Proceedings of the 7th International Conference on Intelligent Computing and Control Systems (ICICCS-2023)
IEEE Xplore Part Number: CFP23K74-ART; ISBN: 979-8-3503-9725-3

2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS) | 979-8-3503-9725-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICICCS56967.2023.10142861

Pneumonia Detection using Chest X-ray Images using CNN Algorithm

Surya Deepta Mazumdar[1]
Student
Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, India.
suryadeeptamazumdar01@gmail.com

Ashish Kumar[2]
Student
Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, India.
ashu07feb@gmail.com

R. Sethuraman[3]
Associate Professor
Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, India.
sethuraman.cse@sathyabama.ac.in

***ABSTRACT*--- Pneumonia is a serious health condition where the patient's lungs get infected by bacteria, virus or sometimes fungi. This causes burning sensation in the alveoli present in the lungs. In response to this, body forms fluids in the lungs that can cause trouble in breathing, coughing and even chest pain. The increasing demand of medical care for elderly care and children below five years of age requires advanced healthcare technologies. Premature diagnosis is vital for fast recovery. This research study intends to propose a novel technique for detecting pneumonia in chest x-ray images by training a model to classify the patient's condition as normal or pneumonic. The proposed model applies CNN, a deep learning algorithm to detect the health condition of the patient by analyzing their chest X-ray images. The objective of this research study is to attain the model accuracy as high as possible to provide accurate results to the user.**

Keywords: Deep Learning, chest X-ray images, ANN, CNN, Pneumococcal, Pneumonia

I. INTRODUCTION

Deep learning has proven to be a big asset in the field of medical science. Its ability to analyze the image and extract useful features can be used for detecting diseases especially pneumonia. It takes the help of the essential features extracted to classify the patient's chest x-ray image as pneumonia or normal. Pneumonia is a disease that targets the lungs. Since lungs are the key organs in the human respiratory system for absorbing the oxygen from the inhaled air, its improper function is enough to lead a patient into the critical health state. As per the reports of WHO, pneumonia is the cause for one in three deaths in India. So, detecting pneumonia at early stage can prevent such cases. Professional radiologists are required for analyzing chest x-rays for confirming the condition of the patient. An incorrect confirmation can lead to wrong treatment which can worsen the situation. This manual work can take time and can slow down the initiation of the treatment and medication.

Deep learning algorithms can accelerate this process of pneumonia detection by automating the whole operation with accurate results. This can help in identifying and curing the disease without any delay especially in the rural areas where the availability of such professionals is low. Hence, it is a real-time project having real-world application.

The model used here for detection is built by employing CNN algorithm which is a deep learning algorithm because if ANN is used directly then the size of the matrix after flattening will be huge to handle. Apart from this, ANN cannot provide required results if the orientation of the input image is changed a bit. CNN is considered as one of the most popular and effective algorithm for analyzing images due to its ability of picking up the pattern and use it to detect the required results. It consists of hidden layers known as convolutional layers which are specified with number of filters. These layers receive input from the previous layer. These filters are responsible for pattern detection. The filter is simply a matrix with a specified number of rows and columns which slides over the input image called stride. Dot product is done between the filter or weight and the pixels or bias of the input image at the hidden layer which results in pattern gathering and also indicates which neuron is to be activated by the activation function which is referred as threshold function. This output becomes input for the next layer. When the input image goes deep down into the network and passed through numerous filters, the model becomes capable of classifying complex things such as dog, cat and even pneumonia in lungs. Here, the proposed model is utilizing an activation function as well. This ensures non-linearity as the data/image passes through each layer. The absence of this activation function can result in the loss of required dimensionality.

II. RESEARCH WORKS

Artificial intelligence can be used to detect various diseases including pneumonia. Researchers have utilized different machine learning algorithms for analyzing medical images for the purpose of identifying and locating diseases. The research performed in the area of investigation of medical images and detection is explained in this section. The papers have been reviewed on the basis of pros & cons. of medical image detection.

[1] Gabruseva, Tatiana, Dmytro Poplavskiy, and Alexandr Kalinin., IEEE 2020: "Deep Learning For Automatic Pneumonia Detection." The key purpose of this proposed research paper is to offer a simple and operative algorithm for localizing the lungs blur region in the chest x-ray caused by the disease.

[2] Jain, Rachna, Preeti Nagrath, Gaurav Kataria, V. Sirish Kaushik, and D. Jude Hemanth, 2020: "Pneumonia Detection Using Convolution Neural Networks (CNNs). This paper mentions the building of four models and comparing them on the basis of their accuracy. The accuracy obtained are 89.74%, 85.26%, 92.31%, 91.67% respectively. In addition to accuracy, F1 and recall values are calculated to compare the models.

[3] Keval Shah, Veer Patel, Indraneel Sarmalkar, Suchetadevi Gaikwad, IRJET, 2022: "Pneumonia Detection using x-ray". This paper describes the construction of a model which can predict pneumonia in chest x-ray by applying machine learning and deep learning algorithms. The accuracy obtained here is 86%.

[4] Jiang, Zebin, IEEE (ICBAIE), 2020: "Chest X-Ray Pneumonia Detection Based On Convolution Neural Network. "This paper discusses about distinct variants of CNN such as InceptionResNet, VGG19 etc. The dataset taken here is very small in order to improve accuracy. The best accuracy obtained is 94.20%.

[5] Chouhan, Vikash, Sanjay Kumar Singh, Aditya Khamparia, Deepak Gupta, Prayag Tiwari, Catarina Moreira, Robertas Damaševičius, and Victor Hugo C. De Albuquerque, App. Sci., 2020: "Transfer Learning approach for Detection of Pneumonia". Here, the article discusses the use of transfer learning to identify pneumonia in chest x-ray. Architectures such as AlexNet and GoogLeNet are employed. The accuracy achieved is 96.4%.

III. EXISTING SYSTEM

Deep Learning in Artificial Intelligence has been developed majorly in recent years and the research & development in this domain is expanding day by day. In medical science, it is growing its legs towards the field of medical imaging system to detect and classify various kind of infections and diseases. This research work includes one of the use cases of these medical imaging system to detect pneumococcal infection using chest x-ray images. Many research and papers have been published in this area of DL. However, there are few limitations such as, some models have very less accuracy which is inapplicable in diagnosing the condition while others have used very less number of dataset to train the model which can affect its accuracy and scalability. Apart from this, the main issue of the priory available system is that GUI has not been integrated with the respective model thus making the model useless for clients who doesn't know how to operate the model.

IV. PROPOSED SYSTEM AND ARCHITECTURE

A. MODEL

The model is constructed using CNN by importing libraries namely numpy, open CV, matplotlib etc. The dataset is prepared such that 0 symbolizes normal condition whereas 1 represents pneumonic condition of the patient. The dataset/the chest x-ray images has been collected from Kaggle. The Normal category data is nearly half in numbers in comparison to the pneumococcal data. So, initially the dataset is found to be imbalance. The Normal Category was 1341 and the pneumococcal data was 3875. Therefore, data augmentation is done by "ImageDataGenerator" to overcome the overfitting problem. It can be defined by the large gap between errors in testing and training data. Thus, more data or images are created from the available dataset so as to prevent the overfitting. This is done by rotating, zooming and shearing of the dataset images which increased the normal category dataset to 3342 which is good to go with. Moreover, dropout layer is added in each layer. Recounting of both dataset is necessary to ensure that the training has been done on equal number of dataset from both categories. This is followed by pre-processing of data or images, such as resizing and adding label to each image. The label 0 is for Normal and 1 is for Pneumonia. Then the data is split into "Features" and "Target" variable as X and Y respectively. The target is labelled as 0 and 1 whereas the feature is the image converted to numpy array. Then Normalization (X/255) is done on the X variable to decrease the computational work. Since the dataset is prepared it is passed in the neural network model to train the model.

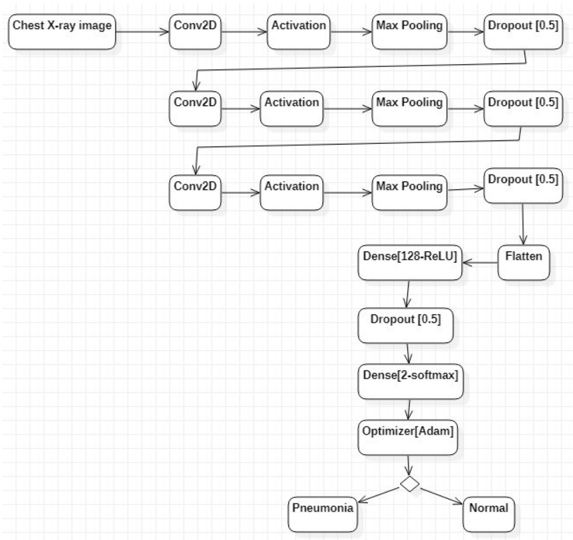


Fig. 1. Flowchart of the proposed system

The Neural Network consist of three convolution layer and each layer consist of a pooling layer and a dropout layer. MaxPooling of pool size of (2, 2), dropout of 0.5 and "ReLU" as activation function is used. The model is then flattened and a dense layer is added with 128 pixels along with the activation function "ReLU". Afterwards a dropout layer of 0.5 is added again to prevent overfitting by excluding the least wanted features. This is followed by a dense layer with an activation function "softmax" to fetch the probabilistic distribution on the output. Adam optimizer (Adaptive Moment Estimation) is used and a loss of "categorical crossentropy" for binary classification and a metrics of "accuracy". Adam optimizer is used as it is considered to provide a better result than any other optimization algorithm. Also, it requires very few parameters, has a faster computation time than others with a smaller learning rate, say 0.001. This faster pace is achieved by the following equation:

$$W_{t+1} = W_t - \alpha m_t$$

$$\text{Where, } m_t = \beta m_{t-1} + (1-\beta)\,[\delta L / \delta W_t]$$

The number of train and test data is 5773 and 1444 respectively. The training data is fitted into the model with an epochs of 15 and a validation split of 0.2. The accuracy, validation loss and validation accuracy is tested which came out to be 97.64, 6.89 and 97.23 respectively. The evaluation of the model is done to visualize the model's accuracy by plotting the line of training loss in a graph of loss versus epochs and reduction in loss is noticed with increase in epochs. The loss is found to be 7.58 percent. Similarly, the line is plotted for training accuracy in accuracy versus epochs graph and then it is speculated that the accuracy of the model started increasing w.r.t the epochs. The accuracy of 97.30 percent is achieved. Then the model is saved and tested with a pneumococcal x-ray and the model detected pneumonia with 87% probability.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 118, 118, 32)	896
activation (Activation)	(None, 118, 118, 32)	0
max_pooling2d (MaxPooling2D)	(None, 59, 59, 32)	0
dropout (Dropout)	(None, 59, 59, 32)	0
conv2d_1 (Conv2D)	(None, 57, 57, 32)	9248
activation_1 (Activation)	(None, 57, 57, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 28, 28, 32)	0
dropout_1 (Dropout)	(None, 28, 28, 32)	0
conv2d_2 (Conv2D)	(None, 26, 26, 64)	18496
activation_2 (Activation)	(None, 26, 26, 64)	0
max_pooling2d_2 (MaxPooling2	(None, 13, 13, 64)	0
dropout_2 (Dropout)	(None, 13, 13, 64)	0
flatten (Flatten)	(None, 10816)	0
dense (Dense)	(None, 128)	1384576
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

Total params: 1,413,474
Trainable params: 1,413,474
Non-trainable params: 0

Fig. 2. Various layers extracting features

B. GUI AND WEBSITE

The website is developed for easing the application of the model for the user. The user can upload the chest x-ray image in the required space and the prediction is received instantly. The website is built using HTML, CSS and JavaScript as frontend and Python Flask as backend where the model's h5 file is added to integrate the model with the website. This provides a user-friendly interface to detect pneumonia in the patient's chest x-ray in the form of percentage.

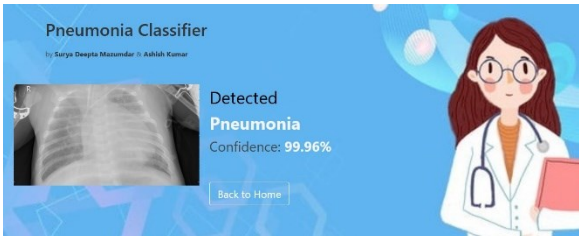


Fig. 3. Website for the model

V. APPLIED ALGORITHM

Why CNN?

In image recognition and processing, convolutional neural networks (CNNs) are commonly used for recognizing patterns in images. With CNN, neurons in one layer are linked solely to a tiny area of neurons in the prior layer in spite of being completely linked.

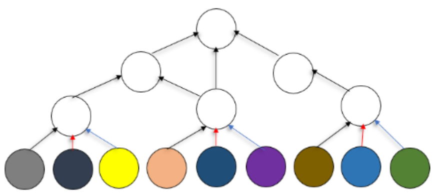


Fig. 4. CNN node representation

By observing the third layer in the above picture, it is spotted that each neuron is linked to three other neurons whereas in a fully connected network, every neuron is linked with every other neurons, thus requiring less processing power.

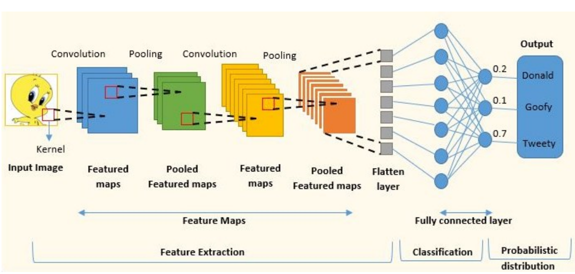


Fig. 5. Convolutional layers

There are various constituent in CNN namely: -

- Kernel=Filter=Feature Detector: It is nothing but a filter that is used to extract the features from the images.
- Stride: Then the kernel is moved across the image with the provided stride value.
- Padding: Then a padding layer is added to the stride output with a value 0 to all the sides.
- Pooling: It is of two types namely Max Pooling and Average Pooling.
- Flatten: Then finally the pooled output is changed into linear vector.
- Dense: Then the flattened linear vector is passed through the deeply connected CNN network.
- Dropout: It is used to prevent overfitting by nullifying the value of some neuron.
- Rectified Linear Unit(ReLU):-

  f(x)=x, x>=0

  f(x)=0, x<0[f(x)=max(0,x)]
- Softmax:-$z_i = \exp(z_i)/\Sigma_j \exp(z_j)$

VI. SAMPLE DATASET

The dataset extracted from "Kaggle" repository called "Chest X-Ray Images (Pneumonia)" is used for this study. The data set comprises of two classes. One is normal lung and the other is pneumococcal lung which can be recognized as the smokiness effect in the lung's section. This effect can be interpreted as affected lungs whereas the X-ray image with a clear picture depicts the normal lung condition.The pictorial representation of the sample dataset is shown below:

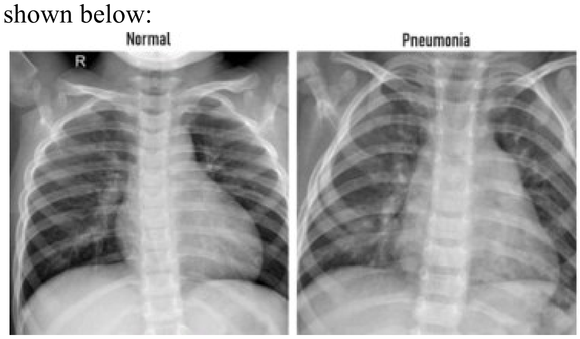


Fig. 6. Pictorial representation of dataset

Initially the dataset gathered was imbalance, the Normal data was 1341 and the pneumococcal data was 3875. So, data augmentation is done on the Normal data to overcome the overfitting problem. After resizing, rotating and shearing of previously available dataset, the Normal data became 3342 which is good to go with. Preprocessing of data (chest x-ray images) is done, such as resizing and adding label to each images. The label 0 is for Normal and 1 is for Pneumococcal lungs. This is followed by splitting the data into features and target variables X and Y respectively. The target is the level (0 or 1) whereas the feature is the converted numpy array of the image. Then Normalization (X/255) is done on the X variable to decrease the computational work. After preparing the dataset, the dataset is passed in the neural network to train the model.

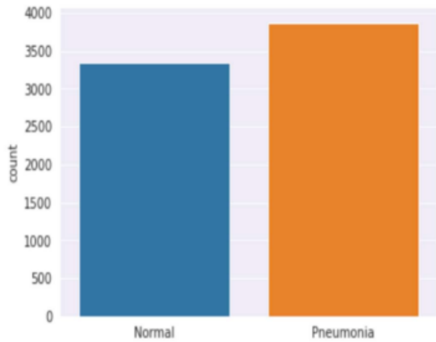


Fig. 7. Dataset count

VII. RESULT AND DISCUSSION

In this model, Convolution Neural Network which is a deep learning algorithm has been employed for classifying the chest x-ray images into two distinct classes either pneumonia diagnosed or normal.

In this, the CNN model consists of three Convolution layers. Each layer comprises of a pooling layer and a dropout layer. Also, an activation function has been applied in each layer which is “ReLU”. The accuracy & loss of the proposed model obtained are:

- Accuracy: Accuracy for both training & validation is being increased together which depicts that the model is a good fit. Accuracy for training & validation is obtained as 97.64% & 97.23% respectively.
- Loss: Loss for both training & validation is being decreased together which depicts that the model is good fit. Loss for both training & validation is obtained as 0.0682 & 0.0689 respectively.

Testing Percentage: The main dataset is divided into training and testing dataset. The testing dataset is the one which is not trained on the model but are used for testing the accuracy. The testing percentage of the model is 20% or 0.2. The number of train data and test data is 5773 and 1444 respectively which is in the ratio of 80:20. Thus, the training dataset is larger compared to that of the test data to increase the learning rate of the model and also to gain its usability by the training dataset which is fed to it. So, the initial step is to feed the training dataset to the model then the training data is further tagged with features and target variables. Finally the model is tested using the test data to ensure its efficiency.

```
from sklearn.model_selection import train_test_split
train_data,test_data,train_target,test_target=train_test_split(X,Y,test_size=0.2)
```

```
train data : 5773
test data : 1444
```

Fig. 8. Splitting the data into train and test

In the following figure (Fig. 11), the accuracy vs epochs graph shows increase in accuracy with increase in epochs whereas in the loss vs epochs graph, loss value decreases with increase in epochs.

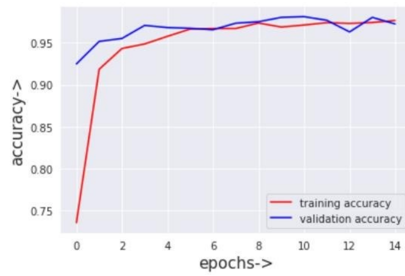
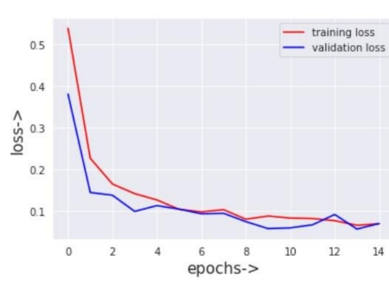


Fig. 9. Accuracy & Loss vs epochs graph

In a neural network an epoch is a single cycle of training with all the training dataset. This data is used exactly once in an epoch. A part of the dataset is used to train the neural network in each epoch, consisting of a forward and a backward pass or in a to and fro cycle. The model is trained on 15 epochs which is represented in fig. 12. As the epochs increases, the accuracy of the model along with the learning rate also increases. On the other hand, the loss in the model decreases. When epochs value reaches 15, the accuracy becomes optimum because after that point the accuracy of the model fell down which causes the overfitting issue to arise. The model has received a loss of 0.0682, accuracy of 0.9764, validation loss of 0.0689 and a validation accuracy of 0.9723 at epochs value of 15. A verbose of 1 is used to get a complete view of the validation, accuracy and the loss in the model.



```
Epoch 1/15
145/145 [==============================] - 90s 612ms/step - loss: 0.5390 - accuracy: 0.7358 - val_loss: 0.3805 - val_accuracy: 0.9247
Epoch 2/15
145/145 [==============================] - 88s 610ms/step - loss: 0.2260 - accuracy: 0.9184 - val_loss: 0.1437 - val_accuracy: 0.9515
Epoch 3/15
145/145 [==============================] - 89s 611ms/step - loss: 0.1639 - accuracy: 0.9430 - val_loss: 0.1369 - val_accuracy: 0.9550
Epoch 4/15
145/145 [==============================] - 89s 613ms/step - loss: 0.1408 - accuracy: 0.9485 - val_loss: 0.0980 - val_accuracy: 0.9706
Epoch 5/15
145/145 [==============================] - 88s 608ms/step - loss: 0.1256 - accuracy: 0.9576 - val_loss: 0.1120 - val_accuracy: 0.9680
Epoch 6/15
145/145 [==============================] - 88s 606ms/step - loss: 0.1030 - accuracy: 0.9662 - val_loss: 0.1035 - val_accuracy: 0.9671
Epoch 7/15
145/145 [==============================] - 88s 606ms/step - loss: 0.0968 - accuracy: 0.9669 - val_loss: 0.0923 - val_accuracy: 0.9654
Epoch 8/15
145/145 [==============================] - 89s 612ms/step - loss: 0.1021 - accuracy: 0.9669 - val_loss: 0.0935 - val_accuracy: 0.9732
Epoch 9/15
145/145 [==============================] - 89s 618ms/step - loss: 0.0792 - accuracy: 0.9734 - val_loss: 0.0734 - val_accuracy: 0.9749
Epoch 10/15
145/145 [==============================] - 88s 606ms/step - loss: 0.0869 - accuracy: 0.9686 - val_loss: 0.0567 - val_accuracy: 0.9801
Epoch 11/15
145/145 [==============================] - 88s 605ms/step - loss: 0.0819 - accuracy: 0.9710 - val_loss: 0.0584 - val_accuracy: 0.9810
Epoch 12/15
145/145 [==============================] - 88s 609ms/step - loss: 0.0809 - accuracy: 0.9738 - val_loss: 0.0657 - val_accuracy: 0.9766
Epoch 13/15
145/145 [==============================] - 88s 605ms/step - loss: 0.0756 - accuracy: 0.9729 - val_loss: 0.0906 - val_accuracy: 0.9628
Epoch 14/15
145/145 [==============================] - 90s 619ms/step - loss: 0.0647 - accuracy: 0.9740 - val_loss: 0.0555 - val_accuracy: 0.9801
Epoch 15/15
145/145 [==============================] - 88s 608ms/step - loss: 0.0682 - accuracy: 0.9764 - val_loss: 0.0689 - val_accuracy: 0.9723
```

Fig. 10. Training the dataset on the model

VIII. CONCLUSION

This project aims to provide a facility for people to detect pneumonia especially in areas where the availability of radiotherapists is very little. The model is integrated to a user-friendly website which makes it easier for people to use it. For website, HTML, CSS and JavaScript is used. To connect the model with it, python flask is chosen. The website created not only represents the lung's condition but also shows the necessary steps to deal with it and provides the pre-requisites needed to cure such syndrome. CNN algorithm has provided the best accuracy among the algorithms employed for classification. DenseNet-121 is used at the dense layer of the architecture. The accuracy obtained is 97.63% which is pretty high as compared to available models which makes it comparatively more accurate model. Even after obtaining such impressive results, there are some restrictions such as, only the front side of the chest x-ray image can be used since the model is trained on such datasets only whereas pneumonia can be detected using the side profile of the chest x-ray as well.

IX. REFERENCES

[1] Chakraborty, Sabyasachi, Satyabrata Aich, Jong Seong Sim, and Hee-Cheol Kim. "Detection of pneumonia from chest x-rays using a convolutional neural network architecture." In *International conference on future information & communication engineering*, vol. 11, no. 1, pp. 98-102. 2019.

[2] Sharma, Ritik, Sugandhi Midha, and Amit Semwal. "Predictive Analysis on Multimodal Medicare Application." In *2022 International Conference on Cyber Resilience (ICCR)*, pp. 1-5. IEEE, 2022.

[3] Rajpurkar, Pranav, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning." *arXiv preprint arXiv:1711.05225* (2017).

[4] Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network." In *2017 international conference on engineering and technology (ICET)*, pp. 1-6. Ieee, 2017.

[5] Sharma, Harsh, Jai Sethia Jain, Priti Bansal, and Sumit Gupta. "Feature extraction and classification of chest x-ray images using cnn to detect pneumonia." In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 227-231. IEEE, 2020.

[6] Jain, Rachna, Preeti Nagrath, Gaurav Kataria, V. Sirish Kaushik, and D. Jude Hemanth. "Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning." *Measurement* 165 (2020): 108046.

[7] Rahman, Tawsifur, Muhammad EH Chowdhury, Amith Khandakar, Khandaker R. Islam, Khandaker F. Islam, Zaid B. Mahbub, Muhammad A. Kadir, and Saad Kashem. "Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray." *Applied Sciences* 10, no. 9 (2020): 3233.

[8] Sebastian, Neenu, and B. Ankayarkanni. "Deep Learning applications to detect pneumonia on chest X-ray: A systematic study." In *2022 International Conference on Advanced Computing Technologies and Applications (ICACTA)*, pp. 1-5. IEEE, 2022.

[9] Gabruseva, Tatiana, Dmytro Poplavskiy, and Alexandr Kalinin. "Deep learning for automatic pneumonia detection." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 350-351. 2020.

[10] Labhane, Gaurav, Rutuja Pansare, Saumil Maheshwari, Ritu Tiwari, and Anupam Shukla. "Detection of pediatric pneumonia from chest X-ray images using CNN and transfer learning." In *2020 3rd international conference on emerging technologies in computer engineering: machine learning and internet of things (ICETCE)*, pp. 85-92. IEEE, 2020.

[11] Jiang, Zebin. "Chest X-ray pneumonia detection based on convolutional neural networks." In *2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pp. 341-344. IEEE, 2020.

[12] Chouhan, Vikash, Sanjay Kumar Singh, Aditya Khamparia, Deepak Gupta, Prayag Tiwari, Catarina Moreira, Robertas Damaševičius, and Victor Hugo C. De Albuquerque. "A novel transfer learning based approach for pneumonia detection in chest X-ray images." *Applied Sciences* 10, no. 2 (2020): 559.

[13] Varshni, Dimpy, Kartik Thakral, Lucky Agarwal, Rahul Nijhawan, and Ankush Mittal. "Pneumonia detection using CNN based feature extraction." In *2019 IEEE international conference on electrical, computer and communication technologies (ICECCT)*, pp. 1-7. IEEE, 2019.

[14] Hashmi, Mohammad Farukh, Satyarth Katiyar, Avinash G. Keskar, Neeraj Dhanraj Bokde, and Zong Woo Geem. "Efficient pneumonia detection in chest xray images using deep transfer learning." *Diagnostics* 10, no. 6 (2020): 417.

[15] Shah, Shanay, Heeket Mehta, and Pankaj Sonawane. "Pneumonia detection using convolutional neural networks." In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 933-939. IEEE, 2020.