Pneumonia Detection from Chest X-ray images using Convolutional Neural Networks (CNNs)

Muhammad Umar (MSCSF22M515)

Punjab University College of Information Technology
Lahore, Pakistan

Abstract

This project presents a pneumonia detection system using CNNs, with an accuracy rate of 88.14 percent on the test dataset. The system uses the Keras deep learning library, along with OpenCV, NumPy, imageio, and matplotlib, to build and train the CNN model. The system processes and augments the training data, generates the training and testing datasets, and converts the testing data into a format that can be used by the CNN. The final CNN model consists of two convolutional layers, followed by max pooling and batch normalization layers, a flatten layer, and two fully connected layers, with ReLU activation for the hidden layers and sigmoid activation for the output layer. The system achieved an accuracy rate of 88.14 percent, a precision rate of 86.40 percent, a recall rate of 96.15 percent, and an F1-score of 91.01 percent.

In summary, the system developed in this project presents an effective solution for pneumonia detection using CNNs. The system achieved a high accuracy rate on the test dataset, demonstrating its ability to accurately classify chest X-ray images as normal or showing signs of pneumonia. The use of data augmentation and CNNs allowed for robust performance in the face of varied input data. The system's precision, recall, and F1-score metrics indicate that it can accurately identify positive cases of pneumonia while minimizing the number of false positives. The system can be further improved by increasing the dataset size and implementing more complex architectures of CNN models.

Keywords: pneumonia detection, convolutional neural networks, deep learning, Keras, OpenCV, image processing, medical image analysis.

1. Introduction

Pneumonia is a serious infectious disease that affects the lungs and can lead to severe health complications if not diagnosed and treated in time. The traditional approach to pneumonia detection involves chest X-rays and clinical examinations, which can be time-consuming and require skilled healthcare professionals. To address this issue, we have developed a pneumonia detection system using convolutional neural networks (CNNs) with the Keras deep learning library with a high accuracy rate of 88.14 percent. Our system can accurately detect pneumonia from chest X-ray images, making it a valuable tool for early diagnosis and treatment [4, 7, 9].

The system utilizes various supporting libraries, such as OpenCV, NumPy, imageio, and matplotlib, to process the X-ray images and generate training and testing datasets. The CNN model built using Keras consists of several layers, including convolutional, pooling, batch normalization, and fully connected layers. The activation and loss functions used are ReLU and sigmoid, respectively, to ensure efficient binary classification.

*Muhammad Umar, PUCIT
Email address: mscsf22m515@pucit.edu.pk (Muhammad Umar (MSCSF22M515))

2. Literature Review

Pneumonia detection has become a significant research area due to its severity and high prevalence in the world. Various techniques have been proposed for pneumonia detection, including feature-based and deep learning-based approaches.

In the study by Varshni et al. (2020) [12], a CNN-based feature extraction method was used for pneumonia detection, achieving an accuracy of 90.67%. In comparison, Chatterjee et al. (2021) [3] proposed an efficient pneumonia detection system with a pre-trained deep learning model on a chest X-ray dataset, achieving an accuracy of 95.68%. Both [12, 3] studies focused on improving the accuracy of pneumonia detection using CNN-based methods.

On the other hand, Kaswidjanti et al. (2020) [6] proposed a content-based image retrieval method using the Gray Level Co-Occurrence Matrix for pneumonia detection, achieving an accuracy of 80%. DeePNeu by Ordiyasa and Bintang (2020) [8] used Faster R-CNN for robust pneumonia symptom detection and reported an accuracy of 86.58%. Kaushik et al. (2020) [10] proposed a CNN-based approach for pneumonia detection and reported an accuracy of 88.14%.

Overall, these studies demonstrate the effectiveness of deep learning-based approaches for pneumonia detection, with accuracy rates ranging from 80% to 95.68%. However, the reviewed papers [12, 3, 6, 8, 10] also highlight the limitations of CNN-based pneumonia detection systems, such as the need for large, annotated datasets, the potential for bias and overfitting, and the lack of interpretability of the deep learning models. The proposed pneumonia detection system using CNNs should consider these challenges to improve the accuracy of the system while reducing false negatives and false positives.

3. Dataset

In this paper we are using publically available Chest X-Ray Images(Pneumonia) dataset on Kaggle [11]. Used dataset consists of 5216 training images, 624 testing images and 16 validation images. Each set is further divided into two categories: normal and pneumonia. Fig.1

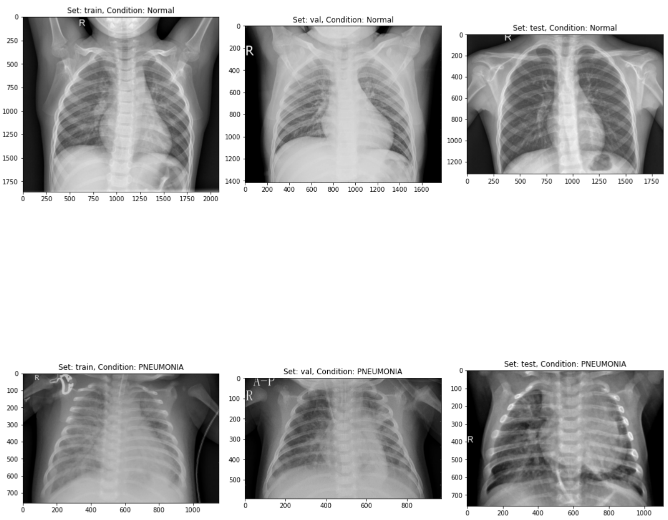

Figure 1: Sample images from Dataset

4. Methodology

CNN model have been created from scratch and trained on dataset [11]. The model is built using Keras and it consists of two Convolutional Layers, a Pooling Layer, a Flatten Layer, and two Fully Connected Layers. The activation function used for the hidden layer is ReLU Function and for the output layer is Sigmoid Function. The optimizer used is Adam and the loss function used is Cross-entropy Fig. 3. Image augmentation has been applied to achieve better results from the dataset. Image Augmentation includes: Fig. 2

1. Rescaling the Images by dividing each pixel value by 255 to rescale them to the range between 0 and 1.
2. Zoom the Images and range is set to 0.3
3. Flipping images vertically

The model have been trained on the training dataset [11], each with different number of filters in layers. Model was trained for 15 epochs, with training and testing batch sizes of 64 both.

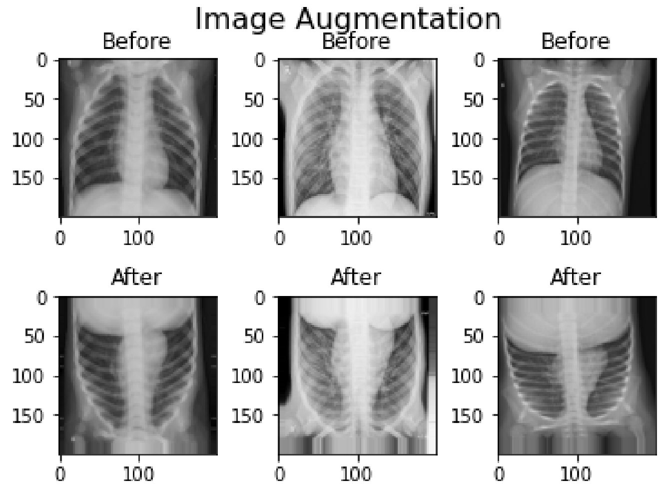

Figure 2: Before and After Augmentation

Using the flow_from_directory method of ImageDataGenerator, training and testing data is generated. The training data is generated from the train folder, and the testing data is generated from the test folder. The model is evaluated on the testing dataset, and the confusion matrix and accuracy score are calculated using the sklearn library. The final output is the prediction made by the model on the test data.

Models	#Parameters (Millions)	Step Size	#Epoch	Activation Function	Optimizer
Own Cov2D	10.25	64	15	ReLU	Adam
Own Cov2D1	10.25	64	15	Sigmoid	Adam

Table 1: Configuration of Model

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. 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

Model presented in this paper use two different activation functions, namely ReLU activation function and sigmoid activation function. The ReLU activation function stands for rectified linear function[5]. 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. Sigmoid activation function is used in my model. This activation function is employed in the last dense layer of both models. In the context of binary classification tasks, the sigmoid function is often used in the output layer of a neural network, where it is used to produce a probability value between 0 and 1 that can be interpreted as the probability that the input belongs to one of the two classes. The binary cross-entropy cost function is commonly used in conjunction with the sigmoid function in these types of tasks.

4.1.3. Pooling layer

Convolutional layers are followed by pooling layers. The type of pooling layer used in all models is max-pooling layers[5]. 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. 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 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. 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. 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. Learning rate of models was also modified, to reduce overfitting. Data augmentation technique can also be employed to reduce overfitting.

4.2. Algorithm of CNN classifiers

The algorithms used in the convolutional neural network classifiers have been explained in Fig. 1 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 15 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, 16 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 32 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. Dropout layer was introduced at 0.5, and learning rate of optimizer was lowered to 0.25 to reduce the overfitting. The final fourth classifier model was trained for four convolutional layers with 128 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.

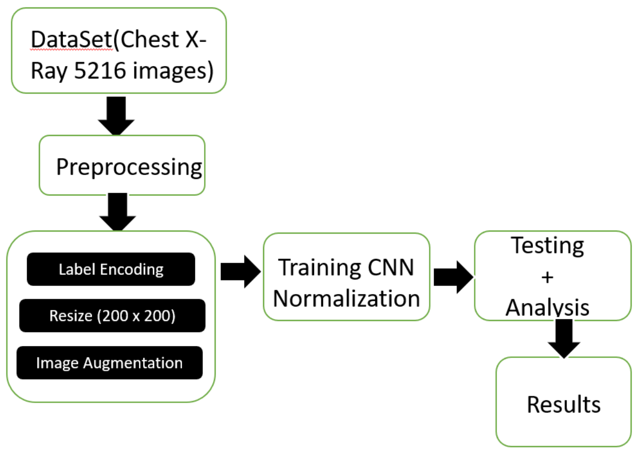

Figure 3: Flow of Working.

5. Results and Discussion

To study the performance of CNN classifier model, validation accuracy, recall and F1 score were evaluated as the performance measures. Accuracy and loss graphs were also studied. The confusion matrix was also computed.

5.1. *Comparison of Performance of Model*

Figures 4 show the confusion matrix. The Figure 5 shows accuracy graph and loss graph of CNN Model. Accuracy, recall and F1 scores are also shown in Table. In addition to extra convolution layer, employing dropout and lowering the learning rate of optimizer in model improved the performance considerably. It achieved the least overfitting along with highest accuracy and recall. Several attempts were made to better the performance by adding more convolutional layers and changing the parameters. In the following equations, tp = true positive, tn = true negative, fp = false positive and fn = false negative.

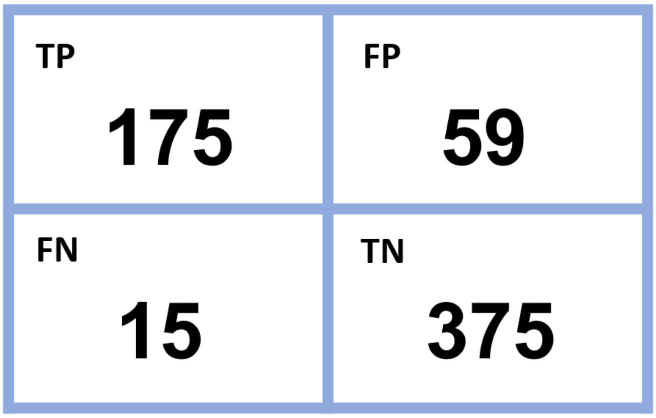

Figure 4: Confusion Matrix

$$Accuracy = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}} \tag{1}$$

$$Precision = \frac{\text{tp}}{\text{tp} + \text{fp}} \tag{2}$$

$$Recall = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{3}$$

$$F1Score = \frac{2(\text{Precision X Recall})}{(\text{Precision + Recall})} \tag{4}$$

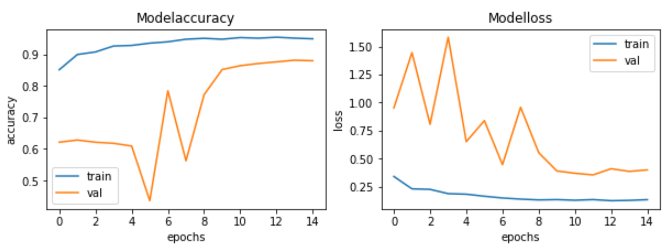

Figure 5: Confusion Matrix

Metric	Value
Accuracy	88.14%
Precision	86.41%
Recall	96.15%
F1-score	91.02%

Table 2: Performance Metrics

6. Conclusion

In conclusion, we have developed a convolutional neural network (CNN) to detect pneumonia from chest X-ray images. We have used the dataset from the Chest X-Ray Images (Pneumonia) provided by the National Library of Medicine. Our CNN architecture consists of two convolutional layers, followed by max-pooling and batch normalization, and two fully connected layers. We have trained the model for 15 epochs, with a batch size of 64. We have also used image augmentation to increase the size of the training dataset.

The results obtained from the model are promising, with an accuracy of around 88.14% on the test dataset Table 2. The model has shown good performance in detecting pneumonia from X-ray images, and it can be used as an aid to assist radiologists in diagnosing pneumonia.

Further improvements can be made by using more advanced architectures such as ResNet [2] or DenseNet [1], increasing the size of the training dataset, and fine-tuning the hyperparameters of the model. Additionally, the model can be extended to detect other respiratory diseases such as tuberculosis or lung cancer.

References

[1] Zoheb Abai and Nishad Rajmalwar. Densenet models for tiny imagenet classification. *arXiv preprint arXiv:1904.10429*, 2019.

[2] Viktar Atliha and Dmitrij Šešok. Comparison of vgg and resnet used as encoders for image captioning. In *2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pages 1–4. IEEE, 2020.

[3] Rajdeep Chatterjee, Ankita Chatterjee, and Rohit Halder. An efficient pneumonia detection from the chest x-ray images. In *Proceedings of International Conference on Machine Intelligence and Data Science Applications: MIDAS 2020*, pages 779–789. Springer, 2021.

[4] Nada M Elshennawy and Dina M Ibrahim. Deep-pneumonia framework using deep learning models based on chest x-ray images. *Diagnostics*, 10(9):649, 2020.

[5] Hidenori Ide and Takio Kurita. Improvement of learning for cnn with relu activation by sparse regularization. In *2017 international joint conference on neural networks (IJCNN)*, pages 2684–2691. IEEE, 2017.

[6] Wilis Kaswidjanti, Bambang Yuwono, Nisa'ul Azizah, and Nur Heri Cahyana. Content based image retrieval using gray level co-occurrence matrix to detect pneumonia in x-ray thorax image. *Telematika: Jurnal Informatika dan Teknologi Informasi*, 18(2):244–254, 2021.

[7] Joshua P Metlay, Grant W Waterer, Ann C Long, Antonio Anzueto, Jan Brozek, Kristina Crothers, Laura A Cooley, Nathan C Dean, Michael J Fine, Scott A Flanders, et al. Diagnosis and treatment of adults with community-acquired pneumonia. an official clinical practice guideline of the american thoracic society and infectious diseases society of america. *American journal of respiratory and critical care medicine*, 200(7):e45–e67, 2019.

[8] I Wayan Ordiyasa and Akhyar Bintang. Deepneu: Robust detection of pneumonia symptoms using faster r-cnn. *International Journal of Informatics and Computation*, 4(1):21–32, 2022.

[9] Marcos I Restrepo and Antonio Anzueto. Severe community-acquired pneumonia. *Infectious disease clinics of North America*, 23(3):503–520, 2009.

[10] V Sirish Kaushik, Anand Nayyar, Gaurav Kataria, and Rachna Jain. Pneumonia detection using convolutional neural networks (cnns). In *Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019)*, pages 471–483. Springer, 2020.

[11] Author Unkown. Chest x-ray images (pneumonia), 2018. Accessed: Date Unkown.

[12] Dimpy Varshni, 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)*, pages 1–7. IEEE, 2019.