Detection Of Pneumonia Using Convolutional Neural Networks



**ABSTRACT**

We develop an algorithm that can detect pneumonia from chest X-rays at a very good level and might be better or at least at par with practicing radiologists. Our algorithm,works on convolutional neural networks which contains three convolutional layers with 64, 128 and 216 number of feature detectors in each convolutional layer and 512 units in the hidden layer followed by the output layer with one node. Our model is trained on the dataset taken from https://data.mendeley.com/datasets/rscbjbr9sj/2 , which has Chest X-ray images (anterior-posterior) selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children’s Medical Center, Guangzhou.

**Chapter 1**

**INTRODUCTION**

* 1. **Overview**

Pneumonia is an infection that inflames the air sacs in one or both lungs. The air sacs

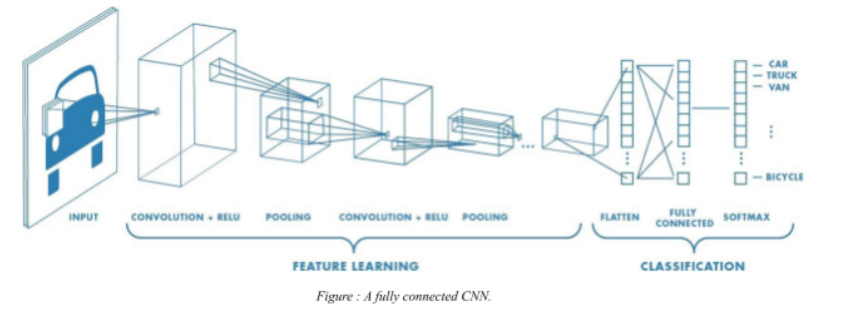
may fill with fluid or pus (purulent material), causing cough with phlegm or pus, fever, chills, and difficulty breathing. A variety of organisms, including bacteria, viruses and fungi, can cause pneumonia. Pneumonia can range in seriousness from mild to life- threatening. It is most serious for infants and young children, people older than age 65, and people with health problems or weakened immune systems.

According to WHO, pneumonia accounts for 15% of all deaths of children under 5 years old, killing 808 694 children in 2017. Pneumonia in India accounts for 20 percent of the deaths worldwide. According to the WHO, one in three deaths in India is caused by pneumonia. Every year almost 200,000 children under five die of pneumonia in India. A child dies every 2 min of pneumonia, diarrhoea in India. On a global level, pneumonia kills around 900,000 children in the world every year.

Convolution is a mathematical concept used heavily in Digital Signal Processing when dealing with signals that take the form of a time series. In lay terms, convolution is a mechanism to combine or “blend”[10] two functions of time 3 in a coherent manner.

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

An image is nothing but a matrix of pixel values, right? So why not just flatten the image (e.g. 3x3 image matrix into a 9x1 vector) and feed it to a Multi-Level Perceptron for classification purposes? In cases of extremely basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout. A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better

CNNs are usually applied to image data. Every image is a matrix of pixel values. The range of values that can be encoded in each pixel depends upon its bit size. Most commonly, we have 8 bit or 1 Byte-sized pixels. Thus the possible range of values a single pixel can represent is [0, 255]. However, with coloured images, particularly RGB (Red, Green, Blue)-based images, the presence of separate colour channels (3 in the case of RGB images) introduces an additional ‘depth’ field to the data, making the input 3-dimensional. Hence, for a given RGB image of size, say 255×255 (Width x Height) pixels, we’ll have 3 matrices associated with each image, one for each of the colour channels. Thus the image in its entirety, constitutes a 3-dimensional structure called the Input Volume (255x255x3). In the figure, we have an RGB image which has been separated by its three color planes — Red, Green, and Blue. There are a number of such color spaces in which images exist — Grayscale, RGB, HSV, CMYK, etc. You can imagine how computationally intensive things would get once the images reach dimensions, say 8K (7680×4320). The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets.

Kernel Operations The exact procedure for convolving a Kernel (say, of size 16 x 16) with the input volume (a 256 x 256 x 3 sized RGB image in our case) involves taking patches from the input image of size equal to that of the kernel (16 x 16), and convolving (or calculating the dot product) between the values in the patch and those in the kernel matrix. The convolved value obtained by summing the resultant terms from the dot product forms a single entry in the activation matrix. The patch selection is then slided (towards the right, or downwards when the boundary of the matrix is reached) by a certain amount called the ‘stride’ value, and the process is repeated until the entire input image has been processed. The process is carried out for all colour channels. For normalization purposes, we divide the calculated value of the activation matrix by the sum of values in the kernel matrix. The process is demonstrated in the figure below , using a toy example consisting of a 3-channel 4×4-pixels input image and a 3×3 kernel matrix.

Note that:

● pixels are numbered from 1 in the example.

● the values in the activation map are normalized to ensure the same intensity range between the input volume and the output volume . Hence, for normalization, we divide the calculated value for the ‘red’ channel by 2 (the sum of values in the kernel matrix).

● we assume the same kernel matrix for all three channels, but it is possible to have a separate kernel matrix for each colour channel.



Dept. of CSE, KIT, Tiptur Page 6

Detection Of Pneumonia Using Convolutional Neural Networks



**Chapter 2**

**LITERATURE SURVEY**

Many researchers have tackled the problem of classifying images with high accuracy. Here are some citations related to our paper: Rubin et al. [6] developed a CNN model to detect common thorax disease from frontal and lateral chest X-ray images. MIMIC-CXR dataset was used to perform large-scale automated recognition of these images. The dataset was split into training, testing and validation sets as 70%, 20% and 10%, respectively. Data augmentation and pixel normalization were used to improve overall performance. Their DualNet CNN model achieved an average AUC of 0.72 and 0.688 for PA and AP, respectively. A deep convolutional neural network to classify pulmonary tuberculosis was developed by Lakhani et al. [7]. Transfer learning models such as AlexNet and GoogleNet were also used to classify chest X-ray images. The dataset was split into training, testing and validation sets as 68%, 14.9% and 17.1%, respectively. Data augmentation and pre-processing techniques were employed to get the best performing model achieving an AUC of 0.99. Precision and recall of the model were 100 and 97.3%. An AG-CNN model was developed by Guan et al. [8] to recognize thorax disease. ChestX-ray14 dataset was used to detect thorax disease from chest X-ray images. Global and local branch attention-guided CNN was used for classification purposes. Their model was better than other models mentioned in their research paper, achieving an AUC of 0.868. A deep convolutional neural network model was developed by Rajpurkar et al. [9] to classify chest X-ray images into pneumonia and other 14 diseases. ChestX-ray14 dataset was used for training the model. They compared their ChXNet model (121 layered model) with practicing academic radiologists. Their ChXNet model achieved an F1 score (95% CI) of 0.435 outperforming radiologists which achieved an F1 score (95% CI) of 0.387. A deep convolutional neural network model having five convolutional layers some followed by max-pooling layers, having three fully connected layers was trained by Krizhevsky et al. [10]. This network contained 60 million different parameters. By employing dropout, this model achieved a top-five error percent of 17%. Simonyan et al. [11] developed a highly accurate model employing multiple small kernel-sized filters to achieve top-five test accuracy 92.7%. This model was trained on the ImageNet dataset and submitted to the ILSVRC 2014 competition. A convolution neural network for classification and segmentation of brain tumor MRIs was developed by Xu et al. [12]. Multiple techniques such as data augmentation, feature selection and pooling techniques were employed in this model. The validation 474 V. Sirish Kaushik et al. accuracy for classification achieved by this model is 97.5%, and validation accuracy of segmentation is 84%, 256 × 256 pixels sized frontal chest radiographs which were fed to a deep convolution neural network to detect abnormalities. A convolutional neural network with five convolution layers employing leaky ReLU, average pooling and three fully connected layers was developed by Anthimopoulos et al. [13] to detect interstitial lung disease patterns in a dataset containing 14,696 images belonging to seven different classes. This model achieved a classification accuracy of 85.5%. He et al. [14] developed a residual neural network (RNN) to classify images present in the ImageNet dataset. RNN introduced the concept of shortcut connections to tackle the problem of vanishing gradients. This model which was submitted to ILSVRC 2015 attained state-of-the-art classification accuracy. A transfer learning model, extension of AlexNet using data augmentation techniques, was developed by Glozman et al. [15]. This model was trained on ADNI database. Two neural network models were presented by Hemanth et al. [16] which are MCPN and MKNN. These models classified MRIs with high accuracies and tackled high convergence time period for Artificial Neural Networks.

Dept. of CSE, KIT, Tiptur Page 14

Detection Of Pneumonia Using Convolutional Neural Networks



**Chapter 3**

**SYSTEM ARCHITECTURE**

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing.

**3.1 Introduction to System Architecture**

System Architecture design-identifies the overall hypermedia structure for the Application. Architecture design is tied to the goals establish for a Application, the content to be presented, the users who will visit, and the navigation philosophy that has been established. Content architecture, focuses on the manner in which content objects and structured for presentation and navigation. Application architecture, addresses the manner in which the application is structure to manage user interaction, handle internal processing tasks, effect navigation, and present content. Application architecture is defined within the context of the development environment in which the application is to be implemented.

**3.2 System Requirements**

Software Requirement Specification (SRS) is a fundamental document, which forms the foundation of the software development process. SRS not only lists the requirements of a system but also has a description of its major features. These recommendations extend the IEEE standards. The recommendations would form the basis for providing clear visibility of the product to be developed serving as baseline for execution of a contract between client and the developer. SRS constitutes the agreement between clients and developers regarding the contents of the software product that is going to be developed. SRS should accurately and completely represent the system requirements as it makes a huge contribution to the overall project plan. The software being developed may be a part of the overall larger system or may be a complete standalone system in its own right.



Dept. of CSE, KIT, Tiptur Page 15

Detection Of Pneumonia Using Convolutional Neural Networks



**3.2.1 Hardware Requirements**

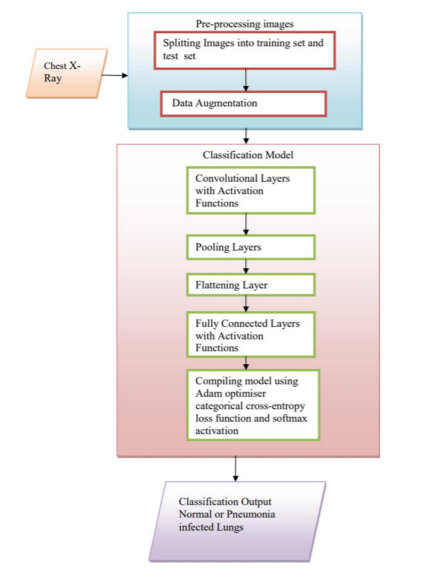
|  |  |  |
| --- | --- | --- |
| • | Processor | : Intel Core 2 Duo |
| • | RAM | :2GB |
| • | Hard Disk | :80GB |

**3.2.2 Software Requirements**

|  |  |  |
| --- | --- | --- |
| • | Operating System | : Windows (Any Version) |
| • | Programming Language | : Python |
| • | IDE | : Pycharm IDE |

**3.3 System Architecture**

System Architecture design-identifies the overall structure for the Liver disease as shown in below diagram.



**Figure 3.1: System Architecture**

Figure 3.1 System Architecture for pneumonia disease ,Compares Naïve Bayes and SVM algorithms are done and it is based on the performance factors classification accuracy and execution time.

Dept. of CSE, KIT, Tiptur Page 21

**Chapter 4**

**METHODOLOGY**

CNN models have been created from scratch and trained on Chest X-Ray Images (Pneumonia) dataset on Kaggle. Keras neural network library with TensorFlow backend has been used to implement the models. Dataset consists of 5216 training images, 624 testing images and 16 validation images. Data augmentation has been applied to achieve better results from the dataset. The four models have been trained on the training dataset, each with different number of convolutional layers. Each model was trained for 20 epochs, with training and testing batch sizes of 32 and 1, respectively. The following sub-headings further explain the above stages in depth.

**4.1 CNN Architecture**

CNN models are feed-forward networks with convolutional layers, pooling layers, flattening layers and fully connected layers employing suitable activation functions.

**4.1.1 Convolutional layer.**

It is the building block of the CNNs. Convolution operation is done in mathematics to merge two functions [17]. In the CNN models, the input image is first converted into matrix form. Convolution filter is applied to the input matrix which slides over it, performing element-wise multiplication and storing the sum. This creates a feature map. 3 × 3 filter is generally employed to create 2D feature maps when images are black and white. Convolutions are performed in 3D when the input image is represented as a 3D matrix where the RGB color represents the third dimension. Several feature detectors are operated with the input matrix to generate a layer of feature maps which thus forms the convolutional layer.

**4.1.2 Activation functions.**

All four models presented in this paper use two different activation functions, namely ReLU activation function and softmax activation function. The ReLU activation function stands for rectified linear function [18]. It is a nonlinear function that outputs zero when the input is negative and outputs one when the input is positive. The ReLU function is given by the following formula: This type of activation function is broadly used in CNNs as it deals with the problem of vanishing gradients and is useful for increasing the nonlinearity of layers. ReLU activation function has many variants such as Noisy ReLUs, Leaky ReLUs and Parametric ReLUs. Advantages of ReLU over other activation functions are computational simplicity and representational sparsity. Softmax activation function is used in all four models presented in this paper. This broadly used activation function is employed in the last dense layer of all the four models [19]. This activation function normalizes inputs into a probability distribution. Categorical cross-entropy cost function is mostly used with this type of activation function.

**4.1.3 Pooling layer.**

Convolutional layers are followed by pooling layers. The type of pooling layer used in all four models is max-pooling layers. The max-pooling layer having a dimension of 2 × 2 selects the maximum pixel intensity values from the window of the image currently covered by the kernel. Max-pooling is used to down sample images, hence reducing the dimensionality and complexity of the image [20]. Two other types of pooling layers can also be used which are general pooling and overlapping pooling. The models presented in this paper use max-pooling technique as it helps recognize salient features in the image.

**4.1.4 Flattening layer and fully connected layers**.

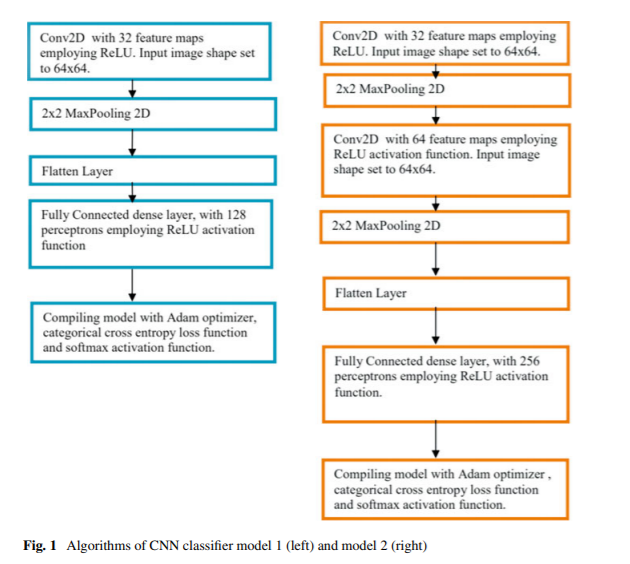
After the input image passes through the convolutional layer and the pooling layer, it is fed into the flattening layer. This layer flattens out the input image into a column, further reducing its computational complexity. This is then fed into the fully connected layer/dense layer. The fully connected layer [21] has multiple layers, and every node in the first layer is connected to every node in the second layer. Each layer in the fully connected layer extracts features, and on this basis, the network makes a prediction [22, 23]. This process is known as forward propagation. After forward propagation, a cost function is calculated. It is a measure of performance of a neural network model. The cost function used in all four models is categorical cross-entropy. After the cost function is calculated, back propagation takes place. This process is repeated until the network achieves optimum performance. Adam optimization algorithm has been used in all four models.

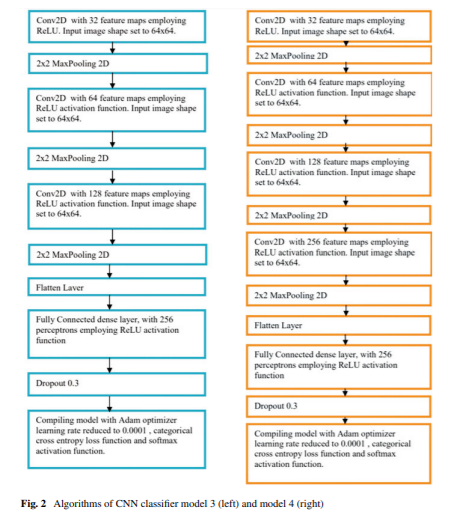
**4.1.5 Reducing overfitting.**

The first model exhibits substantial overfitting; hence, dropout technique was employed in the later models [24]. Dropout technique helps to reduce overfitting and tackles the problem of vanishing gradients. Dropout technique encourages each neuron to form its own individual representation of the input data. This technique on a random basis cuts connections between neurons in successive layers during the training process [25]. Learning rate of models was also modified, to reduce overfitting. Data augmentation technique can also be employed to reduce overfitting.

**4.1.6 Algorithm of CNN classifiers.**

The algorithms used in the convolutional neural network classifiers have been explained in Figs. 1 and 2. Figure 3 shows the flowchart of the overall schema of research. The number of epochs for all the classifier models presented in this paper was fixed at 20 after training and testing several CNN models over the course of research. Classifier models trained for more number of epochs have showed overfitting. Several optimizer functions were also trained and studied. Adam optimizer function was finalized to be used for all classifiers after it gave the best results. Initially, a simple classifier model with convolutional layer of image size set to 64 \* 64, 32 feature maps and employing ReLU activation function was trained. Fully connected dense layer with 128 perceptrons was utilized. To improve the result, the second classifier model was trained with one more convolutional layer of 64 feature maps for better feature extraction. The number of perceptrons in dense layer was also doubled to 256, so that better learning could be achieved. The third model was trained for three convolutional layers with 128 feature maps in third convolutional layer for more detailed feature extraction. Dense layer was kept unchanged. Dropout layer was introduced at 0.3, and learning rate of optimizer was lowered to 0.0001 to reduce the overfitting. The final fourth classifier model was trained for four convolutional layers with 256 feature maps in fourth convolutional layer. Dense layer, dropout layer and learning rate were kept same as third classifier model. The results have been summarized in the subsequent section of this paper. Dataset. Chest X-Ray Images (Pneumonia) dataset of 1.16 GB size has been imported from Kaggle [26], with total of 5856 jpeg images split into Train, Test and Val folders each divided into category Pneumonia and Normal. Chest X-ray images (front and back) were selected from pediatric patients of one- to five-year olds from Guangzhou Women and Children’s Medical Center, Guangzhou.





 Detection Of Pneumonia Using Convolutional Neural Networks



**REFERENCES**

1. https://data.unicef.org/topic/child-health/pneumonia/. Accessed on 15 July 2019

2. Jaiswal, A.K., Tiwari, P., Kumar, S., Gupta, D., Khanna, A., Rodrigues, J.J.: Identifying pneumonia in chest x-rays: a deep learning approach. Measurement 145, 511–518 (2019)

3. Kim, D.H., MacKinnon, T.: Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks. Clin. Radiol. 73(5), 439–445 (2018)

4. Bernal, J., Kushibar, K., Asfaw, D.S., Valverde, S., Oliver, A., Martí, R., Lladó, X.: Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review. Artif. Intell. Med. 95, 64–81 (2019)

5. Arthur, F., Hossein, K.R.: Deep learning in medical image analysis: a third eye for doctors. J. Stomatology Oral Maxillofac. Surg.

6. Rubin, J., Sanghavi, D., Zhao, C., Lee, K., Qadir, A., Xu-Wilson, M.: Large Scale Automated Reading of Frontal and Lateral Chest X-Rays Using Dual Convolutional Neural Networks (2018). arXiv preprint arXiv:1804.07839

7. Lakhani, P., Sundaram, B.: Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology 284(2), 574–582 (2017)

8. Guan, Q., Huang, Y., Zhong, Z., Zheng, Z., Zheng, L., Yang, Y.: Diagnose Like a Radiologist: Attention Guided Convolutional Neural Network for Thorax Disease Classification (2018). arXiv preprint arXiv:1801.09927

9. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M.P.: Chexnet: Radiologist-Level Pneumonia Detection on Chest X-rays with Deep Learning (2017). arXiv preprint arXiv:1711.05225

10. Krizhevsky, A., Sutskever, I., Hinton, G.E:. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)

11. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition (2014). arXiv preprint arXiv:1409.1556

12. Xu, Y., Jia, Z., Ai, Y., Zhang, F., Lai, M., Eric, I., Chang, C.: Deep convolutional activation features for large scale brain tumor histopathology image classification and segmentation. In: 2015 international conference on acoustics, speech and signal processing (ICASSP), pp. 947– 951 (2015)

13. Anthimopoulos, M., Christodoulidis, S., Ebner, L., Christe, A., Mougiakakou, S.: Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. IEEE Trans. Med. Imaging 35(5), 1207–1216 (2016)

14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

15. Glozman, T., Liba, O.: Hidden Cues: Deep Learning for Alzheimer’s Disease Classification CS331B project final report (2016)

16. Hemanth, D.J., Vijila, C.K.S., Selvakumar, A.I., Anitha, J.: Performance improved iterationfree artificial neural networks for abnormal magnetic resonance brain image classification. Neurocomputing 130, 98–107 (2014)

17. Bi, X., Li, S., Xiao, B., Li, Y., Wang, G., Ma, X.: Computer aided Alzheimer’s disease diagnosis by an unsupervised deep learning technology. Neurocomputing (2019)

18. Eckle, K., Schmidt-Hieber, J.: A comparison of deep networks with ReLU activation function and linear spline-type methods. Neural Netw. 110, 232–242 (2019)

19. Ren, S., Jain, D.K., Guo, K., Xu, T., Chi, T.: Towards efficient medical lesion image superresolution based on deep residual networks. Sig. Process. Image Commun. 75, 1–10 (2019)

20. Zheng, Y., Iwana, B.K., Uchida, S.: Mining the displacement of max-pooling for text recognition. Pattern Recogn. 93, 558–569 (2019)

21. Bhumika, P.S.S.S., Nayyar, P.A.: A review paper on algorithms used for text classification. Int. J. Appl. Innov. Eng. Manage. 3(2), 90–99 (2013)

22. Kumar, A., Sangwan, S.R., Arora, A., Nayyar, A., Abdel-Basset, M.: Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. IEEE Access 7, 23319–23328 (2019)

23. Saeed, F., Paul, A., Karthigaikumar, P., Nayyar, A.: Convolutional neural network based early fire detection. In: Multimedia Tools and Applications, pp. 1–17 (2019)

24. Kukkar, A., Mohana, R., Nayyar, A., Kim, J., Kang, B.G., Chilamkurti, N.: A novel deeplearning-based bug severity classification technique using convolutional neural networks and random forest with boosting. Sensors 19(13), 2964 (2019)

25. Khan, S.H., Hayat,M., Porikli, F.: Regularization of deep neural networks with spectral dropout. Neural Netw. 110, 82–90 (2019)

26. https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia. Accessed on 15 July 2019

27. ALzubi, J.A., Bharathikannan, B., Tanwar, S., Manikandan, R., Khanna, A., Thaventhiran, C.: Boosted neural network ensemble classification for lung cancer disease diagnosis. Appl. Soft Comput. 80, 579–591 (2019)

28. Vora, J., Tanwar, S., Polkowski, Z., Tyagi, S., Singh, P.K., Singh, Y.: Machine learning-based software effort estimation: an analysis. In: 11th International Conference on Electronics, computers and Artificial Intelligence (ECAI 2019), pp. 1–6, University of Pitesti, Pitesti, Romania, 27–29 June 2019

29. Chakraborty, S., Aich, S., Sim, J.S., Kim, H.C.: 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)

30. Liang, G., Zheng, L.: A transfer learning method with deep residual network for pediatric pneumonia diagnosis. In: Computer Methods and Programs in Biomedicine (2019)

31. Du, S. S., Zhai, X., Poczos, B., Singh, A.: Gradient Descent Provably Optimizes OverParameterized Neural Networks (2018). arXiv preprint arXiv:1810.02054

32. Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (2015). arXiv preprint arXiv:1511.06434.