Netaji Subhas University of Technology

CBCPC09
Operating System
Project File
Pneumonia Detection

Submitted by

Subhash Kumar (2020UEA6566)

Sunny Tomar (2020UEA6602)

Rajesh K. Yadav (2020UEA6608)

Prakhar Jain (2020UEA6609)

Course Instructor

Somiya Rani.

Abstract :

A large number of children die due to pneumonia every year worldwide. An estimated 1.2 million episodes of pneumonia were reported in children up to 5 years of age, of which 880,000 died in 2016. Hence, pneumonia is a major cause of death amongst children, with a high prevalence rate in South Asia and Sub-Saharan Africa. Even in a developed country like the United States, pneumonia is among the top 10 causes of deaths. Early detection and treatment of pneumonia can reduce mortality rates among children significantly in countries having a high prevalence. Hence, this paper presents Convolutional Neural Network models to detect pneumonia using x-ray images. Several Convolutional Neural Networks were trained to classify x-ray images into two classes viz., pneumonia and non-pneumonia, by changing various parameters, hyperparameters and number of convolutional layers. Six models have been mentioned in the paper. First and second models consist of two and three convolutional layers, respectively. The other four models are pre-trained models, which are VGG16, VGG19, ResNet50, and Inception-v3. The first and second models achieve a validation accuracy of 85.26% and 92.31% respectively. The accuracy of VGG16, VGG19, ResNet50 and Inception-v3 are 87.28%, 88.46%, 77.56% and 70.99% respectively.

Introduction :

Convolutional Neural Networks (CNNs) are inspired from the visual cortex of the brain and are used to solve difficult image-driven pattern recognition tasks, recognizing linear and non-linear patterns. CNN is good for image classification as less number of parameters and connections are required in such networks. This makes the training of such neural networks (CNNs) far easier compared to other neural networks. Artificial Neural Networks, on the other hand, have difficulty in computing image data in view of a high degree of computational complexity involved . Six models are presented in this paper to detect pneumonia in x-ray images, which will help control this deadly infection in children and other age groups. Pneumonia is caused by bacteria, viruses, or fungi in the air we breathe. The patient with pneumonia, have serious inflammation in the lung's air sacs, which get filled with fluid or pus, this makes it extremely difficult to breathe. Pneumonia can be mild or life-threatening. Hence, timely detection and treatment of pneumonia are critical for controlling high mortality rates among children due to this infectious disease in developing and developed countries. Six neural network models presented in this paper, diagnose pneumonia using x-ray images of the patients .Accuracy of the model is directly correlated with the size of the dataset that is, use of large datasets help improve the accuracy of the model, but there is no direct correlation between the depth of the model and the accuracy of the model.

VGG16, VGG19, ResNet50, and Inception-v3 are transfer learning models consisting of many layers. One of the biggest problems faced when developing deep networks is vanishing gradient. In vanishing gradient problems, during back propagation, the gradients become infinitesimally small, leading to loss of integral information. Due to this, the accuracy of the network saturates and then starts degrading. Different techniques have been employed by the models used in this paper to overcome the vanishing gradient problem. Training deep networks have certain restrictions viz., the dataset should be large, a large number of computational resources are used to achieve high performance, and the process of fine-tuning each parameter and hyper-parameter to achieve the optimum results is quite mundane.

Methodology :

1. **Dataset Collection**

-> The dataset was collected from the [Chest X-ray pneumonia dataset]. It contains three folders named train (for training), val (for validation), and test (for testing). Each of them has two sub-folders with label "NORMAL" and "PNEUMONIA".

2. **Importing necessary modules**

-> We used the following tools : TensorFlow, Keras, Sklearn, Numpy, Matplotlib. For feature extraction we used the VGG16 model.

3. **Data Augmentation and splitting**

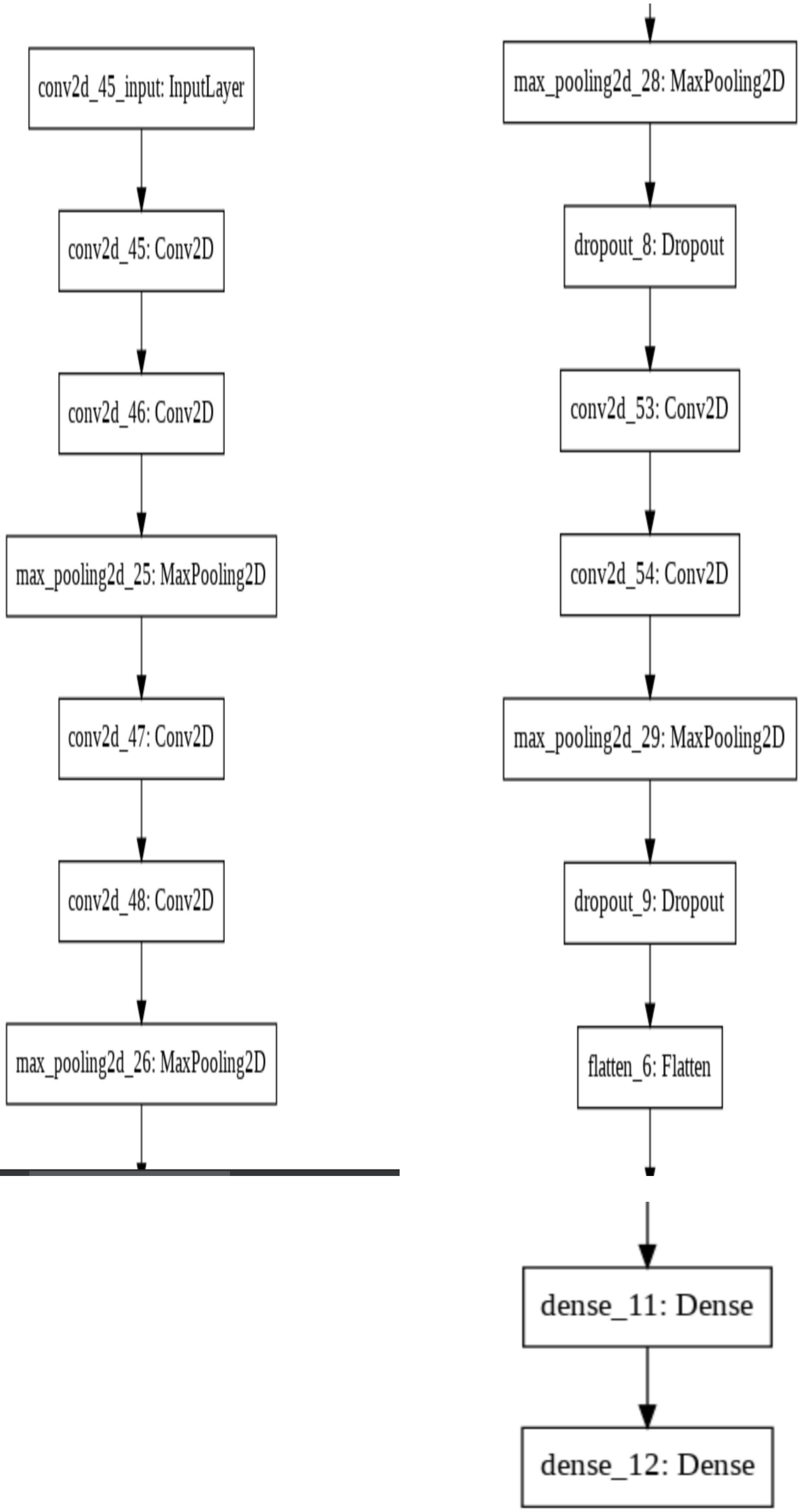
-> To perform the data augmentation, here the ImageDataGenerator class provided by Keras has been initialized. 5216 images are used for training and 624 are used for testing the model.

4. **Model and Architecture**

-> We have implemented a Sequential model with 7 layers (excluding output layer). The first 5 layers are convolutional layer and the last two are dense layers also Known as fully connected layers. Only for the first Conv layer, you need to define the input shape. The later layers take the previous layer's output as it's input padding is used to add extra pixel values of zero to the image.

We used the relu activation function in each layer except for the output layer, where we have used the sigmoid activation function for binary classification. The MaxPooling layer extracts the pixels with max values and use it for next steps. The Adam optimizer and binary cross entropy loss function is applied for training.

Out[ ]:



conv2d_45_input: InputLayer
conv2d_45: Conv2D
conv2d_46: Conv2D
max_pooling2d_25: MaxPooling2D
conv2d_47: Conv2D
conv2d_48: Conv2D
max_pooling2d_26: MaxPooling2D
max_pooling2d_28: MaxPooling2D
dropout_8: Dropout
conv2d_53: Conv2D
conv2d_54: Conv2D
max_pooling2d_29: MaxPooling2D
dropout_9: Dropout
flatten_6: Flatten
dense_11: Dense
dense_12: Dense

7. Training of the model

Training the data given as input to the model for 20 Epoch for 2 CNN model for better classification and accuracy.

1st CNN

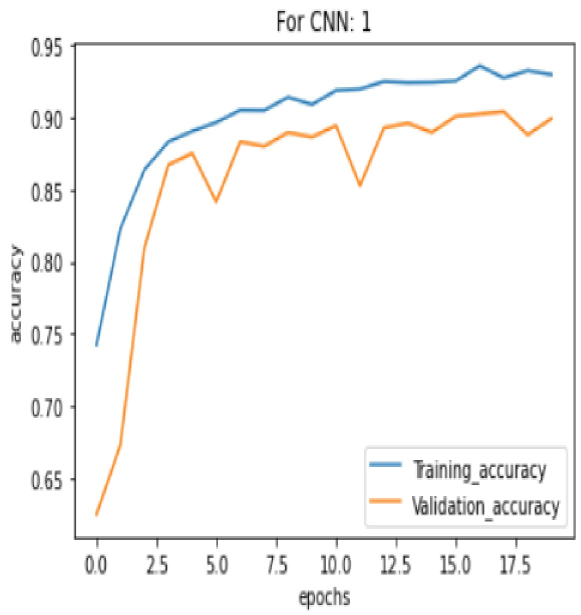
```
Epoch 15/20
81/81 [==============================] - 100s 1s/step - loss: 0.1918 - accuracy: 0.9241 - val_loss: 0.3580 - val_accurac
y: 0.8894
Epoch 16/20
81/81 [==============================] - 101s 1s/step - loss: 0.1847 - accuracy: 0.9251 - val_loss: 0.2663 - val_accurac
y: 0.9006
Epoch 17/20
81/81 [==============================] - 100s 1s/step - loss: 0.1787 - accuracy: 0.9356 - val_loss: 0.2737 - val_accurac
y: 0.9022
Epoch 18/20
81/81 [==============================] - 100s 1s/step - loss: 0.1711 - accuracy: 0.9272 - val_loss: 0.2802 - val_accurac
y: 0.9038
Epoch 19/20
81/81 [==============================] - 100s 1s/step - loss: 0.1730 - accuracy: 0.9323 - val_loss: 0.2760 - val_accurac
y: 0.8878
Epoch 20/20
81/81 [==============================] - 100s 1s/step - loss: 0.1778 - accuracy: 0.9295 - val_loss: 0.3045 - val_accurac
y: 0.8990
CNN 1: Epochs=20, Train accuracy=0.93556, Validation accuracy=0.90385
```

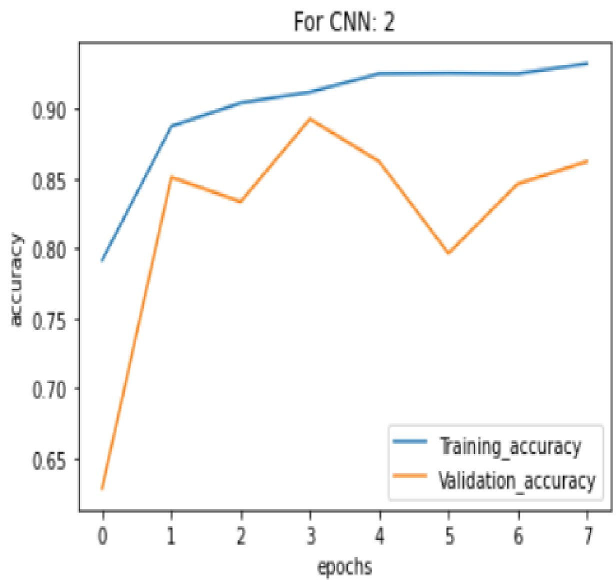
2nd CNN

```
Epoch 4/20
81/81 [==============================] - 101s 1s/step - loss: 0.2230 - accuracy: 0.9119 - val_loss: 0.2999 - val_accurac
y: 0.8926
Epoch 5/20
81/81 [==============================] - 100s 1s/step - loss: 0.2042 - accuracy: 0.9251 - val_loss: 0.3038 - val_accurac
y: 0.8622
Epoch 6/20
81/81 [==============================] - 101s 1s/step - loss: 0.1925 - accuracy: 0.9255 - val_loss: 0.3887 - val_accurac
y: 0.7965
Epoch 7/20
81/81 [==============================] - 101s 1s/step - loss: 0.1833 - accuracy: 0.9251 - val_loss: 0.3215 - val_accurac
y: 0.8462
Epoch 8/20
81/81 [==============================] - 102s 1s/step - loss: 0.1767 - accuracy: 0.9323 - val_loss: 0.2885 - val_accurac
y: 0.8622
CNN 2: Epochs=20, Train accuracy=0.93226, Validation accuracy=0.89263
```

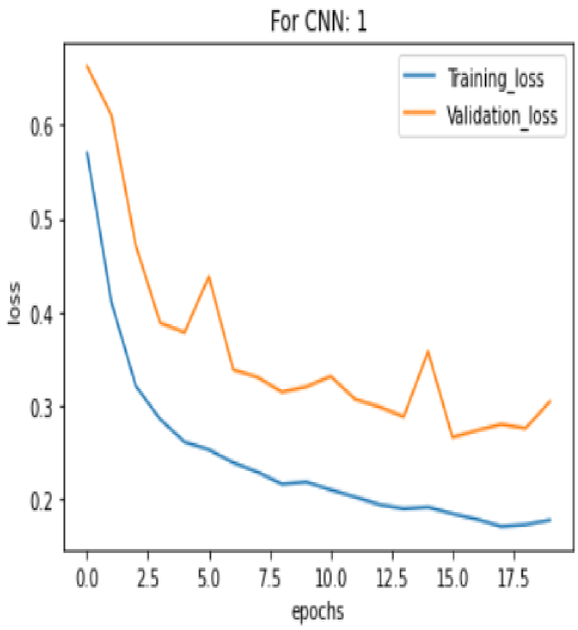
Result :

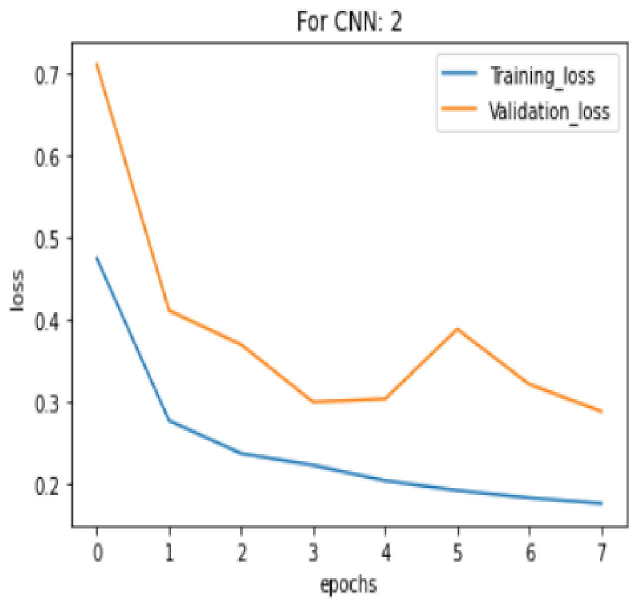
Accuracy plot for training and validation data





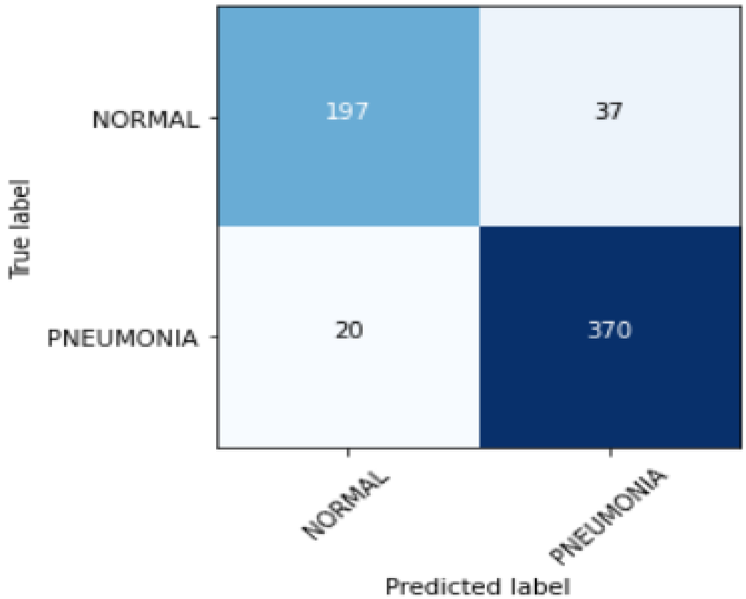
Loss plot for training and validation data





The confusion matrix was plotted without normalization with the help of The testing data.

```
Confusion matrix, without normalization
[[197  37]
 [ 20 370]]
```



The Classification Report

```
              precision    recall  f1-score   support

           0       0.91      0.84      0.87       234
           1       0.91      0.95      0.93       390

    accuracy                           0.91       624
   macro avg       0.91      0.90      0.90       624
weighted avg       0.91      0.91      0.91       624
```

Conclusion :

Disease detection such as Pneumonia through computer vision using Two convolutional neural network models. Both CNN models identified are assessed physically so that the framework in the feature extraction and fine-tuning are employed. We obtained Accuracy of 90% .

```
acc = accuracy_score(test_labels, results)*100
tn, fp, fn, tp = cm.ravel()
precision = tp/(tp+fp)*100
recall = tp/(tp+fn)*100
print('Accuracy: {0:0.2f}%'.format(acc))
print('Precision: {0:0.2f}%'.format(precision))
print('Recall: {0:0.2f}%'.format(recall))
print('F1-score: {0:0.2f}'.format(2*precision*recall/(precision+recall)))
```

```
Accuracy: 90.87%
Precision: 90.91%
Recall: 94.87%
F1-score: 92.85
```

Future Work :

There are several suggestions given below that could be implemented in the future to make the project more optimal:

(i) Using Different Model for Image Processing such as VGG19 or ResNet for faster training

(ii) Predicting other disease with same scan of chest data such as Covid-19 or lung cancer

(iii) Deploying web app where users can upload chest x-ray scan and this model at backend can Predict whether the user is suffering from Pneumonia or not.

References :

[1]. E. Sayed, et. al, Computer-aided Diagnosis of Human Brain Tumor though MRI: A Survey and a new algorithm, Expert System with Applications (41): 2014.

[2]. M. Cicero, A. Bilbily, E. Colak, T. Dowdell, B. Gra, K. Perapaladas, and J. Barfett, Training and Validating a Deep Convolutional Neutral Network for Computer-Aided Detection and Classification of Abnormalities on Frontal Chest Radiographs, Investigative Radiology 52(5) 281- 287, 2017.

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