian WM, Cheng WB, Zhang Y, He JW, Liu QF, Yang JS, Zhang CJ, Wu KS, Zhang HD. Interpretable Pneumonia Detection by Combining Deep Learning and Explainable Models With Multisource Data. IEEE Access 2021;9:95872–83. https://doi.org/10.1109/ACCESS.2021.3090215.
[33] Arias-Londoño JD, Gómez-García JA, Moro-Velázquez L, Godino-Llorente JI. Artificial Intelligence Applied to Chest X-Ray Images for the Automatic Detection of COVID-19. A Thoughtful Evaluation Approach. IEEE Access 2020;8:226811–27. https://doi.org/10.1109/ACCESS.2020.3044858.
[34] Sakib S, Tazrin T, Fouda MM, Fadlullah ZM, Guizani M. DL-CRC: Deep Learning-Based Chest Radiograph Classification for COVID-19 Detection: A Novel Approach. IEEE Access 2020;8:171575–89. https://doi.org/10.1109/ACCESS.2020.3025010.
[35] Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Rajendra Acharya U. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med 2020;121 . https://doi.org/10.1016/j.compbiomed.2020.103792 103792.
[36] Nikolaou V, Massaro S, Fakhimi M, Stergioulas L, Garn W. COVID-19 diagnosis from chest x-rays: developing a simple, fast, and accurate neural network. Health Inf Sci Syst 2021;9:36. https://doi.org/10.1007/s13755-021-00166-4.
[37] Das AK, Ghosh S, Thunder S, Dutta R, Sachin A, Chakrabati A. Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network. Pattern Anal Appl 2021;24:1111–24. https://doi.org/10.1007/s10044-021-00970-4.
[38] Munusamy H, Muthukumar KJ, Gnanaprakasam S, Shanmugakani TR, Sekar A. FractalCovNet architecture for COVID-19 chest X-ray image classification and CT-scan image segmentation. Biocybern Biomed Eng 2021;41:1025–38. https://doi.org/10.1016/j.bbe.2021.06.011.
[39] Joshi RC, Yadav S, Pathak VK, Malhotra HS, Khokar HVS, Parihar A, Kohli N, Himanshu D, Garg RK, Bhatt MLB, Kumar R, Singh NP, Sardana V, Burget R, Alippi C, Traviso-Gonzalez CM, Dutta MK. A deep learning-based COVID-19 automatic diagnostic framework using chest X-ray images. Biocybern Biomed Eng 2021;41:239–54. https://doi.org/10.1016/j.bbe.2021.01.002.
[40] Singh S, Tripathi BK. Pneumonia classification using quaternion deep learning. Multimed Tools Appl 2022;81 (2):1743–64. https://doi.org/10.1007/s11042-021-11409-7.
[41] Dash AK, Mohapatra P. A fine-tuned deep convolutional neural network for chest radiography image classification on COVID-19 cases. Multimed. Tools Appl., available online 21 Sep 2021. doi: 10.1007/s11042-021-11388-9.
[42] Gour M, Jain S. Automated COVID-19 detection from X-ray and CT images with stacked ensemble convolutional neural network. Biocybern Biomed Eng 2022;42:27–41. https://doi.org/10.1016/j.bbe.2021.12.001.
[43] Kelly CJ, Karthikesalingam A, Suleyman M, Corrado G, King D. Key challenges for delivering clinical impact with artificial intelligence. BMC Med 2019;17:195. https://doi.org/10.1186/s12916-019-1426-2.
[44] Zech JR, Badgeley MA, Liu M, Costa AB, Titano JJ, Oermann EK. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. PLoS Med 2018;15 . https://doi.org/10.1371/journal.pmed.1002683 e1002683.
[45] Csiszár O, Csiszár G, Dombi J. How to implement MCDM tools and continuous logic into neural computation?: Towards better interpretability of neural networks. Knowl Based Syst 2020;210 . https://doi.org/10.1016/j.knosys.2020.106530 106530.