

A Deep Learning Approach for Diagnosis Chest Diseases

Tuba Torabipour*

Department of Computer Engineering and Information Technology
Payame Noor University (PNU)
Tehran-Iran
s.torabi25@gmail.com

Yousef Jahangiri Golshavari

Electronic Engineering- Bushehr Islamic Azad University
Bushehr- Iran
Yj71365@gmail.com

Safieh Siadat

Department of Computer Engineering and Information Technology Payame Noor University (PNU)
Tehran-Iran
safieh.siadat@gmail.com

Received: 2021/05/13

Revised: 2021/09/27

Accepted: 2021/10/09

Abstract—The human chest contains vital organs such as the heart, lungs, and other organs. Chest radiology is one of the best and least costly methods to diagnose chest diseases. In this study, proposed a new method to diagnose 14 main diseases of the chest such as (cardiomegaly, emphysema, effusion, hernia, nodule, pneumothorax, atelectasis, pleural - thickening, mass, edema, integration, penetration, fibrosis, pneumonia) using the neural network and deep learning to increase accuracy, sensitivity, and specificity. The proposed method is implemented in the form of a web application and is available as a decision-making system for physicians to diagnose chest diseases. The results of the simulation on the sample dataset showed that the diagnosis of chest diseases was 98.93%, indicating the high efficiency of the new method. Finally, the proposed method was compared with other deep learning architectures such as densenet121, vgg16, exception architecture on the same dataset, which showed a 5% higher accuracy than them.

Keywords— *Deep learning; Convolutional Neural Network; Chest X-Ray* .

1. INTRODUCTION

The neurological and nutritional problems in recent years, there has been indiscriminate growth in the emergence of various diseases in vital chest organs such as lung, heart, esophagus, and tray. Regarding the diagnosis of chest pain, the first diagnostic techniques were X-ray and radiology imaging. However, these images are not always available for diagnosing the diseases in all radiology centers, and if there is a specialist physician, there is still the likelihood of misdiagnosis due to human error. Unfortunately, due to frequent misdiagnosis cases, we are witnessing the progression of chest diseases such as edema and emphysema and their conversion to cancer. The most problem to diagnose chest diseases not exist a diagnosis method with high accuracy and high sensitivity. In this paper, to Problem Solving, we present a web application method to diagnose chest diseases with radiology images. Our web application could improve accuracy and sensitivity with improved convolutional neural networks based on medical images.

Convolutional neural networks are a particular type of multilayer neural network. Moreover, like any other neural

network, they're trained with a back-to-back algorithm version for different architecture areas. In this network, pixel images with minimal preprocessing are used to design visual patterns directly. They can detect patterns with drastic changes (e.g., handwriting identification patterns) by considering strengths over distortion and simple geometric transformations [1].

The proposed network in the chest area is based on space and other medical symptoms. The present study is outlined as follows: we have related works in the second section, and the proposed model in Section three. In Section four, we have the simulation, and Section five contains the research results.

2. RELATED WORKS

Several research studies have addressed the diagnosis of chest diseases using artificial intelligence methods. Err et al. [2] used multimedia, probation, quantum variance learning, and generalized regression neural networks to diagnose chest diseases. Moreover, pulmonary diseases and persistent hypertension are also diagnosed using neural networks and an artificial immune system.

Khobragade et al. [3] conducted a valuable study on diagnosing lung diseases such as pneumonia and lung cancer using chest X-rays, including the histogram scale in image segmentation for image preprocessing and feedback neural network for classification. Furthermore, browsers were also used to classify medical diseases. However, the performance of these techniques as deep networks has not been acceptable regarding the accuracy, calculation time, and minimum square error of advanced learning-based systems to promote image classification accuracy.

Avendi et al. [4] managed to use these networks on medical images to perform disease classification tasks, and their findings revealed that deep networks could easily detect useful features in different categories of images.

To diagnose several diseases such as Mers, COVID-19, Asnaoui et al. [5] compared the deep convolutional neural network (dcnn) architectures to provide the automatic binary classification for the adjusted versions of pneumonia images (namely vgg16, vgg19, densenet201, inception_resnet_v2, inception_v3, resnet50, and mobilenet_v2). Their proposed

technique was performed using the chest X-Ray; Chest CT dataset, including 5,856 images (4273 pneumonia cases and 1583 normal cases). The researchers concluded that the regulated version of Resnet50, MobileNet_V2, and Inception_Resnet_V2 exhibited higher performance (>96% accuracy). Unlike CNN, Xception, VGG16, VGG19, Inception_V3, and DenseNet201 exhibited lower performance (>84% accuracy). demonstrated highly satisfactory performance with increased education speed and the authenticity of credibility.

Prayogo et al. [6] used deep learning methods to further classify the causes of pneumonia. They used the Siamese Convolution Network (SCN) to classify chest lung pneumonia images on optical datasets of Coherence Tomography (OCT) and Chest X-Ray images. To this end, they included 5863 chest X-ray images (CXr) of patients aged one to five years and classified them into three categories of natural conditions, bacterial pneumonia, and viral pneumonia. They achieved the accuracy of 80.03%, and F1 score of 79.59 by using SCN; however, the drawback was the low accuracy of this technique.

Luján-García et al. [7] used a transmission learning method to classify pneumonia automatically. These scientists used chest X-ray images datasets containing 3883 chest X-ray images and classified them into two healthy and pneumonia categories. Their proposed pre-training method used Xception network with ImageNet weights. They achieved 0.84% accuracy, 0.99% recall, and an F1-score of 0.91%; however, accuracy <90% is not widely used in this type of disease classification.

Al-Mamlook et al. [8] used the CNN-based deep learning architecture for classification. They trained modified chest x-ray images (pneumonia images) at several stages and then compared their proposed method with seven other machine learning methods. In this study, the researchers achieved an accuracy of 98.46 in the diagnosis of chest infection. Despite its high accuracy, this method has a long run time.

Sekuboyina et al. [9] published a multi-labeled knowledge diagram of chest X-ray images, which extracted labels from the dataset and, in addition to classifying them, used them as a prediction problem. Merging the information helped them in adding additional nodes and relationships. CheXpert dataset was used in this study. The accuracy under the curve was 83.5% and was improved by 1% compared to other methods; however, the achieved accuracy was still <90%, as such, it's not very practical for the classification issues.

Rasheed et al. [10] investigated the potential of machine learning methods for automatic diagnosis of corona virus with high accuracy from X-ray images. The two most commonly used classifiers were selected: logistic regression (LR) and convolutional neural networks (CNN). They used principal component analysis (PCA) to further speed up the learning process and improve the classification accuracy by selecting the highly discriminate features. The available online dataset and incorporated GAN was used to have 500 X-ray images, and they achieved an accuracy of 95.2–97.6% without PCA and 97.6–100% with PCA.

Ibrahim et al. [11] used a deep learning approach based on the pretrained AlexNet model to classify COVID-19, non-COVID-19 viral pneumonia, bacterial pneumonia, and normal CXr scans obtained from different public databases. The

model was trained to perform two-way classification, three-way classification, and four-way classification. This study achieved 94.43% accuracy, 98.19% sensitivity, and 95.78% specificity.

However, the main research gap regarding the diagnosis of chest diseases wasn't a method to diagnose all the most important diseases in the chest areas with high accuracy. Berranib et al. [12] used CT scan images to diagnose COVID-19. They adopted deep learning architectures to classify individuals as either patients or healthy ones. They used four sets of datasets and convolutional neural network (CNN) and Grad-CAM and Fast-CAM models. Finally, their method achieved 92% accuracy. That is why they developed a web application to provide the grounds for diagnosing the diseases by the CT scan images. The only disadvantage of their study may be is <95% accuracy.

Dey et al. [13] used artificial intelligence techniques to diagnose COVID-19 based on radiology images. To develop a web application, they adopted the CNN architecture and pre-training of ResNet, VGG, Inception, and Efficient Net [14]. In this study, their proposed model, DLH-COVID, diagnosed COVID-19 with 96% accuracy. They used a chest X-ray dataset (COVID-19 pneumonia) containing 6432 chest radiology images, and their major task was to classify images into pneumonia and healthy cases. One weakness of this study was the diagnosis of COVID-19 based on pneumonia since some patients with pneumonia, according to physicians, are not necessarily COVID-19 positive. Furthermore, we decided to present a method with a web application capable of detecting major chest diseases so that, in addition to the diagnosis of COVID-19, the other diseases can be diagnosed in this area.

3. PROPOSED MODEL

Our dataset, i.e., sample, was first uploaded. In the concerned dataset, there were two types of data, namely Test and Train. In the first step, the data was read and normalized. Then, the CNN model was created, and the frequency of weights and errors were calculated. In the following, we tested the network on Train data (datasets for dataset Training). This cycle is recursive, and as shown in Fig.1, the number of rounds required for Training data is repeated. After each cycle of repetition on Train data, the error rate decreases. When the Training network is done with Train data, finally test the network on the Test data, and the result is categorizing a variety of diseases.

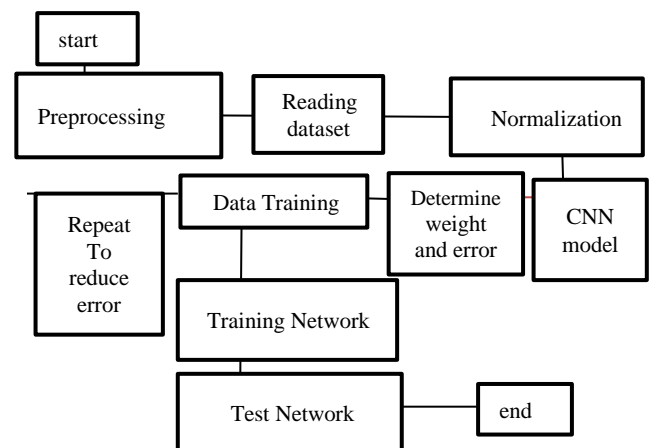


Fig. 1. proposed model.

3-1. Structure of Convolutional Neural Network

As shown in Fig.2, the chest images inputs size is 128*128 * 3 (mean of 3 is tree channel of picture R*G*B) were used in this design. In this regard, the first convolution layer multiplies 32 filters sized 3*3 in the input, and the result is obtained after applying the activation function as a 2D feature map i^* ; however, the third dimension or depth ($*n$) is determined by the number of filters. Then it's reduced to its minimum by the Sub-Sampling layer with Max Pooling 2*2 window. The maxpooling layers and convolution layers are collectively called hidden layers. In this regard, a layer is required for normalization. In total, three layers were used in this design, and we calculated the filter in each convolution layer 3*3. Finally, the fully connected layer using the weights in the softmax function and creates the classification [15].

3-2. Features of CNN MODEL

In this section, we present the features of different layers of CNNs in the design. The concerned CNN comprised four layers, namely convolutional layer, max pooling layer, fully-connected layer, and subsampling layer with normalization and flattening as sub-layers [15].

1) Convolution Layer

In this layer, an input image $R*C$ is matrixed with a core (filter) $a*a$, and the multiplication result is inserted into a feature map. Hence, any $R*C$ block existing in the input creates an output pixel in the feature map following the convolution operations. Moreover, at the following interlocking stages, a block of the created image in the feature map is multiplied by a matrix filter, and the result in the next layer feature map is created as a 2D image $j*j$ [15].

2) Sub Sampling Layer

This layer mainly aims to reduce the spatial dimensions of feature maps obtained from the previous convolution layer. To this end, a mask $b*b$ is selected, and the pooling operation

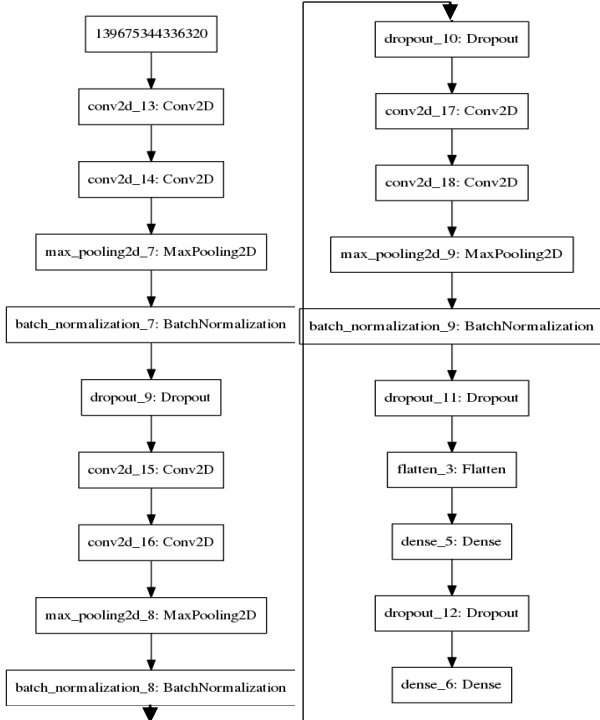


Fig. 2. The structure of the proposed convolution neural network.

is performed between the mask and the feature maps. Different pooling methods such as average, summation, and maximum are suggested. However, our optimal pooling method was to select the full merge, where the pixel value of the output image was derived from the maximum value of each block. A merge pooling layer helps the convolution layer interpret rotation and translation among the input images [15].

3) Batch Normalization Layer

In Python and cross-library, there is a normalization layer called Batch normalization. Batch normalization is a method used to make artificial neural networks faster and more stable via the normalization of layers' inputs by re-centering and re-scaling. In a neural network, batch normalization is achieved through a normalization step that fixes the means and variances of each layer's inputs. Ideally, the normalization would be conducted over the entire Training set, but to use this step jointly with stochastic optimization methods, it's impractical to use the global information. Thus, the normalization is restrained to each mini-batch in the Training process [16]. The primary task of this layer is to normalize the activities of the previous layer in each category. It applies a transformation maintaining the average activation close to 0 and the standard deviation of activation close to 1. Using this layer, normalization is performed optimally [15].

4) Flatten Layer

The flattening layer turns the data into a one-dimensional array to enter them into the next layer. We smooth the output of convolution layers to create a long vector of features and attach it to the final classification model, which is called a complete connector layer. In other words, we put all pixel data in one line and interact with the final layer [17].

5) Fully Connected Layer

This is the final layer of CNN and uses the Softmax normalization function, ((Equ.(1) and(2)).

$$y_i^{(l)} = f(z_i^{(l)}) \quad (1)$$

$$\text{where } z_i^{(l)} = \sum_{j=1}^{m_i^{(l-1)}} w_{i,j}^{(l)} y_j^{(l-1)} \quad (2)$$

$w_{i,j}^{(l)}$: Weights to be connected by the full connector layer to create each class.

6) Nonlinear Transfer Function

We need to if the nonlinearity occurs in the full connector layer within the neurons, not in separate layers such as convolution and integration. Nevertheless, when we ultimately reach the output signals, CNN training begins. The tutorial is performed using a random simulation algorithm [15,16]. The algorithm estimates gradients using a random sample of the Train set. The CNN parameters are performed following the Training phase [17].

4. SIMULATION

We used chest datasets in [18] and sample datasets including 1582 chest radiology images from the popular NIH dataset, which provided us with the best set of chest radiology images in two Test and Train datasets. This dataset contains 1186 and 396 radiology images for Training and Testing, respectively.

We evaluate the proposed network analysis of radiology images in the convolutional neural network. To this end, we implement the proposed convolutional neural network in Python. Multiple experiments determine the appropriate values of the network learning parameters. A collection of images was used (75% for Training and 25% for network validation). In the following, we describe the detail of the proposed web application.

4-1. Structure of the Proposed Cnn Model in Python

To implement the proposed design, we first changed the image dimensions to $3*128*128$ in the main network structure containing three hidden layers, one Flatten layer, and fully-connected layers.

The first hidden layer combines two convolution layers with ReLu activation function, one merge layer, normalization and a dropout function to regularize. The first convolution layer with 32 filters is the creation of 32 feature maps. The second convolution layer, which received 32 previous feature maps, has the same kernel size of $3*3$ in each convolution layer, and finally, the convolution layer increased the number of parameters. Connections from 896 to 9248 are compared to the previous layer, and the continued merged maxpooling layer with the central kernel is $2*2$ with. Step 22 is considered in normalizing data, as presented in the first row of Table I. Finally, the normalization layer and the dropout function regularize and normalize the data, as shown in the first row of Table I.

The second hidden layer combines two convolution layers with a ReLu normalization function, a pooling layer, and normalization and a dropout function to regularize. The result of the first convolution layer of this layer with 128 filters are creating 128 feature maps. The second layer of convolution, which received 128 previous feature maps and finally with a kernel size of $3*3$ and 3 128 filters, increased the number of parameters and connections from 36992 to 147584 compared to the previous layer and in the continuing layer of maxpooling with the central kernel is size $2*2$ and step 22 is considered. Finally, the normalization layer and the dropout function regularize and normalize the data, as shown in the second row of Table I.

The third hidden layer combines two convolution layers with ReLu normalization function, one pooling layer, normalization, and a dropout function to regularize. The result of the first convolution layer of this layer with 86 filters is creating 86 feature maps. The second layer of convolution, which received 86 previous feature maps and finally with a kernel size of $3*3$ and 86 Filter reduces the number of parameters and connections from 99158 to 66650 compared to the previous layer and is considered as a continuing layer of integration with central kernel $2*2$ and step 2. Finally, the normalization layer and the dropout function to regularize and normalize the data, as presented in the third row of Table I.

TABLE I. IMPLEMENTATION OF THE PROPOSED MODEL

Layer Type	Output Form	Parameters	Layer Name
conv2d_1(Conv2D)	(None,128,128, 32)	896	The first hidden layer
conv2d_2(Conv2D)	(None,128,128, 32)	9248	
max_pooling2d (MaxPooling)	(None,64,64, 3)	0	
batch_normalization _1 (Batch)	(None,64,64,32)	128	
dropout_1(Dropout)	(None,64,64, 32)	0	
conv2d_3(Conv2D)	(None,64,64,128)	36992	The second hidden layer
conv2d_4(Conv2D)	(None,64,64, 128)	147584	
max_pooling2d (MaxPooling)	(None,32,32,128)	0	
batch_normalization _2 (Batch)	(None,32,32,128)	512	
dropout_2 (Dropout)	(None,32,32,128)	0	
conv2d_5 (Conv2D)	(None,32,32,86)	99158	The third hidden layer
conv2d_6 (Conv2D)	(None,32,32,86)	66650	
max_pooling2d (MaxPooling)	(None,16,16,86)	0	
batch_normalization _3 (Batch)	(None,16,16,86)	344	
dropout_3 (Dropout)	(None,16,16,86)	0	
flatten_1 (Flatten)	(None,22016)	0	flattenlayer
dense_1 (Dense)	(None,512)	11.272.704	fully connected layer
dropout_4 (Dropout)	(None,512)	0	
dense_2 (Dense)	(None,14)	7182	

Flatten layer: The output of third hidden layer is inserted into the Flatten layer to prepare it to enter the classification layers. The Flatten layer in the Python keras library Flatten, performs best and reduces the number of parameters to zero, as presented in the fourth row of Table I.

Fully-connected layers: At this stage, the first complete connector layer is classified with 512 neurons. After running on the Flatten layer, it increases the number of parameters by 11272704, uses the dropout function to regularize, and reduces the number of parameters to zero. Finally, a fully-connected layer is created using 14 neurons and the Softmax normalization function. Furthermore, it increases the number of parameters to 7182 after running on the output of the second normalization layer, resulting in the classification of the chest radiology images into 14 classes: Cardiomegaly, Emphysema, Effusion, Hernia, Nodule, Pneumothorax, Atelectasis,

Pleural_Thickening, Mass, Edema, Consolidation, Infiltration, Fibrosis, and pneumonia.

Table I presents the structure of all layers of the proposed model. This table has four columns showing the characteristics of the layers, including layer name, parameters, output shape, and layer type in Python and rows representing a layer, the name of the layer, and the number of output parameters. In the proposed model, there are 11641398 parameters, 11640906 trained parameters, and 492 non- trained parameters.

4-2. Evaluation of the Results of the Proposed Model and the 3 other Model

This study aimed to classify CXRs disease in14 classes. F1-score, recall, precision, and accuracy are used to evaluate the classification. Accuracy is a measure which results from dividing the number of correctly identified cases by the total number of cases. Accuracy is the proximity of a calculated value to a natural or real value. In other words, this evaluation criterion can measure a quantity whose accuracy can be measured, (Equ.(3)).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

In deep learning architectures, the precision for a class is equal to the number of correctly classified (labeled) items divided by the sum of true or false classified items.(Equ.(4)).

$$precision = \frac{TP}{TP + FP} \quad (4)$$

Recall refers to the fraction of correctly classified items to the total number of items classified in a class. (Equ.(5)).

$$recall = \frac{TP}{TP + FN} \quad (5)$$

Based on recall and precision calculations, the F1-score value is calculated. F1-score is a useful criterion to evaluate the classification performance and defining the weighted average of recall and precision values [19].

The ideal and worst F1-score values for the classification algorithm are equal to 1 and 0, respectively.

This section simulates the proposed model and gets the 4 major plots, called accuracy, loss, confusion matrix, and roc curves. Fig.3 shows the accuracy of the proposed model with the horizontal axis indicating the number of Test rounds and the vertical axis representing the model accuracy. Fig.4 indicates the loss of the proposed model, with its horizontal axis displaying the number of Test rounds and the vertical axis showing the loss model. The blue line displays the Training data in these figures, and the red line shows the Test data. Fig.5 shows the confusion matrix, the horizontal and vertical axis shows classes of diseases., and Fig.6 presents the roc curve. The horizontal axis shows the false positive rate in these shapes, and the vertical axis shows the true positive rate.

4-3. Comparison the Accuracy of the Proposed Model and 3 other Model

In this section, First, we have a brief explanation about 3Famous models in CNN, including VGG16, xception, and

Densenet121 then we compared the proposed model with these models with our dataset and 50 rounds of Training, and for each model, we obtained the accuracy, precision, recall, and F1-score assessment parameters separately.

VGG16 is a convolution neural net (CNN) architecture that was used to win ILSVR(Imagenet) competition in 2014. It's considered to be one of the excellent vision model architecture till date. The most unique thing about VGG16 is that instead of

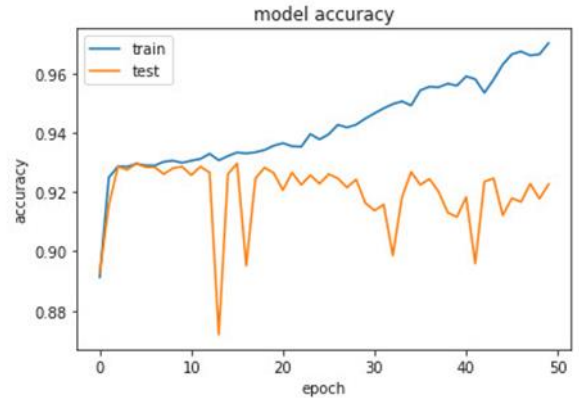


Fig. 3. The model accuracy plot.

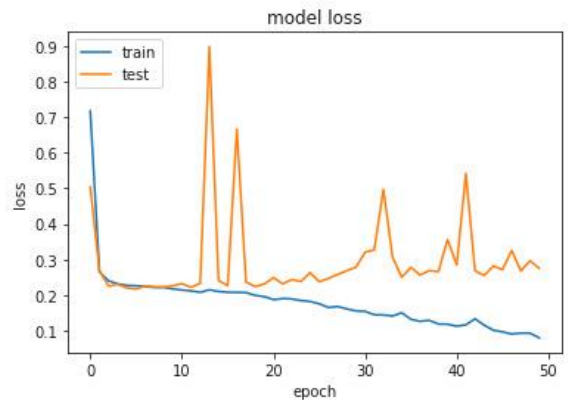


Fig. 4. The model loss plot.

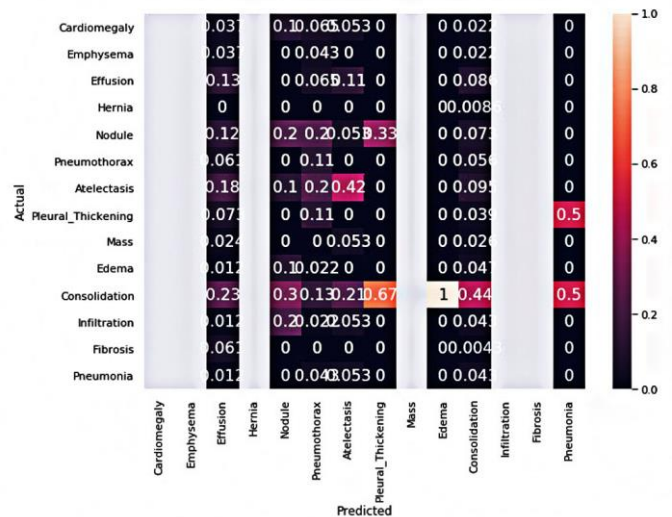


Fig. 5. The confusion matrix plot.

having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used the same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end, it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights [20].

DenseNet 121is a convolutional neural network where each layer is connected to all other layers that are deeper in the network, that is, the first layer is connected to the 2nd, 3rd, 4th, and so on, the second layer is connected to the 3rd, 4th, 5th and so on. This is done to enable maximum information flow among the layers of the network. To preserve the feed-forward nature, each layer obtains inputs from all the previous layers and passes on its own feature maps to all the layers which will come after it. The 121 in DenseNet 121 refers to it has 121 convolutional layers [21].

Xception is an adaptation from Inception, where the Inception modules have been replaced with depthwise separable convolutions. It also has roughly the same number of parameters as Inception-v1 (23M).Xception takes the Inception hypothesis to an extreme.

Firstly, cross-channel (or cross-feature map) correlations are captured by 1x1 convolutions.

Consequently, spatial correlations within each channel are captured via the regular 3x3 or 5x5 convolutions.

Taking this idea to an extreme means performing 1x1 to every channel, then performing a 3x3 to each output. This is identical to replacing the Inception module with depthwise separable convolutions [22].

The following results were corrected as you can see in Table II.

In the following, we will compare the accuracy and roc curve plots of the above 4 models.

Fig 7-A shows the accuracy of DenseNet121 model, Fig 7-B shows the accuracy of the VGG16 model, Fig 7-C shows the accuracy of the Exception model, and Fig 7-D shows the proposed accuracy. In these shapes, the horizontal axis shows the number of Training rounds and the vertical axis the model accuracy, the Training blue line, and the Test red line:

Fig 8-A shows the roc curve of the DenseNet121 model, Fig 8-B shows the roc curve of VGG16 model, Fig 8-C shows the roc curve of the Exception model, and Fig 8-D shows the proposed roc curve. In these shapes, the horizontal axis shows the false positive rate, and the vertical axis shows the true positive rate:

At last, we compare accuracy, f1 score, precision and recall of the 4 models.

Fig.9 shows the accuracy of the 4 models, Fig.10 shows the f1 score of the 4 models, Fig.11 shows the precision of the 4 models, and Fig.12 shows the recall of 4 modes.

4-4. Build a Web Application with the Proposed Model

In this section, we will first add the Streamlit library to our Python code and create a separate file in Python, first file one called tapp.py containing the application code and second file

TABLE II. COMPARISON THE ACCURACY OF THE PROPOSED MODEL AND 3 OTHER MODEL

F1-Score	Recall	Precision	Accuracy	Architecture
92.26%	89.17%	94.77%	98.93%	Our model
51.82%	39.80%	75.27%	94.77%	DENSENET121
37.2%	37.2%	37.2%	92.86%	VGG16
94.37%	93.09%	94.37%	87%	XPTION

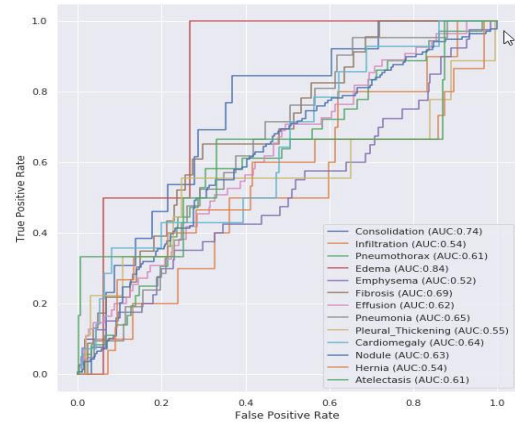


Fig. 6. The roc curve plot.

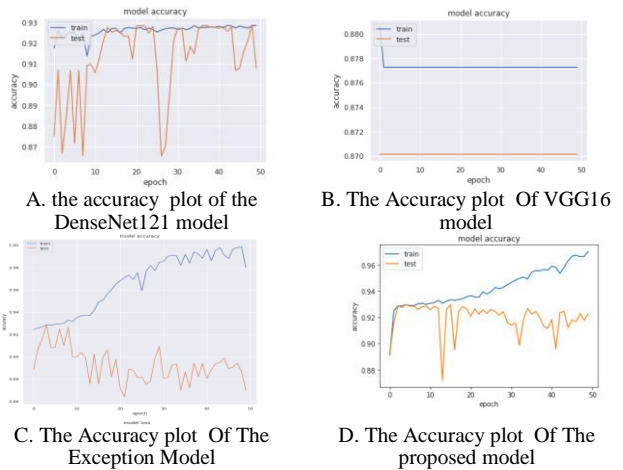


Fig. 7. Comparison The Accuracy Of 4 models

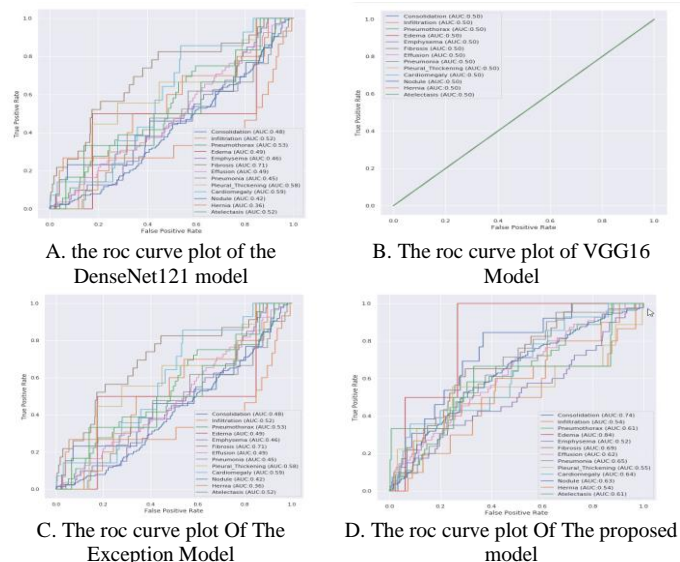


Fig. 8. Comparison The roc curve Of 4 models

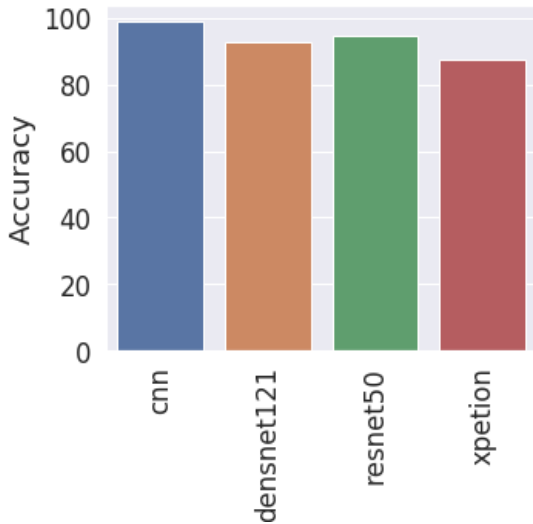


Fig. 9. shows the accuracy of the 4 models

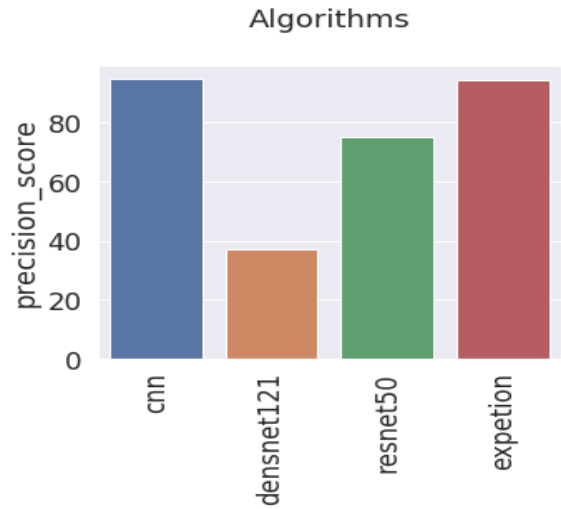


Fig. 11. shows the precision of the 4 models.

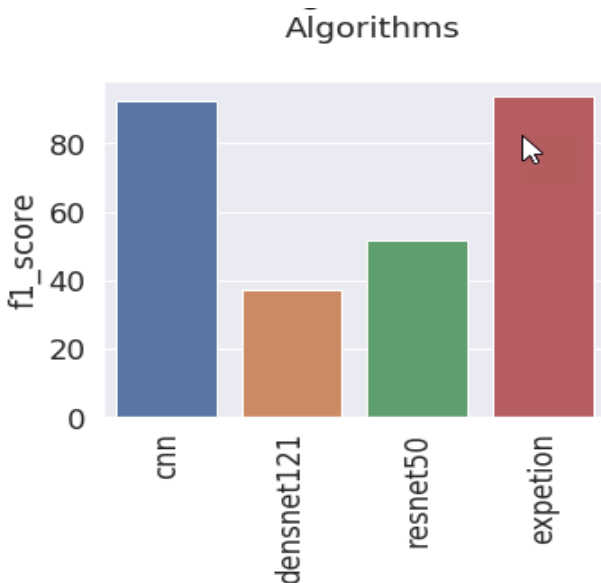


Fig. 10. shows the f1 score of 4 models.

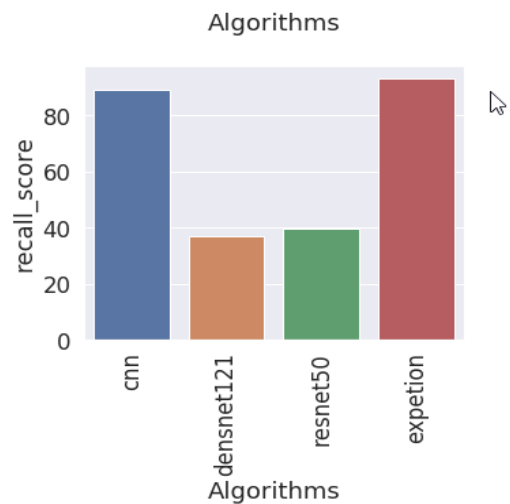


Fig. 12. shows the recall of the 4 models.

another one named tmodel.py containing the proposed model codes.

Then, in the CMD platform of the Anaconda platform with streamlit run command run taap.py run and the application in the internal host addresses and with the default browser is raised. We can develop the application by the heroku website or other web servers such as AWS and use it to diagnose chest diseases by uploading radiology images .but now we don't publish this web app until publishing in Iranian host for using Iranian radiologists and physicians. in Fig.13 we show our web application in the internal host.

5. CONCLUSION

In this study, to improve the diagnosis of chest disease, we improve a Cnn model and build a web application with the proposed model, and regarding its power in the classification and diagnosis of thoracic diseases, the accuracy rates of 91.51% and 98.93% were obtained for the Test and Train data, respectively. In the primary model of the proposed network, 1483 chest images in the sample dataset were used for Training



Fig. 13. shows our web application in the internal host.

(75%) and Testing (25%). Finally, we compared the proposed model and three other deep learning models (namely VGG16, Exception, and Densenet121) with the accuracy rates <90% and revealed the higher accuracy of the proposed model to diagnose chest diseases. It's hoped that in the future and with further efforts, we can improve the CNN models with errors rates <1%.

REFERENCES

[1] Udeshani, K. A. G., Meegama, R. G. N., & Fernando, T. G. I. (2011) "Statistical Feature-based Neural Network Approach for the Detection

of Lung Cancer in Chest X-Ray Images”, International Journal of Image Processing (IJIP), Vol. 5, pp. 425-434, October 2011.

[2] Er, O., Yumusak, N., & Temurtas, F. (2010). “Chest diseases diagnosis using artificial neural networks,” Expert Systems with Applications, Vol. 37, pp. 7648-7655, December 2010.

[3] Khobragade, S., Tiwari, A., Patil, C. Y., & Narke, V. (2016). “Automatic detection of major lung diseases using chest radiographs and classification by feed-forward artificial neural network”, Proceedings of 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), pp. 1-5, July. 4-6 (2016), Delhi, India.

[4] Aveni, M. R., Kheradvar, A., & Jafarkhani, H. (2016). “A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac MRI” ,Medical Image Analysis, Vol. 30, pp. 108-119, May 2016.

[5] Asnaoui, K. E., Chawki, Y., & Idri, A. (2020). “Automated methods for detection and classification pneumonia based on x-ray images using deep learning”, arXiv preprint arXiv:2003.14363, Vol. 1, pp. 15-25, March 2020.

[6] Prayogo, K. A., Suryadibrata, A., & Young, J. C. (2020). “Classification of pneumonia from X-ray images using siamese convolutional network”, Telkomnika, Vol. 18, pp. 1302-1309, February 2020.

[7] Luján-García, J. E., Yáñez-Márquez, C., Villuendas-Rey, Y., AND Camacho-Nieto, O. (2020). “A transfer learning method for pneumonia classification and visualization” Applied Sciences, Vol. 10, pp. 2908, April 2020.

[8] Al Mamlook, R. E., Chen, S., & Bzizi, H. F. (2020). “Investigation of the performance of Machine Learning Classifiers for Pneumonia Detection in Chest X-ray Images”, In 2020 IEEE International Conference on Electro Information Technology (EIT), pp. 098-104, July. 31-1 (2020), Chicago, IL, USA.

[9] Sekuboyina, A., Oñoro-Rubio, D., Kleesiek, J., & Malone, B. (2021). “A Relational-learning Perspective to Multi-label Chest X-ray Classification”. arXiv preprint arXiv:2103.06220, Vol. 1, pp. 30-42, March 2020.

[10] Rasheed, J., Hameed, A. A., Djeddi, C., Jamil, A., & Al-Turjman, F. (2021). “A machine learning-based framework for diagnosis of COVID-19 from chest X-ray images. Interdisciplinary Sciences: Computational Life Sciences”, Vol. 13, pp. 103-117, January 2021.

[11] Ibrahim, A. U., Ozsoz, M., Serte, S., Al-Turjman, F., & Yakoi, P. S. (2021). “Pneumonia classification using deep learning from chest X-ray images during COVID-19”. Cognitive Computation, Vol. 1, pp. 1-13, January 2021.

[12] BERRANIB, H. B., NAÏLI, Q., ABDI, M. E. H., BENSALAH, K., & BELAZZOUGUI, D. (2021). “DIAG a Diagnostic Web Application Based on Lung CT Scan Images and Deep Learning”. Stud Health Technol Inform, Vol. 1, pp. 332-336, May 2021.

[13] Dey, S., Bacellar, G. C., Chandrappa, M. B., & Kulkarni, R. (2021). “COVID-19 Chest X-Ray Image Classification Using Deep Learning”. medRxiv 2021.07.15.21260605, Vol. 1, pp. 12-20, July 2021.

[14] <https://toad.li/xray>

[15] Bouvrie J.(2006).” Notes on Convolutional Networks” Computer Science, Vol. 1, pp. 1-10, March 2006.

[16] Bou J.(2018).”Glossary of Deep Learning: Batch Normalisation”. Computer Science, Vol. 1, pp. 10-17, March 2018.

[17] Valueva, M.V., Nagornov, N.N., Lyakhov, P.A., Valuev, G.V., & Chervyakov, N.I. (2020). “Application of the residue number system to reduce hardware costs of the convolutional neural network implementation”. Mathematics and Computers in Simulation, Vol. 177, pp. 232-243, April 2020.

[18] <https://www.kaggle.com/nih-chest-xrays/sample>

[19] Fouladi, S., Ebadi, M. J., Safaei, A. A., Bajuri, M. Y., & Ahmadian, A. (2021). “Efficient deep neural networks for classification of COVID-19 based on CT images: Virtualization via software defined radio”, Computer Communications, Vol. 176, pp. 234-248, August 2021.

[20] Saxena, A., & Chandra, S. (2021). “Transfer Learning in Biological and Health Care”. In Artificial Intelligence and Machine Learning in Healthcare ", Springer, Vol. 1, pp. 89-98, May 2021.

[21] Singh, D., Kumar, V., & Kaur, M. (2021). "Densely connected convolutional networks-based COVID-19 screening model", Applied Intelligence, Vol. 51, pp. 3044-3051, February 2021.

[22] Chollet, F. (2017). "Xception: Deep Learning with Depthwise Separable Convolutions", In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251-1258, July. 4-5 (2017), San Juan, PR, USA.



Touba Torabipour has a bachelor's degree, master's degree in computer engineering, majoring in software engineering. Her areas of interest include Deep learning, reinforcement learning, machine vision, artificial intelligence, and the Internet of Things. She has published more than 4 articles in prestigious Iranian and international journals and conferences.



Yousef Jahangiri Golshvari has a bachelor's degree in electrical engineering with a focus on electronic engineering from Bushehr Azad University. He has been active in the field of industry-university communication since 1386 in Wireless, design and implementation of urban fire alarm network, fire forecasting systems, process automation of oil, steel and petrochemical industries, the use of artificial intelligence in the prediction of plant diseases and smart plant.



Seyedeh Safieh Siadat, Assistant Professor, Payame Noor University, has a bachelor's, master's and doctoral degree in computer engineering majoring in software engineering. She has been a full-time faculty member at Payame Noor University since 1996. Her areas of interest include the Internet of Things, game theory, machine learning, and cloud computing. She has published more than 40 articles in prestigious international journals and conferences.