Pneumonia detection using Convolutional Neural Network

Mohit Gupta
mgupta6@ncsu.edu

Rajat Dange
rdange@ncsu.edu

Rutvik Kolhe
rkolhe@ncsu.edu

1 Introduction

Pneumonia is a form of an acute respiratory infection that affects the lungs. The risk of pneumonia is enormous for many, especially for children. According to the World Health Organisation, Pneumonia accounts for 15 percent of all deaths of children under five years old, killing 808,694 children in 2017. The problem is even worse for developing and under-developed countries. The proposed method would help in the early detection of the disease, and consequently, it may benefit many.

Convolutional Neural Network(CNN) is used in multiple computer vision tasks, such as image classification, object detection, etc. While many efforts have been given to general tasks, now the focus has started shifted to medical images. Further, with significant data growth in healthcare communities, medical data could be analysed for early disease detection.

The project proposes and builds a Convolutional Neural Network to detect pneumonia (bacterial or viral) using chest X-ray images database. The project develops a seven-layer CNN especially designed and trained from scratch to classify bacterial and viral pneumonia from X-Ray images correctly. Our project(model) is showing encouraging results(accuracy, precision, recall), and the experimental results show the potential for medical image analysis for pneumonia detection. Further, we will also explore techniques such as dropout and data augmentation to boost accuracy furthermore.

2 Background

Convolutional neural networks(CNNs) are a category of neural networks which perform well on image data. CNNs have been very successful in identifying and classifying faces and objects and are used to power vision in robots and self-driving cars. The architecture of a CNN 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.

The main components of a CNN are - Convolutional layers, Pooling layers, and Fully Connected layer[1]. The input consists of raw pixel values of the image, with three color channels R,G,B. A convolutional layer computes the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. We can have as many filters as we want. The output of convolution layer is a matrix of dimensions W x H x N where W is the width of image, H is the height of image and N is the number of filters. After the convolution operation we perform activation on the output. The most common activation function used in CNNs is ReLU. The pooling layer performs downsampling operations which reduce the input to lower dimensions. Finally, the fully connected layer is same as a regular neural network where every neuron is connected to every neuron in the next layer.

3 Method

3.1 Data Pre-processing

Before feeding the data to our CNN we need to resize images to reduce our training time. As we know that images are stored in the form of pixel in matrices, hence big image sizes means large matrix multiplications. Thus more computational power and training time would be required when we feed these images to CNN. Therefore, we have reduced the size of the image to 200*200 pixels. Since, we are dealing with X-rays, we have processed the images in grayscale. We labeled the each image into three categories- Normal, Bacteria-Pneumonia and Virus-Pneumonia. We then, split the data into training and testing data arrays which were later converted to numpy arrays. These numpy arrays were then fed into the neural network. While training the model, we used 10% of the training data as validation data to improve the accuracy of the model.

3.2 Model Architecture

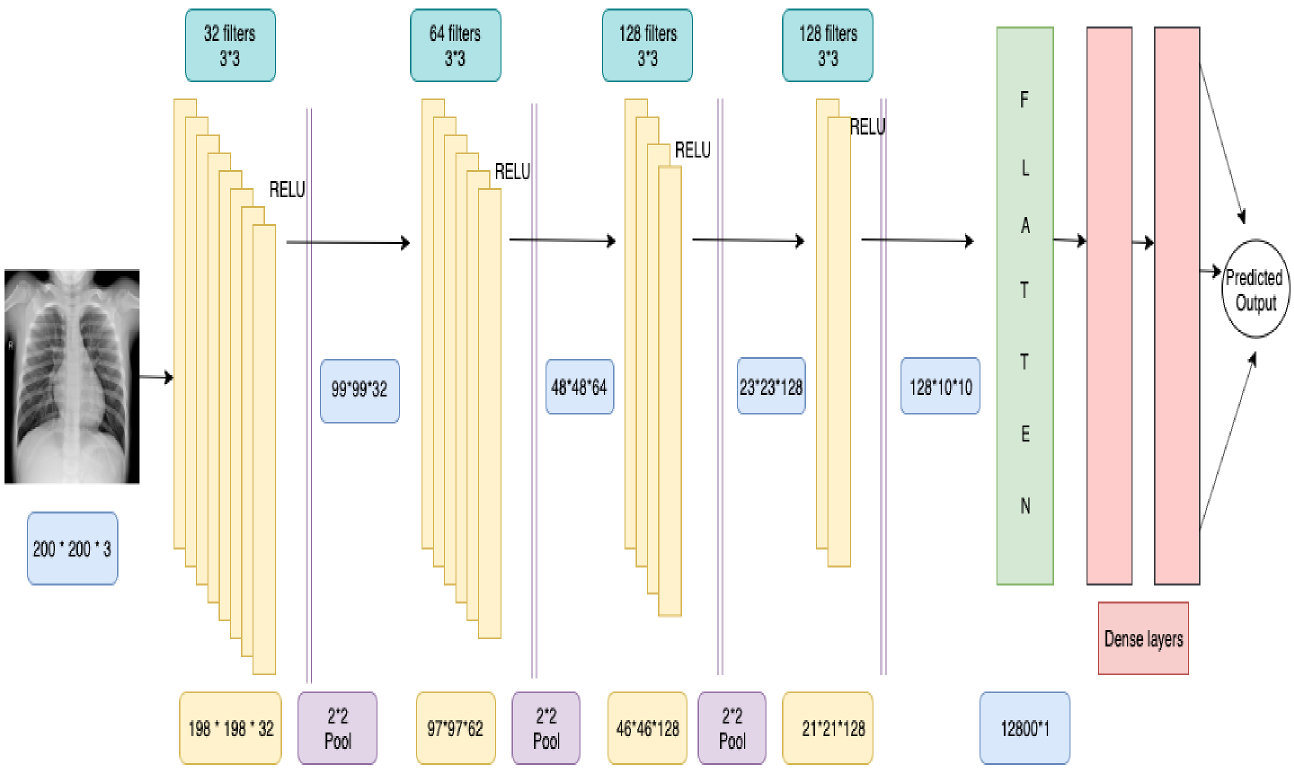

Figure 1: Figure: Architecture

The above figure shows our proposed CNN architecture. Our model consists of four feature extractors[2][3], each of which consist of one 2-D convolutional layer, a max-pooling layer and a use ReLU activator, a flattening layer and finally two dense layers which act as classifier. Each of these layers take the preceding layer's output as it's input except for the first layer. The feature extractors comprise of a 32 3x3 convolutional layer, 64 3x3 convolutional layer, 128 3x3 convolutional layer, and 128 3x3 convolutional layer respectively[4] with a 2x2 max-pooling layers between them. The output of these convolution operations are assembled into feature maps of size 198x198x32, 97x97x62, 46x46x128, 21x21x128 respectively and the output of pooling operations are assembled into feature maps of size 99x99x32, 48x48x64, 23x23x128, and 10x10x128 respectively[5]. Our output labels consist of three classes represented in the form of integer values 0,1,2. Therefore,we used sparse_categorical_crossentropy and sparse_categorical_accuracy as hyperparameters for loss and metric for the model.

4 Experiments

4.1 Dropout

Since we are building this model from scratch there's a good chance we might overfit the model. Hence we will be using dropout technique in the dense layer to better generalize the model. Dropout is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data.

4.2 Data Augmentation

Although we have a decent amount of training data available we can make use of data augmentation techniques to bring even more diversity in our dataset and train the model on the augmented data to achieve better results. Some of the techniques we plan on using are - affine transformations, contrast changes, gaussian noise etc.

4.3 Different Image sizes

We are currently reducing the original image size to 200x200 pixels. We can also experiment by varying the image sizes. Lower image size corresponds to lower training time, hence the lower the image size the more we can experiment by tuning other hyperparameters as our training time will be low.

5 Results

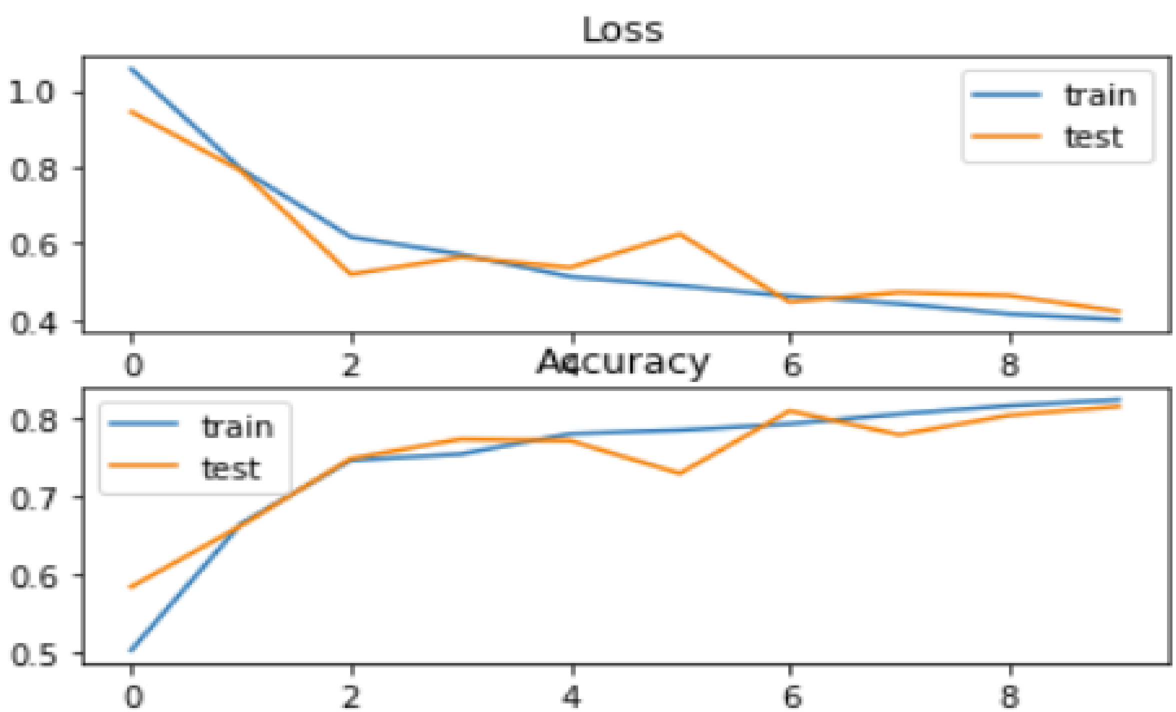

Figure 2: Results

The above figure shows accuracy and loss over 10 epochs. X- axis represents the number of epochs and y-axis represents Loss and Accuracy respectively.

The other results are as follows:
Accuracy: 0.775641

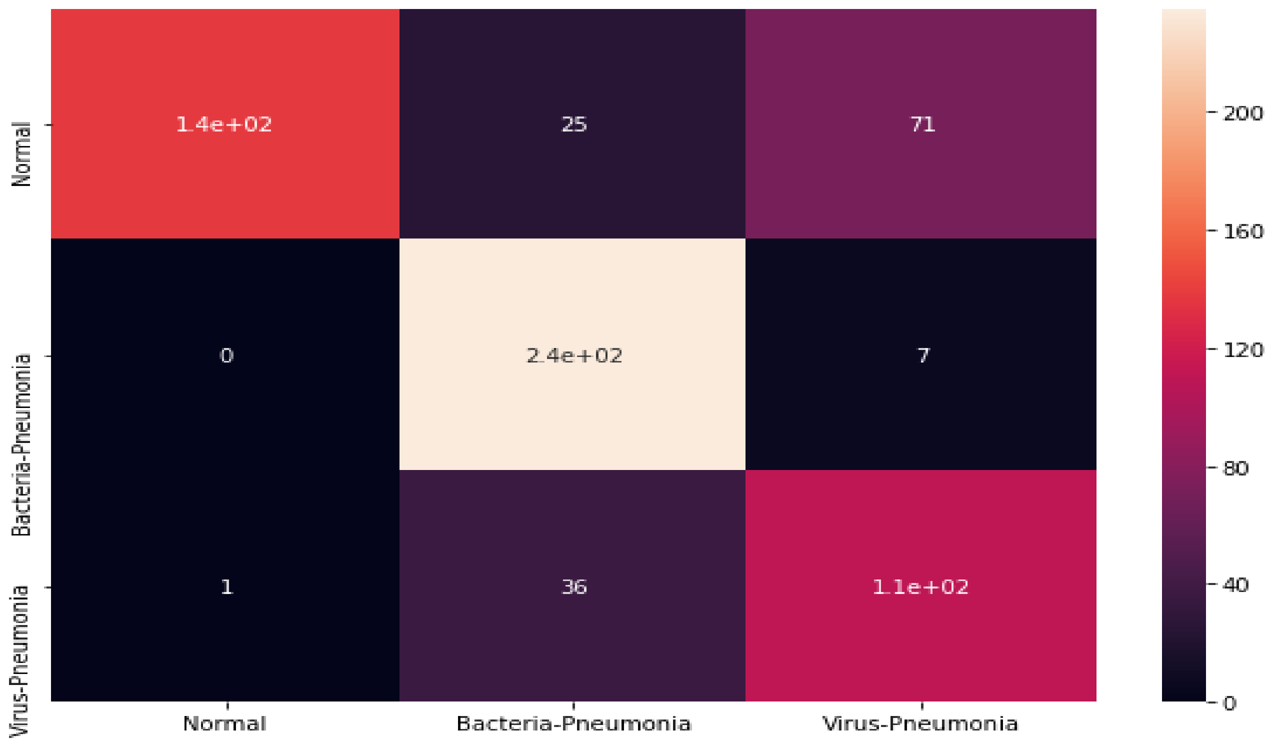

Figure 3: Confusion Matrix

Precision: 0.791342
Recall: 0.770273
F1 score: 0.757435

5.1 Dropout

We are working on it and results will be added in final report.

5.2 Data Augmentation

We are working on it and results will be added in final report.

5.3 Different Image sizes

We are working on it and results will be added in final report.

6 Conclusion

We have built a basic Convolutional Neural Network to classify pneumonia with an accuracy of 0.77. We are getting very decent results, and we will be performing some more experiments to improve further our results.

7 References

[1]: Olaf R., Philipp F., Thomas B. U-Net: Convolutional Networks for Biomedical Image Segmentation. New York, NY, USA: MICCAI Springer; 2015.

[2]:Varun G., Lily P., Marc C., et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA. 2017;316(22):2402–2410. doi: 10.1001/jama.2016.17216.

[3]:Huang P., Park S., Yan R., et al. Added value of computer-aided CT image features for early lung cancer diagnosis with small pulmonary nodules: a matched case-control study. Radiology. 2017;286(1):286–295. doi: 10.1148/radiol.2017162725

[4]: Mohammad T. I., Md A. A., Ahmed T. M., Khalid A. Abnormality detection and localization in chest x-rays using deep convolutional neural networks. 2017.

[5]:Yao Li., Poblenz E., Dagunts D., Covington B., Bernard D., Lyman K. Learning to diagnose from scratch by exploiting dependencies among labels. 2017.