Chest X-Ray Pneumonia Detection Based on Convolution Neural Networks

ECE 740 LEC B01 – Design Document

15th April,2021

Submitted by:

Shyam Prasad Reddy Bachu – 1654290

Table of Contents

Abstract

Pneumonia is one of the most severe and life-threatening diseases caused by a bacterial or viral lung infection that can lead to severe effects within a brief period of time. Therefore, in terms of the effective recovery process, early diagnosis is a major factor. In one or both of the lungs, Pneumonia is an infection which causes inflammation and fluid build-up [1]. It can cause oxygen exchange problems. WHO estimated that Pneumonia causes around 700,000 children to die each year and affects 7% of the world's population [2]. There is a need for a system that can detect pneumonia using chest x-rays that can allow specialists and radiologists to recognize it effectively and rapidly. In this project, I will create a model that will help to distinguish chest x-rays and forecast pneumonia. Here I am using Convolution Neural Networks which is based on the latest state-of-the-art models used for the highest precision and performance. Also implemented VGG_16 using transfer learning.

1. Introduction

Pneumonia is a condition that affects the human respiratory system. The disease affects the lungs, specifically the alveoli, which are small air sacs that fill up while breathing. Viruses, bacteria, and fungi cause pneumonia. Alveoli are filled with pus and fluid in pneumonia patients, making breathing difficult and limiting oxygen consumption [1].

Pulse oximetry, which is a calculation of oxygen content in the blood, chest x-ray picture, blood test to prove infection and identify the type of organism, urine test, and sputum sample checking are all part of the standard examination process for pneumonia patients. A lengthy review and a significant financial outlay become a challenge [2].

In the same way as humans use their prior experience to learn and solve new problems, neural networks are educated and validated on a variety of datasets. The network's learned information will then be used to train and validate new datasets.

Computer technology with the aid of GPU significantly increases the computing capacity of computers, and the use of deep learning approach for data processing reveals a steady growth pattern with the ongoing development of computer science and technology. Deep learning is a form of artificial neural network that has been improved.

In this project, I concentrated on CNN model to detect whether a given chest x-ray is having pneumonia or not. Also, I have enhanced the model using transfer learning to VGG-16 and predicted.

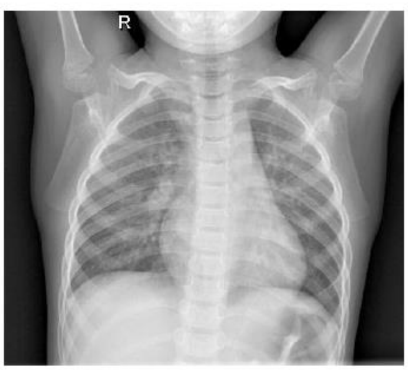
2. Dataset Description

In this project I have used chest x-rays dataset from Kaggle [3]. It totally contains 5856 images of children's chest x-rays which contains Normal, and Pneumonia effected images. The dataset consists of 4273 infected images and the remaining 1583 are normal images. These images are further divided into train, test and validation datasets that are fed into CNN model.

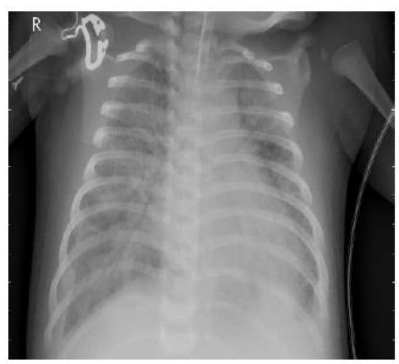
These train, test and validation folder further divided into Normal and Pneumonia folder which is having chest x-ray images.

No of images per each folder.

Folder	Train	Test	Validation
Normal	1341	234	8
Pneumonia	3876	390	8



Normal



Infected

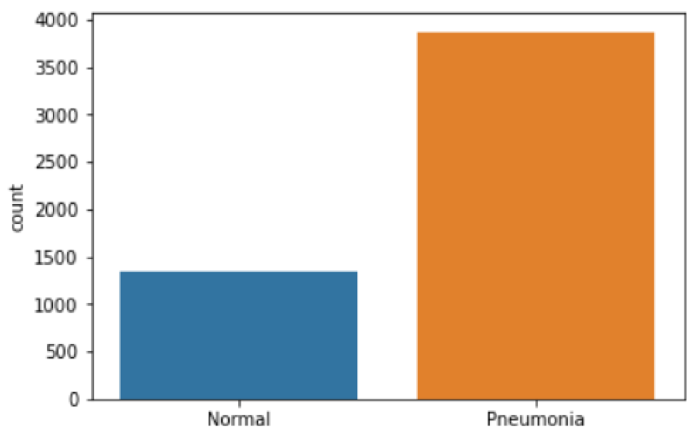
Dataset Link:

1. https://data.mendeley.com/datasets/rscbjbr9sj/2 [4]
2.https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

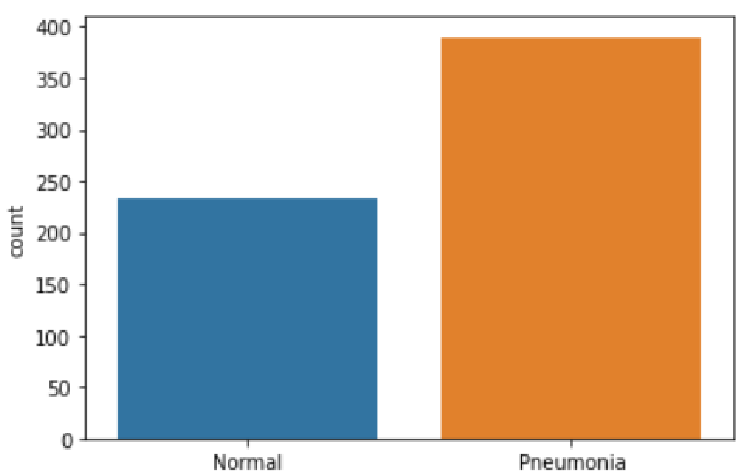
The picture of a normal left chest X-ray shows healthy lungs with no regions of pathological opacification. Bacterial pneumonia (middle) is characterized by a focal lobar convergence, as seen here in the right upper lobe (white arrows), while viral pneumonia is characterized by a diffuse interstitial pattern in right lung.

Anterior-posterior chest X-ray images were chosen from retrospective cohorts of pediatric patients aged one to five years old at Guangzhou Women and Children's Medical Center in Guangzhou. Both chest X-ray screening was done as part of the patients' regular medical treatment. [5]

Train Dataset

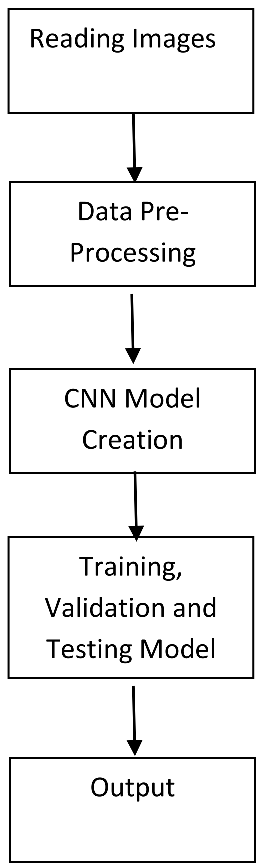


Test Dataset



3. Methodology

With the understanding of what needs to be done, over time, the advances in Computer Vision through Deep Learning have been developed and refined, predominantly by a specific algorithm, a Convolutional Neural Network. It can capture spatial and temporal dependencies in an image. Owing to the decrease in the number of parameters involved and the reusability of weights, the architecture is best suited to the image dataset [6].



Convolutional Neural-Network it is a type of Artificial Neural Network from which showed an impressive progress in image recognition and classification. It can assign importance to different aspects of the image in the input image, which shows learning weights and biases, and can be separated from each other.

Compared to other classification algorithms, the pre-processing needed by CNN is much lower. While the characteristics are hand-engineered in simple processes, with sufficient planning, CNN can learn these functions. It can capture spatial and temporal dependencies in an image. Owing to the decrease in the number of parameters involved and the reusability of weights, the architecture is best suited to the image dataset [7].

Convolutional layers, Activation functions, Pooling layer, and Fully Connected (FC) layer are the fundamental building blocks of a CNN-based framework. CNN is simply created by stacking these layers one after the other. A CNN's architecture is like that of Neurons connectivity pattern in the Human Brain and was influenced by the Visual Cortex organization.

- Convolution layer – It will achieve in feature recognition from original images.
- Pooling Layer – It will down sample the original image for quicker execution.
- Flattening Layer – It will convert the data into a 1-dimensional array for inputting it to the next layer.
- Fully Connected Layer – It is used to classify images to respective categories.

The network can be trained to better understand the complexities of the picture.

3.1 Data Pre-Processing and Considerations:

We must interpret and transform the images to numerical array style data to construct a classification model. It is expected that data resizing would help the model generalize better over noise and at the same time reduce memory needs for data processing.

- **Resizing of image:** As each image in the dataset is image with different pixels. If we use it without resizing, more memory can be consumed, raising the chances of the system crashing. Always, lesser memory usage often means the results come faster, which is an important consideration for real-time distraction detection. I will check the below two steps and then use one for building the model. In this model I have resized images to 50x50.

- **Image alternatives:** The model accuracy and speed changes based on the colored image vs grey scale image. To understand the distraction of the driver, the actions are important, and the actions will not be identified by color. While color-to-grayscale conversion can result in loss of information, but it can be handled [8].

- Few other operations are performed before building model like normalizing the images, separating the features & its labels, and converting the class labels to categorical values.

- **Data Augmentation & Normalization:** Data augmentation is a strategy for the diversity of a dataset without collecting any additional actual data, while also improving sample precision and preventing overfitting.

  Here I have done data augmentation for train, test, and validation dataset which is used for better performance. There after normalization is done to each image.

```
train_datagen = ImageDataGenerator(rescale= 1./255,
                   shear_range = 0.3,
                   horizontal_flip=True,
                   zoom_range = 0.3)

test_datagen = ImageDataGenerator(rescale= 1./255)

val_datagen = ImageDataGenerator(rescale= 1./255)
```

Normalization:

```
X_train = np.array(X_train) / 255
X_val = np.array(X_val) / 255
X_test = np.array(X_test) / 255
```

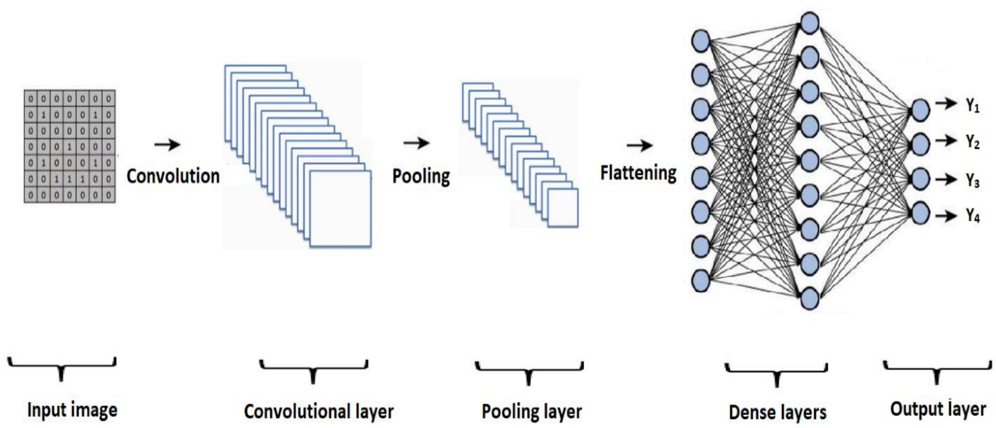
3.2 Building a Model:

All images in train, test and validation are further separated by features and label train into x_train (features) and y_train (labels) [9].

In this project I am using 3 CNN Layers,1 Flattening layer and Fully Connected Dense Layers.

Each Layer has:

- 3 Conv2D layers with input shape of 1x240x240 for 1st layer.
- 1 Activation layer
- 1 Pooling Layer
- 1 Dropout layer



Basic Architecture of CNN [10]

Implemented in TensorFlow 2.0 using Keras as high level api. Architecture used is Convolutional Neural Network (CNN). Below is the CNN model architecture with sequential which I implemented.

Implemented CNN Model

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 62, 62, 32)        320
_________________________________________________________________
activation (Activation)      (None, 62, 62, 32)        0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
activation_1 (Activation)    (None, 29, 29, 32)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 12, 12, 64)        18496
_________________________________________________________________
activation_2 (Activation)    (None, 12, 12, 64)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 2304)              0
_________________________________________________________________
dense (Dense)                (None, 1024)              2360320
_________________________________________________________________
activation_3 (Activation)    (None, 1024)              0
_________________________________________________________________
dropout (Dropout)            (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 1025
_________________________________________________________________
activation_4 (Activation)    (None, 1)                 0
=================================================================
Total params: 2,389,409
Trainable params: 2,389,409
Non-trainable params: 0
```

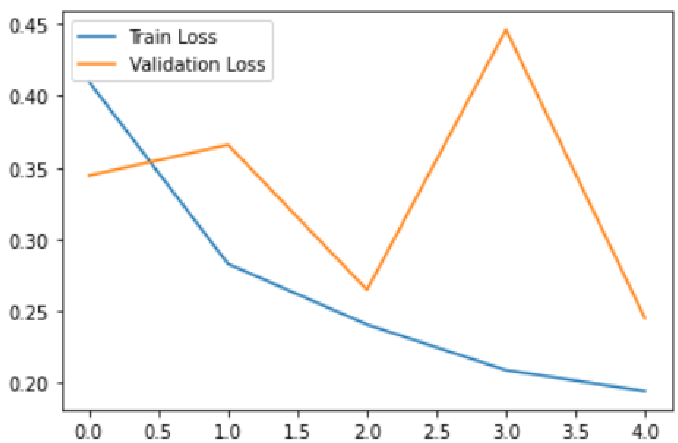
Compiled the model for further improvement with binary cross entropy loss, adam optimizer and metrics as accuracy. Used 25 epochs with validation data from the validation folder.

```
model.compile(loss="binary_crossentropy", optimizer="adam", metrics=["accuracy"])
```

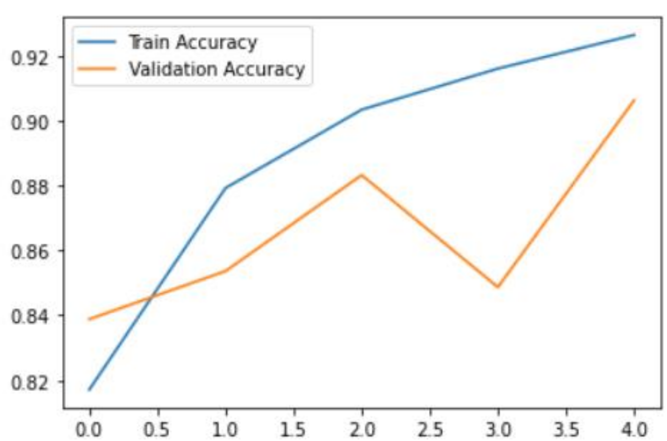
4. Experiments and Results

4.1 Evaluating the result:

- Model performance is calculated using the loss, metrics, and optimizer. Using this compilation, we can plot the graph of both train and test between Model training vs validation accuracy and loss trend for each epoch.
- To check if there are any imbalances in the prediction of the validation class, I will plot the confusion matrix.
- To find the precision, recall and F1 score to understand our model I will print the Classification report.



Train Loss vs Validation Loss



Train Accuracy vs Validation Accuracy

4.2 Validation on Test Data:

- Selecting random image from the test dataset and performing the steps involved in data preprocessing.
- Importing image from google and applying data preparation and predicting the google image.

4.3 Possible Improvements for the Model:

- By Increasing the number of Conv2D layers, more dense layers we can get better accuracy.
- **Accuracy :** Deep neural networks require a diversified collection of training data to train the model effectively. We used the Image Augmentation approach to address this problem. A new collection of images was generated for each epoch by creating different alterations for each image like zoom, left shift, right shift etc. Also, we can solve the error induced by variance from images captured under poor light conditions or from images captured from different angles.
- **Memory :** For the model to use less memory we need to resize the image in the better way. Which is resizing using padding.
- **Speed :** There are some state-of-the-art deep neural networks which can be considered for transfer learning like VGG16.

Conclusion

Here the proposed model is designed and developed to detect and classify Pneumonia from Chest X-ray images. Here Data Preprocessing and Convolutional Neural Network are used it starts by translating Chest X-ray images into sizes smaller than the original. Followed by defining and classifying images within the structure of the convolution neural network, which extracts and classifies characteristics from the images. Finding pneumonia using chest x-rays is easy but due to lack of radiologists and make faster this system is introduced which is accurate output. Got an accuracy of 90.5% and loss value of 0.24.

References

[1] N. I. f. H. a. Care, "Pneumonia in adults : diagnosis and management," 2014. [Online].

[2] R. R. W. a. T. L. Lemonovich, "Diagnosis and Management of Community-Acquired," pp. 1299-1306, 2011.

[3] P. Mooney, "Chest X-Ray Images (Pneumonia," Kaggle, [Online]. Available: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.

[4] "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification," Mendeley Data, [Online]. Available: https://data.mendeley.com/datasets/rscbjbr9sj/2.

[5] D. S. K. &. et.al., "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," [Online]. Available: https://www.cell.com/cell/fulltext/S0092-8674(18)30154-5.

[6] A. Bonner, "The Complete Beginner's Guide to Deep Learning: Convolutional Neural Networks and Image Classification".

[7] "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way," towardsdatascience, [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[8] S. Masood, A. Rai, A. Aggarwal and D. M.N, "Detecting Distraction of drivers using Convolutional Neural Network," *Pattern Recogniton Letetr (2018).*

[9] P. N. G. K. V. S. K. D. J. Rachna Jain, "Pneumonia Detection in chest X-ray images using Convolutional Neural Networks and Transfer Learning".

[10] †. C. E. G.-T. 1. Valeria Maeda-Gutiérrez 1, "Comparison of Convolutional Neural Network".

[11] A. Vidhya, "Learn Image Classification on 3 Datasets using Convolutional Neural Networks (CNN)," [Online]. Available: https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/.