PNEUMONIA PREDICTION

Using Convolutional Neural Networks

Developed by: Bindu D.P, Kavita L Kusagatti, Pooja G V , Sanskriti Kumar

Smart Bridge-Remote Summer Internship Program

1. INTRODUCTION

Over the recent years, Deep Learning (DL) has had a tremendous impact on various fields in science. It has led to significant improvements in image recognition. Over the last decades, we have witnessed the importance of medical imaging, *e.g.*, computed tomography (CT), magnetic resonance (MR), positron emission tomography (PET), mammography, ultrasound, X-ray, and so on, for the early detection, diagnosis, and treatment of diseases. However, due to large variations in pathology and potential fatigue of human experts, researchers and doctors have recently begun to benefit from computer-assisted interventions. While, compared to the advances in medical imaging technologies, it is belated for the advances in computational medical image analysis; it has recently been improving with the help of machine learning techniques. In the stream of applying machine learning for data analysis, meaningful feature extraction or feature representation lies at the heart of its success to accomplish target tasks. Conventionally, meaningful or task-related features were mostly designed by human experts based on their knowledge about the target domains, which thus made it challenging for non-experts to exploit machine learning techniques for their own studies. However, deep learning has relieved such obstacles by absorbing the feature engineering step into a learning step. That is, instead of extracting features in a hand-designed manner, deep learning requires only a set of data with minor preprocessing, if necessary, and then discovers the informative representations in a self-taught manner. So, now the burden of feature engineering has shifted from a human-side to a computer-side, thus allowing non-experts in machine learning to effectively use deep learning for their own researches and/or applications, especially in medical image analysis. The unprecedented success of deep learning arises mostly from the following factors: (1) advancements of high-tech central processing units (CPUs) and graphics processing units (GPUs); (ii) availability of a huge amount of data (*i.e.*, big data); (iii) developments of learning algorithms. Technically,

deep learning can be regarded as an improvement of the conventional artificial neural networks by building networks with multiple (more than two) layers. It is empirically shown that deep neural networks can discover hierarchical feature representations such that the higher level features can be derived from the lower level features . Thanks to its nice characteristic of learning hierarchical feature representations solely from data, deep learning has achieved record-breaking performance in a variety of artificial intelligence applications and grand challenges. Particularly, great improvements in computer vision inspired its use to medical image analysis such as image segmentation, image registration, image fusion ,image annotation ,computer-aided diagnosis and prognosis, lesion/landmark detection , and microscopic imaging analysis , to name a few.

1.1 Overview

There is a great growing interest in the domain of deep learning techniques for identifying and classifying images with various datasets. An enormous availability of datasets has developed a keen interest in deep learning. Pneumonia is a disease that is caused by various bacteria, virus etc. X-ray is one of the major diagnosis tools for diagnosing pneumonia. It is a fact that the disease like pneumonia is spreading very vast and also its threat is very tremendous and causing a barrier in developing a disease free nation. It has been predicted by WHO that 4 million sudden misfortunes happen each year from nuclear family air tainting diseases, maximum people are suffering from pneumonia disease. Also, it has been found in a survey that approx. 160 million people suffered from pneumonia in which there were children of fewer than 5 years of age .This model is a convolutional neural system (CNN) model prepared without any preparation to group and identify the occurrence of pneumonia disease from a given assortment of chest X-ray image tests. Dissimilar to different strategies that depend exclusively on more learning draws near or conventional carefully assembled systems to accomplish an amazing grouping execution, and developed a convolutional neural arrange model without any preparation to separate and character the images to decide whether an individual is suffering with pneumonia. This model could help alleviate the dependability and difficult challenges frequently confronted to manage therapeutic problems. In this paper, CNN algorithm has been used along with different data augmentation techniques for improving the classification accuracies which has been discussed to increase the performance which will help in improving the validation and training accuracies and characterization of exactness of the CNN model and accomplished various results. This experiment was carried out using python language and has shown improved outcomes.

1.2 Purpose

Our aim from the project is to make use of tensorflow, scikit, & flask libraries from python to extract the libraries for machine learning for the pneumonia prediction. Then, to predict whether pneumonia or normal we have used techniques of convolutional neural network algorithms and withdrawing the conclusions.

2. LITERATURE SURVEY

CNNs are one of the recent advance techniques which are able to perform better feature extraction in an efficient way but it is mostly suitable for the sequential data. Recently, many data scientists have proved that using CNNs in Deep Learning will improve the performance of the algorithms and theses scientists have used energy physics for the particle collision analysis in energy physics which has shown great results. Therefore, CNNs have proved very efficient in classification tasks used in Deep Learning.

2.1 Existing Problem

In general, a patient suffering from Pneumonia goes to the hospital to take an X-ray image waits for the doctor and then the doctor will check the X-ray then he decides whether the person has pneumonia or not. The results are not only concluded based on just seeing the X-ray images but furthermore, tests were conducted on the patient to verify the results of the doctor. The process is time-consuming and if the patient has severe pneumonia or not he has to wait several days to get the test results. But in recent developments of the artificial intelligence and the computational powers of the computers have increased it helps in predicting pneumonia by just passing the X-ray image as an input to our model.

2.2 Proposed Solution

The main objective of this project is to help the doctors to predict the pneumonia disease more accurately using a deep learning model. The objective is not only to help the doctors but also to the patients to verify whether they have pneumonia or not. By using this model, we can precisely predict pneumonia. A convolutional neural network model is built from scratch to extract features from a given chest X-ray image and classify it to determine if a person is infected with pneumonia. a web is built where the user can upload the x - ray image and the result is shown on the UI.

3. THEORETICAL ANALYSIS

When it comes to Machine Learning, Artificial Neural Networks perform really well. Artificial Neural Networks are used in various classification tasks like image, audio, words. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use **Convolution Neural Network**.

To achieve our goal, we have used one of the famous machine learning algorithms out there which is used for Image Classification i.e. Convolutional Neural Network(or CNN).As we know its a machine learning algorithm for machines to understand the features of the image with foresight and remember the features to guess whether the name of the new image fed to the machine. At first we created our very own dataset which includes the x-ray images which consist of both normal and one with pneumonia. Now after getting the data set, we pre-process the data a bit and provide labels to each of the images provided.

Libraries used:

- DataGenerator- The ImageDataGenerator is an easy way to load and augment images in batches for image classification tasks.
- tensorflow – To add layers as well as compare the loss and adam curve our result data or obtained log.

Layers used to build ConvNets

A convnets is a sequence of layers, and every layer transforms one volume to another through differentiable functions.

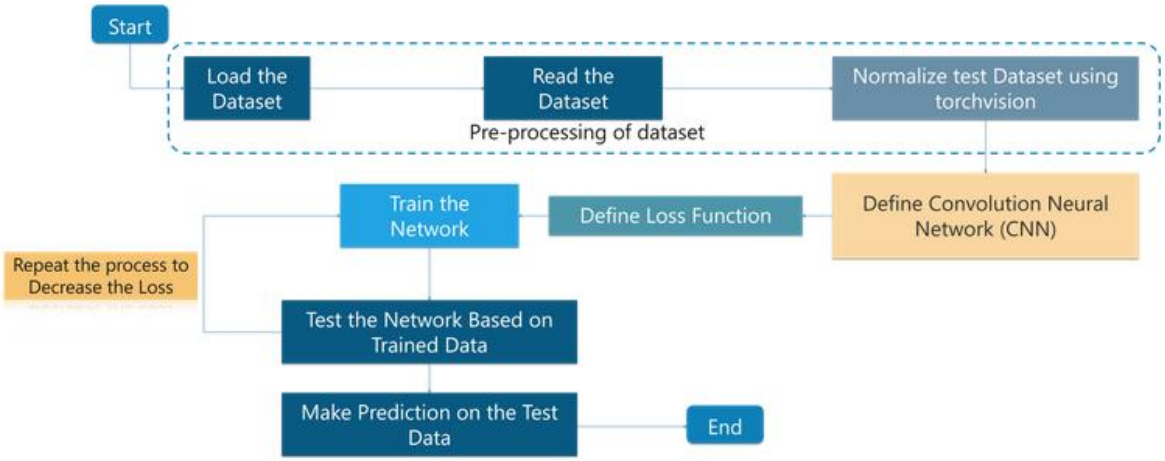
Types of layers:

1. **Input Layer**: This layer holds the raw input of the image.
2. **Convolution Layer:** This layer computes the output volume by computing dot product between all filters and image patches.
3. **Activation Function Layer**: This layer will apply element wise activation function to the output of the convolution layer. Some common activation functions are RELU, Sigmoid, Tanh, Leaky RELU, etc. We have used RELU as well as softmax for our model as this is a multi classification problem.
4. **Pool Layer:** This layer is periodically inserted in the convNets and its main function is to reduce the size of volume which makes the computation fast, reduces memory and also prevents from overfitting. Two common types of pooling layers are max pooling and average pooling.
5. **Dense layer -** It is the regular deeply connected neural network **layer**

Accuracy is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set.

We have used Non-Linearity (ReLU) activation function. ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = max(0, x)$. ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values. We have used the softmax function as this is a multi class classification problem.

3.1 Flow Chart



3.2 Software Designing

● Jupyter Notebook Environment

● Spyder Ide

●Deep Learning Algorithm (CNN)

● Python

● HTML

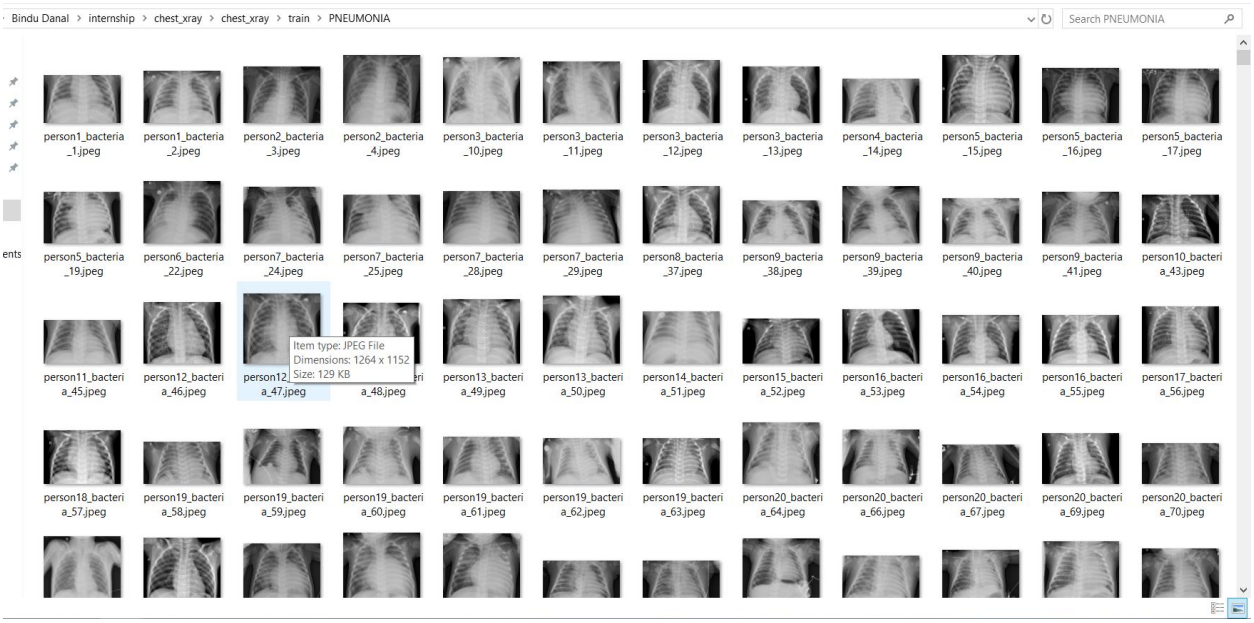
● Flask

We developed the Pneumonia prediction using x-rays by using the Python language which is an interpreted and high level programming language and using the Deep Learning algorithms. For coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language. For pneumonia prediction we used the Flask. It is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a web page is HTML by creating the

templates to use in the functions of the Flask and HTML.

4. EXPERIMENTAL INVESTIGATION

In this project, the dataset we downloaded from some dataset available on Kaggle. It contains subfolders namely train_set, test_set . Each set consist of both normal and pneumonic x-ray images. The dataset consist total of 5844 images. The screenshot of the pneumonia x-rays has been attached below:



5. RESULT

We trained and tested our algorithms on the complete data set to start with. Later we randomly separated the data set into training data and test data so that we had samples from each class.80% of data is used for training data and 20% is used for test data. The dataset consists of 5844 pictures and was used as a development set for CNN. The model was able to classify more than 90% of the images. The testing accuracy of the system is about 93%. Depending on the classification. Depending on the number of epochs we give, the accuracy and loss value varies. The following figures and tables show the results we observed:

```
In [19]: model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 21, 21, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 10, 10, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 8, 8, 32)          9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 4, 4, 32)          0
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 128)               65664
_________________________________________________________________
dense_1 (Dense)              (None, 128)               16512
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 129
=================================================================
Total params: 92,449
Trainable params: 92,449
Non-trainable params: 0
_________________________________________________________________
```

Fig 1. Represents the model.summary()

```
In [21]: from keras.preprocessing.image import ImageDataGenerator
```

```
Using TensorFlow backend.
```

```
In [22]: train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,horizontal_flip=True,zoom_range=0.2)
         test_datagen=ImageDataGenerator(rescale=1./255)
```

```
In [29]: x_train=train_datagen.flow_from_directory(r'C:\Users\hp\internship\chest_xray\chest_xray\train',target_size=(64,64),batch_size=3
         x_test=test_datagen.flow_from_directory(r'C:\Users\hp\internship\chest_xray\chest_xray\test',target_size=(64,64),batch_size=32,cl
```

```
Found 5216 images belonging to 2 classes.
Found 624 images belonging to 2 classes.
```

```
In [24]: print(x_train.class_indices)
```

```
{'NORMAL': 0, 'PNEUMONIA': 1}
```

```
In [25]: print(x_test.class_indices)
```

```
{'NORMAL': 0, 'PNEUMONIA': 1}
```

```
In [26]: model.fit_generator(x_train,steps_per_epoch=5216,validation_data=x_test,epochs=1,validation_steps=624)
```

```
5216/5216 [==============================] - 4298s 824ms/step - loss: 0.1671 - accuracy: 0.9329 - val_loss: 0.3625 - val_accura
cy: 0.8977
```

```
Out[26]: <tensorflow.python.keras.callbacks.History at 0x22a77c3a088>
```

```
In [28]: plt=model.save('mymodel1.h5')
```

Fig 2. Represents model fitting with epoch=1.

jupyter Pneumonia CNN model Last Checkpoint: Last Tuesday at 6:51 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [15]:
```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

In [16]:
```
print(x_train.class_indices)
```

```
{'NORMAL': 0, 'PNEUMONIA': 1}
```

In [17]:
```
model.fit_generator(x_train,steps_per_epoch=5216//32,validation_data=x_test,epochs=25,validation_steps=624//32)
```

```
WARNING:tensorflow:From <ipython-input-17-4db775e1eefc>:1: Model.fit_generator (from tensorflow.python.keras.engine.training) i
s deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
WARNING:tensorflow:sample_weight modes were coerced from
  ...
    to
  ['...']
WARNING:tensorflow:sample_weight modes were coerced from
  ...
    to
  ['...']
Train for 163 steps, validate for 19 steps
Epoch 1/25
163/163 [==============================] - 778s 5s/step - loss: 0.4871 - accuracy: 0.7788 - val_loss: 0.4083 - val_accuracy: 0.
8273
Epoch 2/25
163/163 [==============================] - 719s 4s/step - loss: 0.2588 - accuracy: 0.8900 - val_loss: 0.4360 - val_accuracy: 0.
8010
```

Fig 3. Represents model fitting with epoch=25.

In [5]:
```
from skimage.transform import resize
```

In [6]:
```
def detect(frame):
    try:
        img=resize(frame,(64,64))
        img=np.expand_dims(img,axis=0)
        if(np.max(img)>1):
            img=img/255.0
        prediction=model.predict(img)
        print(prediction)
        print(model.predict_classes(img))
    except AttributeError:
        print("shape not found")
```

In [7]:
```
frame=cv2.imread(r"C:\Users\hp\internship\chest_xray\chest_xray\test\PNEUMONIA\person1685_virus_2903.jpeg")
data=detect(frame)
```

```
[[0.99772877]]
[[1]]
```

In [8]:
```
frame=cv2.imread(r"C:\Users\hp\internship\chest_xray\chest_xray\test\NORMAL\IM-0001-0001.jpeg")
data=detect(frame)
```

```
[[0.05135697]]
[[0]]
```

Fig 4. Represents the prediction in ipynb

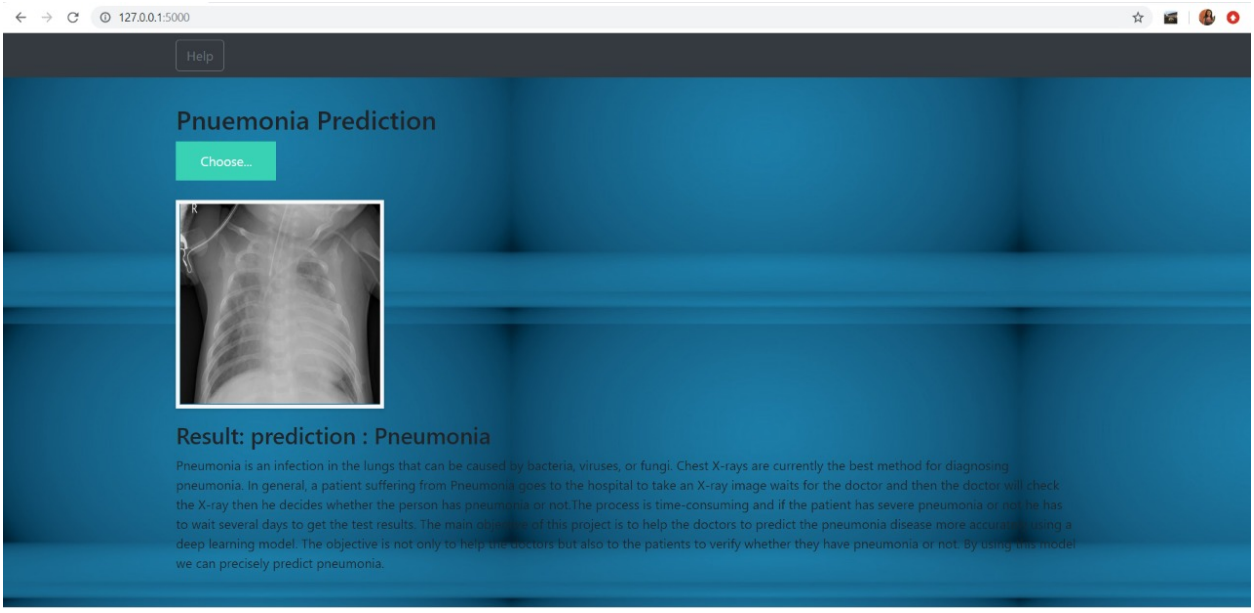


Fig 5. Represents the final web page done with flask application

6. APPLICATIONS:

- CNNs are now-a-days widely used in the computer vision and automation fields. This helps in developing such artificial systems which has capability of performing complex tasks with efficiency.
- CNNs are also being used in the domain of natural language processing for language analysis, language modeling, language designing. CNN models helps in determining the various semantics of any sentence for knowing the better about the client's requirements.
- CNNs are being used for object detection purpose for identifying the objects in the way. Segmentation of images is also being done using the CNNs.
- Image Classification is one of the very important task which is done using the CNNs in the present scenario by various data augmentation techniques and feature extraction techniques.
- One of the most important applications is the speech recognition in which the speech is

being recognized using some automated devices. For example, Google's speech recorder.

- CNNs are also widely used for the data which are computationally very limited in resources. There are several techniques which are still being working on small datasets with improved accuracy of classification.
- CNNs are also being used for the images which are having low resolution. Many researchers have given different techniques to work on the images having low resolution using CNN.

7. ADVANTAGES AND DISADVANTAGES OF CNN MODEL

Advantages:

- Except from the improvements in precision observed in the classification/prediction problems at the surveyed works, there are some other important advantages of using CNN in image processing. Previously, traditional approaches for image classification tasks were based on hand-engineered features, whose performance and accuracy greatly affected the overall results. Feature engineering (FE) is a complex, time-consuming process which needs to be altered whenever the problem or the data set changes. Thus, FE constitutes an expensive effort that depends on experts' knowledge and does not generalize well.

- Convolutional neural networks seem to generalize well and they are quite robust even under challenging conditions such as illumination, complex background, size and orientation of the images, and different resolution. It helps farmers to identify which animals are causing harm to their crops.so They can take preventive measures.

Disadvantages:

- The main disadvantage is that CNN can sometimes take much longer to train. However, after training, their testing time efficiency is much faster than other methods

- Other disadvantages include problems that might occur when using pre-trained models on

similar and smaller data sets, optimization issues because of the models' complexity, as well as hardware restrictions.

8. CONCLUSION

In the given study, it has been demonstrated that how one can classify the true and false cases of pneumonia easily from a small dataset of X-ray images. This model was basically built from the scratch that helps in separating it from all other existing methods like transfer learning etc. The proposed method will further help in effective diagnosing the pneumonia patients more easily and this CNN approach is computationally effective. For the future work, the proposed work can be extended to further classify and detect lung cancer and pneumonia X-ray images as classifying the two diseases has become one of the major concerns now-a-days. Therefore, possible solutions for considering these problems are great area for the research domain.

9. BIBLIOGRAPHY

- https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia for dataset.
- Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, Aurelien Geron
- https://mc.ai/build-a-web-app-to-detect-pneumonia-from-chest-x-ray-images-part-2-building-the-web-app/