Report

On

Pneumonia Detection

Prepared by

Ladani Utsav (18012011040)

**Guided
By**

Prof. Ketan J. Sarvakar

B.Tech Semester VII

Nov-Dec -2021

Subject: Deep Learning

Subject code: 2CEIT78PE1

Department of Computer Engineering

U. V. Patel College of Engineering

Ganpat University, Ganpat Vidyanagar – 384012

Abstract

Pneumonia is the common type of infection found in the world. The infection spreads in the lungs area of a human body. The chest x-ray is performed to diagnose this infection. Physicians use this X-ray image to diagnose or monitor treatment for conditions of pneumonia. This type of chest X-ray is also used in the diagnosis of diseases like emphysema, lung cancer, line and tube placement and tuberculosis. It may cause cough with phlegm, fever, chills and difficulty in breathing. This project aims to detect whether a person has pneumonia or not.

Introduction

One of the major factors associated with pneumonia in children is indoor air pollution. Apart from this, under-nutrition, lack of safe water, sanitation and basic health facilities are also major factors. Pneumonia is an interstitial lung disease caused by bacteria, fungi or viruses. This paper presents convolutional neural network models to accurately detect pneumonic lungs from chest X-rays, which can be utilized in the real world by medical practitioners to treat pneumonia. Four classification models were built using CNN to detect pneumonia from chest X-ray images to help control this deadly infection in children and other age groups. Accuracy of the model is directly correlated with the size of the dataset, that is, the use of large datasets helps improve the accuracy of the model.

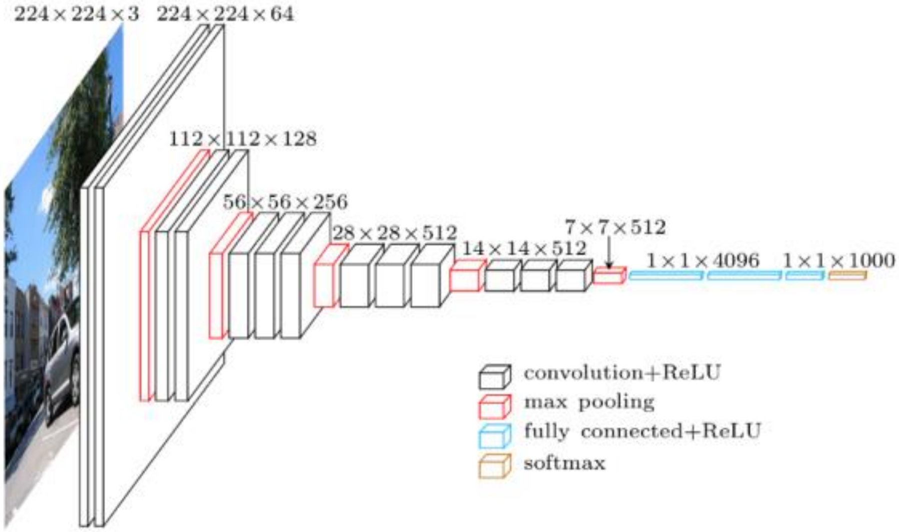
Architecture of CNN

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. Data augmentation has been applied to achieve better results from the dataset. Each model was trained for 20 epochs, with training and testing batch sizes.

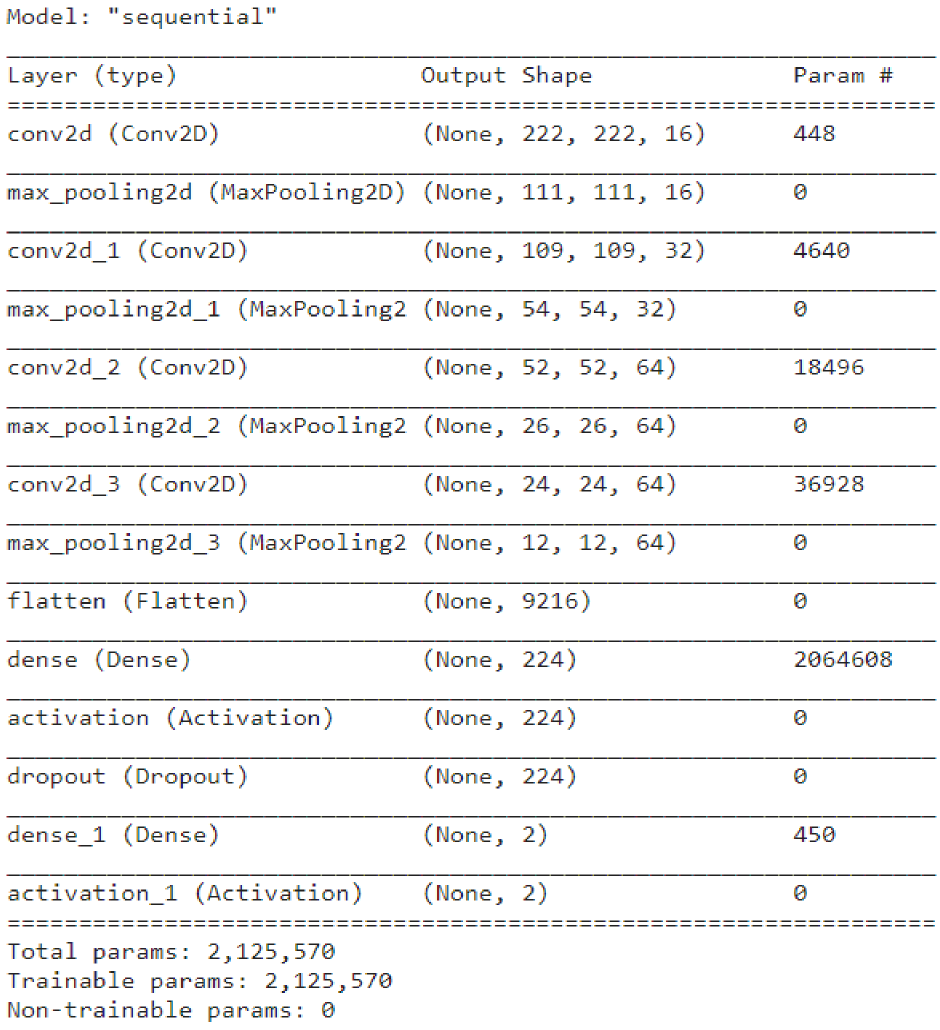
There are two block in ccn:-

The first block makes the particularity of this type of neural network since it functions as a feature extractor. To do this, it performs template matching by applying convolution filtering operations. The first layer filters the image with several convolution kernels and returns “feature maps”, which are then normalized (with an activation function) and/or resized.

This process can be repeated several times: we filter the features maps obtained with new kernels, which gives us new features maps to normalize and resize, and we can filter again, and so on. Finally, the values of the last feature maps are concatenated into a vector. This vector defines the output of the first block and the input of the second.



The second block is not characteristic of a CNN: it is in fact at the end of all the neural networks used for classification. The input vector values are transformed (with several linear combinations and activation functions) to return a new vector to the output. This last vector contains as many elements as there are classes: element i represents the probability that the image belongs to class i. Each element is therefore between 0 and 1, and the sum of all is worth 1. These probabilities are calculated by the last layer of this block (and therefore of the network), which uses a logistic function (binary classification) or a softmax function (multi-class classification) as an activation function.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d (Conv2D)              (None, 222, 222, 16)      448       
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 111, 111, 16)      0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 109, 109, 32)      4640      
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 32)        0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 52, 52, 64)        18496     
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 64)        0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 24, 24, 64)        36928     
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 12, 12, 64)        0         
_________________________________________________________________
flatten (Flatten)            (None, 9216)              0         
_________________________________________________________________
dense (Dense)                (None, 224)               2064608   
_________________________________________________________________
activation (Activation)      (None, 224)               0         
_________________________________________________________________
dropout (Dropout)            (None, 224)               0         
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 450       
_________________________________________________________________
activation_1 (Activation)    (None, 2)                 0         
=================================================================
Total params: 2,125,570
Trainable params: 2,125,570
Non-trainable params: 0
```

As with ordinary neural networks, the parameters of the layers are determined by gradient backpropagation: the cross-entropy is minimized during the training phase. But in the case of CNN, these parameters refer in particular to the image features.

The different layers of a CNN

There are four types of layers for a convolutional neural network: the convolutional layer, the pooling layer, the flattening layer and the fully-connected layer.

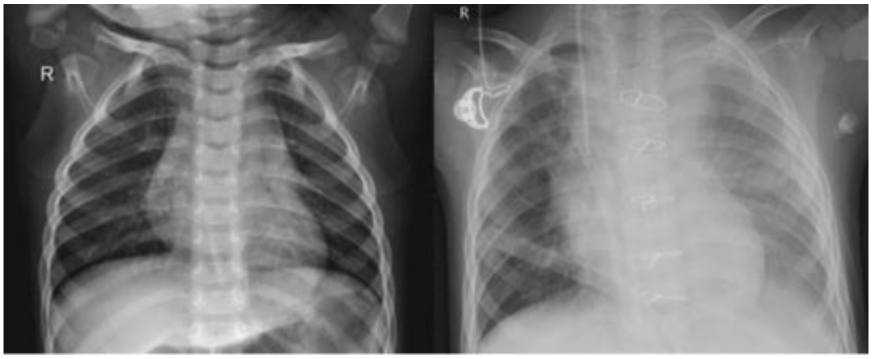
Convolutional layer : It is the building block of the CNNs. Convolution operation is done in mathematics to merge two functions . In the CNN models, the input image is first converted into matrix form. Convolution filter is applied to the input matrix which slides over it, performing element-wise multiplication and storing the sum. This creates a feature map.

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.

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.

Dataset

Chest X-Ray Images (Pneumonia) dataset has been imported from Kaggle ,with images split into Train, Test each divided into category Pneumonia and Normal.



Left image depicts normal lungs and right image depicts pneumonic lungs

The sample images from the dataset used during the research.

Tools and Libraries

Sr no.	Tools & Libraries	Description
1	TensorFlow	It is an open source artificial intelligence library, using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers. TensorFlow is mainly used for: Classification, Perception, Understanding, Discovering, Prediction and Creation.
2	Keras	Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine.
3	Numpy	We are using it for the image matrix handling
4	Kaggle	Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.
5	pandas	Pandas is mainly used for data analysis.

Dataset Name:

covid19-xray

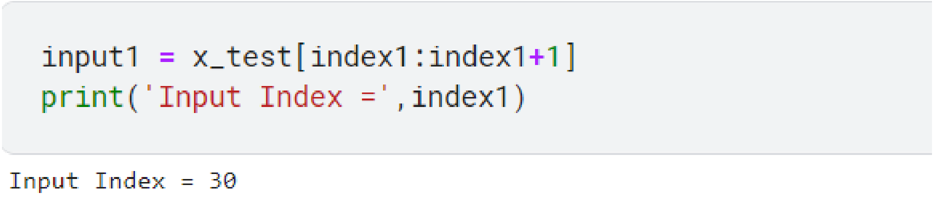
Dataset location:

https://www.kaggle.com/khoongweihao/covid19-xray-dataset-train-test-sets

GitHub Code link:

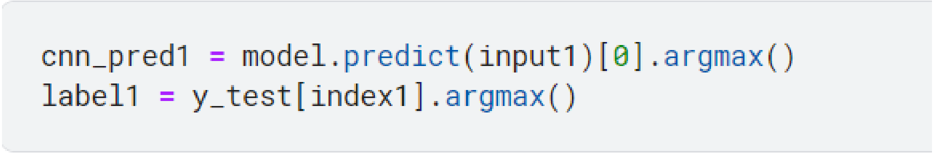
https://github.com/UtsavLadani/Pneumonia-Detection

Output

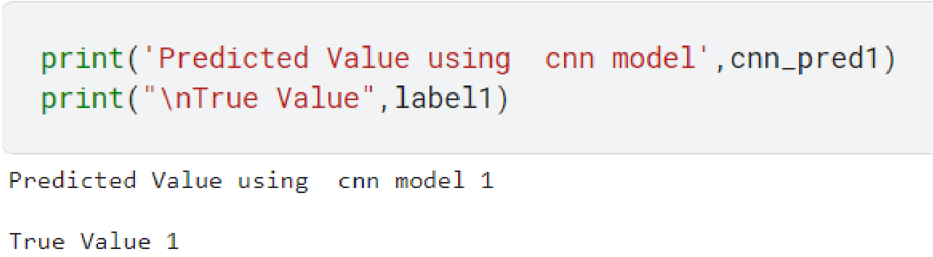


```
input1 = x_test[index1:index1+1]
print('Input Index =',index1)
```

```
Input Index = 30
```



```
cnn_pred1 = model.predict(input1)[0].argmax()
label1 = y_test[index1].argmax()
```



```
print('Predicted Value using  cnn model',cnn_pred1)
print("\nTrue Value",label1)
```

```
Predicted Value using  cnn model 1

True Value 1
```