Feature Extraction and Classification of Chest X-Ray Images Using CNN to Detect Pneumonia

[1]Harsh Sharma, [2]Jai Sethia Jain, [3]Priti Bansal
Department of IT
[1,2]NSIT, [3] NSUT
New Delhi, India
harshsharma10998@gmail.com, jai200498@gmail.com,
bansalpriti79@gmail.com

Sumit Gupta
Department of Computer Science and Engineering
Amity Institute of Engineering and Technology
Noida, Uttar Pradesh
skumar59@amity.edu

***Abstract*—Pneumonia is an infection that causes inflammation of lungs and can be deadly if not detected on time. The commonly used method to detect Pneumonia is using chest X-ray which requires careful examination of chest X-ray images by an expert. The method of detecting pneumonia using chest X-ray images by an expert is time-consuming and less accurate. In this paper, we propose different deep convolution neural network (CNN) architectures to extract features from images of chest X-ray and classify the images to detect if a person has pneumonia. To evaluate the effect of dataset size on the performance of CNN, we train the proposed CNN's using both the original as well as augmented dataset and the results are reported.**

Keywords—Pneumonia; Classification; Chest X-Ray; Convolution Neural Network;

I. INTRODUCTION

Pneumonia causes inflammation of lungs especially air sacs which may filled with fluid or pus causing cough and difficulty in breathing. More than 150 million people especially children below 5 years get infected with pneumonia yearly world-wide [1]. The mortality rate due to pneumonia is specifically higher in children below five years in developing nations. This mandates the detection of pneumonia on time so that proper treatment can be provided to the person. The most practiced method world-wide to detect pneumonia is using chest X-ray images. To detect pneumonia, careful examination of chest X-ray images is required which in turn necessitate experienced and knowledgeable radiologist/ experts. This makes the process of pneumonia detection a challenging task. Moreover, it is a time-consuming task and a little error can have fatal consequences.

Due to the tedious nature of X-ray image analysis task, many computer algorithms [2, 3] and computer aided diagnostic tools [4] have been proposed by researchers in order to analyze X-ray images; however, they were not proved very significant in assisting experts in making decisions [5]. Recently, handcrafted techniques and deep learning techniques are being applied successfully by researchers in the area of medical imaging to analyze and classify medical images for detection of various diseases like skin cancer [6], breast cancer [7], tuberculosis [8], brain tumor [9] etc. This motivated us to propose a deep convolution neural network (CNN) that helps to extract features from chest X-ray image. The extracted features are further used to classify the images to detect whether a person has pneumonia or not, in less time and with greater accuracy as compared to manual methods.

CNN is a subclass of deep neural networks that has attained considerable success in computer vision domain for example, image segmentation [3], image classification [6, 8], object detection [10] etc. A CNN comprises of convolution layers, pooling layers and a fully-connected layer. Various models such as AlexNet, LeNet, VGGNet, Inception etc. exist which are pre-trained on millions of images and can be used for image classification using transfer learning. The disadvantages of pre-trained models are that they are very large with millions of trainable parameters which requires lots of computing power and are time-consuming [11]. When the dataset size is small, these models may overfit the training data resulting in poor classification accuracy. This motivated us to build a CNN architecture from scratch instead of using transfer learning. In this paper, we propose various CNN architectures and these are trained on chest X-ray image dataset [12].

The paper is organized as follows. Section II presents the related work. Section III presents the proposed architecture to classify images using CNN to detect pneumonia. In Section IV, experimental results are shown and discussed. Finally, conclusion and future work are presented in Section V.

II. RELATED WORK

Several methods exist in literature that can help to detect pneumonia using chest X-ray images. Some of them uses handcrafted feature extraction techniques along with machine learning algorithm to classify chest X-ray whereas others use deep learning techniques for feature extraction and classification.

In [13], a prototype Pneumo-CAD based on computer-aided diagnostic (CAD) and wavelet transformation was proposed to diagnose pneumonia in children. It was observed that Pneumo-CAD achieved the best accuracy when using Haar wavelet. Rajpurkar et al. [14] proposed an algorithm CheXNet, a 121-layer CNN to classify 14 different diseases including pneumonia seen on the chest X-ray. CheXNet was trained using Chest X-ray14 dataset. The performance of CheXNet was compared to that of radiologists and it was found that CheXNet surpasses average radiologist performance on F1 metric. In [15], transfer learning was used to classify pediatric chest X-ray images into normal or pneumonia-infected images. They were further classified into two categories: pneumonia caused by bacteria and pneumonia caused by virus. In [16], Jaiswal et al. proposed a model based on Mask-RCCN to detect pneumonia from chest X-ray images. In [11], Stephen et al. built a CNN model from scratch for classification and detection of pneumonia from chest X-ray images. To avoid overfitting as well as to reduce generalization error because of the small size of dataset, they incorporated various data augmentation methods, variation in learning rate and annealing.

III. PROPOSED ARCHITECTURE

In this section, first we present the proposed architectures to classify chest X-ray images to detect pneumonia. It is a binary classification problem. Secondly, we give a brief outline of the dataset used in this paper and the various steps which have been applied before training the proposed architectures.

A. ARCHITECTURE

Here, we present two CNN architectures- one with a dropout layer and another without a dropout layer. Both CNN consist of convolution layer, max pooling and a classification layer. A series of convolution and max-pooling layers act as a feature extractor that is divided into two parts. The first part consists of two Convolution layers with 32 - 32 units each along with a max-pooling layer of size 3×3 and a Relu activator. While the other also has two Convolution layers but with 64 and 128 units respectively along with a max-pooling layer of size 2×2 and a Relu activator. Relu is a popular activation function which is generally used in neural networks especially in CNNs. Relu layer introduces nonlinearity to the model.

Features extracted from the feature extractor part of the CNN are given as input to the dense layer which classifies the image. Before feeding the extracted features to the dense layer, a flatten layer a used. As the dense layer takes 1-Dimensional input, hence, flatten layer flattens the feature data and gives a 1-Dimensional output which is fed to the dense layer.

While training a CNN it might be possible that output through a certain layer is more dependent on a few selected neural units. To reduce this dependency and prevent overfitting, the concept of dropout is introduced. During training, in each epoch, a neuron is momentarily dropped with a dropout probability p. Due to this, all the inputs and outputs to this neuron become disabled in the current epoch which results in some loss of data enhancing regularization in the model so that at times it would predict with much higher accuracy. The dropped-out neurons are resampled with probability p at every training step, so a dropped-out neuron at one step may become active in the next step. A dropout probability of 0.5, corresponds to 50% of the neurons being dropped out. In the proposed CNN architecture with dropout layer, we apply dropout at two places. First, it is applied at the feature extractor part i.e. after convolution and max-pool layers. As, the convolutional layers have not too many parameters, hence overfitting is not an issue in this case. Hence, here we take a low drop probability of 0.2. Second, it is used at dense layer. Dropout in the lower layers helps because it provides noisy inputs for the higher fully connected layers which prevents them from overfitting. At dense layers a drop probability of usually 0.5 is used as it has been observed that dense probability of 0.5 gives best regularization in most of the cases [17]. We also use a drop probability of 0.5 here. The model summary of CNN with and without dropout is shown in Fig. 1 and Fig. 2 respectively. The architecture of CNN with dropout is shown in Fig. 3.

B. DATASET

The dataset employed in this work is picked from Kaggle [12]. It consists of 5863 images of chest X-ray. The dataset is split into three portions namely training dataset, validation dataset and testing dataset. The details of number of images under each part is shown in Table I. Some sample chest X-ray images are given in Fig. 4.

TABLE I. DATASET INFORMATION

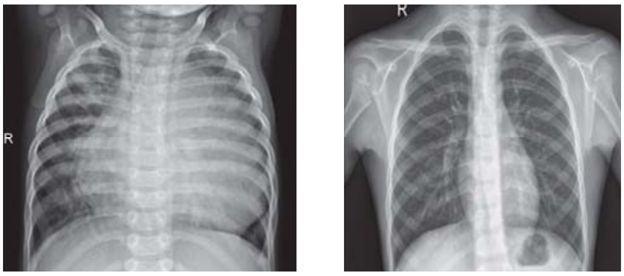
Dataset	Number of Normal Images	Number of Pneumonia Images
Training dataset	1341	3875
Validation dataset	72	119
Testing Dataset	170	279

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 32)        896
_________________________________________________________________
activation_1 (Activation)    (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 62, 62, 32)        9248
_________________________________________________________________
activation_2 (Activation)    (None, 62, 62, 32)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 21, 21, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 21, 21, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 21, 21, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 21, 21, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 20, 20, 128)       32896
_________________________________________________________________
activation_4 (Activation)    (None, 20, 20, 128)       0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 10, 10, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 10, 10, 128)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 12800)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               3277056
_________________________________________________________________
activation_5 (Activation)    (None, 256)               0
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 512)               131584
_________________________________________________________________
activation_6 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_4 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 513
_________________________________________________________________
activation_7 (Activation)    (None, 1)                 0
=================================================================
Total params: 3,470,689
Trainable params: 3,470,689
Non-trainable params: 0
_________________________________________________________________
```

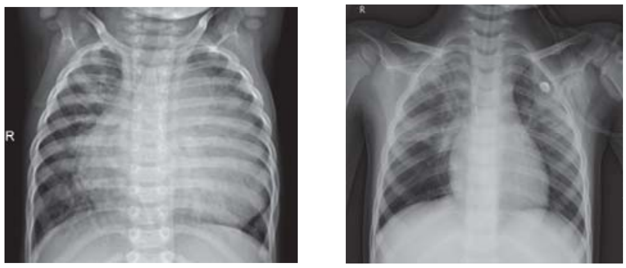
Fig. 1 Model summary of CNN with dropout

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 32)        896
_________________________________________________________________
activation_1 (Activation)    (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 62, 62, 32)        9248
_________________________________________________________________
activation_2 (Activation)    (None, 62, 62, 32)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 21, 21, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 21, 21, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 21, 21, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 20, 20, 128)       32896
_________________________________________________________________
activation_4 (Activation)    (None, 20, 20, 128)       0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 10, 10, 128)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 12800)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               3277056
_________________________________________________________________
activation_5 (Activation)    (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 512)               131584
_________________________________________________________________
activation_6 (Activation)    (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 513
_________________________________________________________________
activation_7 (Activation)    (None, 1)                 0
=================================================================
Total params: 3,470,689
Trainable params: 3,470,689
Non-trainable params: 0
```

Fig. 2 Model summary of CNN without dropout



(a)



(b)

Fig. 4 Chest X-ray images (a) without pneumonia (b) with pneumonia

Initially, the images in the dataset are of varying sizes. However, all the input images given to CNN should be of same size. To solve this problem, all the images in the dataset are resized to 64 × 64. To avoid overfitting and to enhance the generalization capabilities of the proposed CNN architectures, various in-place data augmentation techniques such as rescaling, flipping etc. are used in this paper. Use of this type of data augmentation ensures that our CNN see new variations of data at each and every epoch during training. The various data augmentation techniques used in this work are listed in Table II.

TABLE II. DATA AUGMENTATION TECHNIQUES

Data Augmentation Techniques	Values
Rescale	1.0 / 255
Rotation Range	20
Zoom Range	0.2
Width Shift Range	0.2
Height Shift Range	0.2
Shear Range	0.2
Horizontal Flip	True
Fill Mode	Nearest

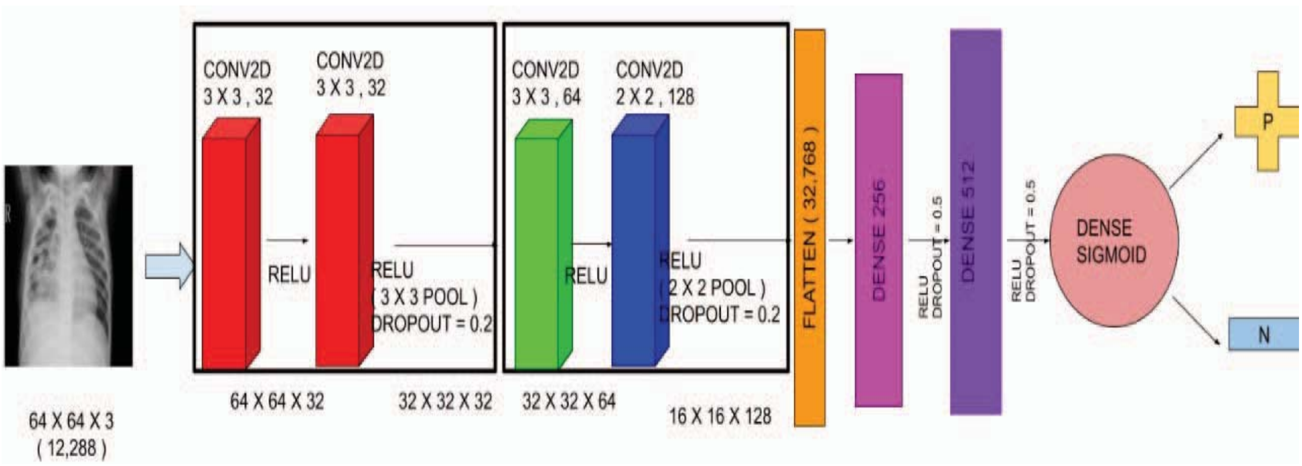


Fig. 3 The Proposed Architecture with Dropout

IV. EXPERIMENTAL RESULTS

To assess the consequences of data augmentation methods on the performance of proposed CNN architectures, we trained the two CNN's with the original dataset as well as the augmented dataset. The detail of CNN's with the type of dataset used for training them are given in Table III.

TABLE III. TYPES OF MODEL

Model	Dataset and CNN architecture
Model 1	With Augmentation, With Dropout
Model 2	With Augmentation, Without Dropout
Model 3	Without Augmentation, With Dropout
Model 4	Without Augmentation, Without Dropout

The four models are trained for 20 epochs with Adam optimizer and learning rate is set to 0.0001. The batch size is 32. During training and validation, loss is calculated using binary cross-entropy error as given in (1). Training accuracy and validation accuracy is calculated using (2).

$$Loss = -\frac{1}{N}\sum_{j=1}^{N}(y^j \log \hat{y}^j + (1-y^j)\log(1-\hat{y}^j)) \quad (1)$$

Where, y^j and $\hat{y}^j$ are the target and output values respectively of i^{th} sample of training/validation dataset and N represents the total images present in the training/validation dataset.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Where, TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative. The plot of loss and accuracy of all the four proposed models on the training set as well as the validation set over the training epochs are shown in Fig. 5 - Fig. 8 respectively. The test accuracy of the four models calculated using (2) are presented in Table IV. It is evident from Table IV that although Model 1 having dropout and trained using augmented data has higher loss and lower training accuracy as compared to other models but it performs better on validation data and testing data as compared to others. This is due to the fact that the other models are overfitted to the training data.

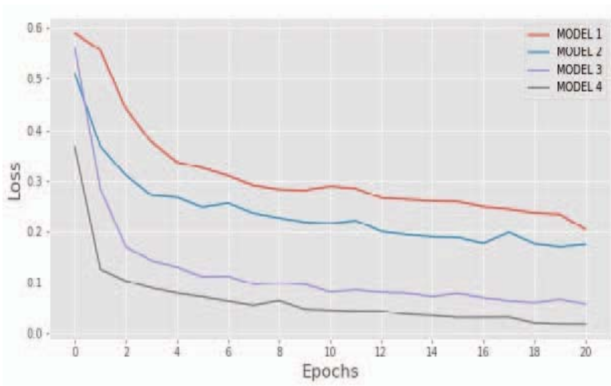


Fig. 5 Plot of training loss

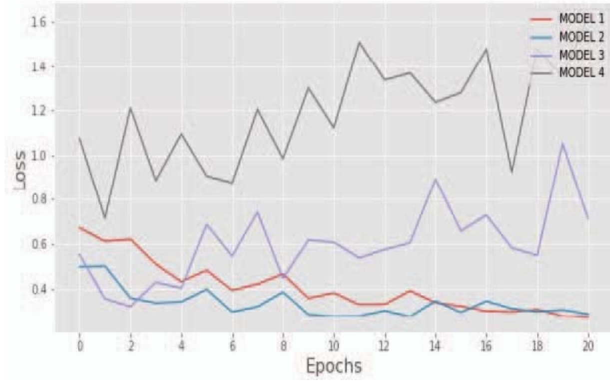


Fig. 6 Plot of validation loss

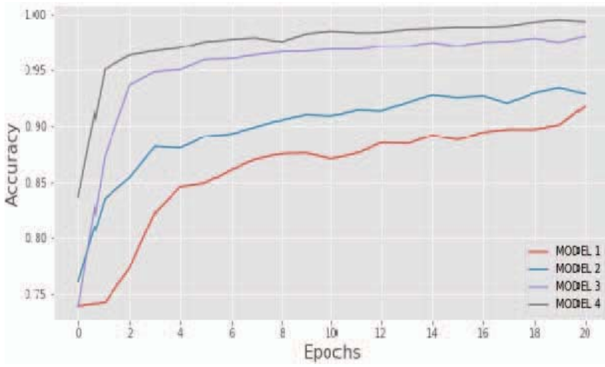


Fig. 7 Plot of training accuracy

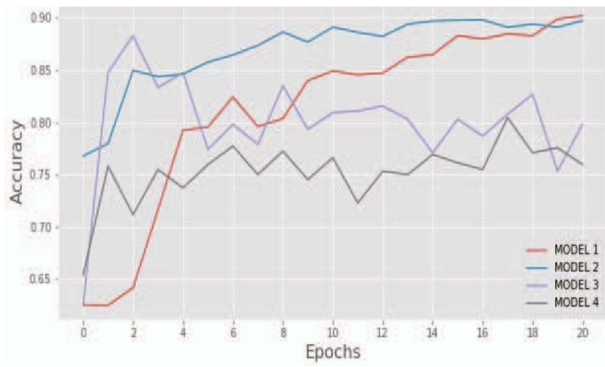


Fig. 8 Plot of validation accuracy

TABLE IV. TESTING ACCURACY OF DIFFERENT MODELS

Model	Testing Accuracy
Model 1	0.9068
Model 2	0.8932
Model 3	0.7980
Model 4	0.7498

V. CONCLUSION AND FUTURE WORK

In this paper, we propose two CNN architectures that are designed from scratch to detect pneumonia from images of chest X-ray. To avoid overfitting, data augmentation techniques are used. The result of experiments performed to assess the performance of the proposed architectures and the effect of data augmentation on the performance of the proposed CNN's show that CNN with dropout trained on augmented data outperforms the other models.

In future, we plan to use different optimizers and other data augmentation techniques in an attempt to further improve the classification accuracy of the proposed CNN architecture with data augmentation. We also plan to use early stopping and batch normalization instead of dropout layer to see their effect in avoiding overfitting.

REFERENCES

[1] I. Rudan, L. Tomaskovic, C. Boschi-Pinto, and H. Campbell, "Global estimate of the incidence of clinical pneumonia among children under five years of age," Bulletin of the World Health Organization, vol. 82, pp. 85–903, 2004.

[2] U. Avni, H. Greenspan, E. Konen, M. Sharon, and J. Goldberger, "X-ray categorization and retrieval on the organ and pathology level, using patch-based visual words," IEEE Trans. Med. Imaging, 30 (3), pp. 733–746, 2011.

[3] P. Pattrapisetwong, and W. Chiracharit, "Automatic lung segmentation in chest radiographs using shadow filter and multilevel thresholding," International Computer Science and Engineering Conference (ICSEC), IEEE, 2016, pp. 1–6.

[4] Q. Li, and R.M. Nishikawa, "Computer-aided Detection and Diagnosis in Medical Imaging," Taylor & Francis, 2015.

[5] A. A. El-Solh, C. -B. Hsiao, S. Goodnough, J. Serghani, and B. J. Grant, "Predicting active pulmonary tuberculosis using an artificial neural network," Chest, 116 (4), pp. 968–973, 1999.

[6] J. A. A. Salido and C. Jr. Ruiz, "Using Deep Learning to Detect Melanoma in Dermoscopy Images," International Journal of Machine Learning and Computing, 8(1), 2018.

[7] D. A. Ragab, et al., "Breast cancer detection using deep convolutional neural networks and support vector machines," PeerJ 7:e6201, 2018.

[8] P. Lakhani and B. Sundaram, "Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks," Radiology, 284(2), pp. 574-582, 2017.

[9] X. W. Gao, R. Hui, and Z. Tian, "Classification of CT brain images based on deep learning networks," Computer Methods and Programs in Biomedicine, 138, pp. 49-56, 2017.

[10] A. R. Pathaka, M. Pandeya and S. Rautaraya,"Application of Deep Learning for Object Detection," ICCIDS, vol.132, pp.1706–1717, 2018. doi:10.1016/j.procs.2018.05.14

[11] O. Stephen, M. Sain, U. J. Maduh, and Do-Un Jeong, "An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare", Hindawi, Journal of Healthcare Engineering, 2019.

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

[13] L. L. G. Oliveira, S. A. E. Silva, L. H. V. Ribeiro, R. M. De Oliveira, C. J. Coelho, and A. L. S. Ana Lúcia,"Computer-Aided Diagnosis in Chest Radiography for Detection of Childhood Pneumonia," Int. J. Med. Inform, 77, pp. 555–564, 2008.

[14] P. Rajpurkar et al.,''CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning,'' 2017. https://arxiv.org/abs/1711.05225

[15] D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, F.; et al., "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," Cell, 172, pp. 1122–1131, 2018.

[16] A. K. Jaiswal, P. Tiwari , S. Kumar, D. Gupta, A. Khanna, and J. J. P. C. Rodrigues, "Identifying pneumonia in chest X-rays: A deep learning approach:, Measurement, 145, pp. 511-518, 2019.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting,"Journal of Machine Learning Research, 15(1), pp. 1929–1958, 2014.