Classification of Medical Health Records Using Convolutional Neural Networks for Optimal Diagnosis

M. H. Chaithra and S. Vagdevi

Abstract Pneumonia is considered to be one of the lungs affecting inflammation for small air sacs. Dry cough, chest pain, fever, and breathing difficulty are some of the common symptoms during this situation. The seriousness of the condition of the patient is variable based on several parameters. Viruses, bacteria, and by other microorganisms usually cause pneumonia. Some of the risk factors during this situation are: cystic fibrosis, chronic obstructive pulmonary disease (COPD), asthma, diabetes, and heart failure. Sometimes a weak immune system may also increase the severity of the situation. The medical diagnostics using machine learning powered by computer vision and deep learning will help us to extract useful information by filtering out the non-essential and insignificant information from the diagnosis report. Computer vision, neural networks, and artificial intelligence methods like convolutional neural network will lead to identify and extract the useful information from the diagnosis report, and in turn, it will help to assist in medical diagnosis. In this regard, the main objective of this work is to classify disease based on symptoms. Clinical and laboratory symptoms are considered as the basic for this investigation.

Keywords Machine learning · Computer vision · Artificial neural networks · Convolutional neural networks · Medical diagnostics · Data augmentation · Image acquisition

1 Introduction

As per the research released by the Indian Journal of Medical Research under the title "Doctor Population ratio of India—The Reality," it has been estimated that six

M. H. Chaithra (✉)
Department of Computer Science and Engineering, REVA University, Bangalore, India

Visvesvaraya Technological University, Belagavi, Karnataka, India
e-mail: chaithra.mh14@gmail.com

S. Vagdevi
Department of Computer Science and Engineering, City Engineering College, Bangalore, India

D. P. Agrawal et al. (eds.), *Cyber Security, Privacy and Networking*, Lecture Notes
in Networks and Systems 370, https://doi.org/10.1007/978-981-16-8664-1_22

lakh doctors and twenty lakh nurses shortage of medical personnel in India. India is planning to establish two hundred new medical colleges in next decade to meet the above requirement. Also, the cost of medical treatment is increasing for a common man in India and 65% of the health expenditure in borne by the individual itself, and as per recent release of the data by the government of India, it has been estimated that the medical expenses push 57 million people into poverty each year.

The medical diagnostics using machine learning powered by computer vision and deep learning will help us to extract useful information by filtering out the non-essential and insignificant information from the diagnosis report [1]. Computer vision, neural networks, and artificial intelligence methods like convolutional neural network will lead to identify and extract the useful information from the diagnosis report, and in turns, it will help to assist in medical diagnosis. We will train and develop the medical diagnostics tool which will help organization or government or user which will assist doctors/medical personnel in medical diagnosis.

2 Background

The National Electronic Health Records Survey (NEHRS) is an annual comprehensive survey of employed, office-based physicians. Usually based on the role of diet and prescribed exercise, the health risks are evaluated and research will be conducted. Recently, many researchers have achieved promising results based on electronic database and applying computational techniques. At the same time, secured data and maintaining patient's privacy are also primary concerns while maintaining health records electronically. This research article provides an optimal method to identify a specific disease by suitable computational methods and also justifies the reliability on developed system.

3 Objectives

As per the above scope, the following objectives are defined in this research work:

- Medical diagnostics using machine learning
- Developing a medical diagnostics tool
- Medical diagnostics tool powered by machine learning and deep learning will high prediction capability than the traditional models
- Identifying and extract the most critical/important information from the diagnostic
- Reducing the manual touchpoint while performing the model diagnostic.

4 Proposed Process Flow

To carry out the proposed research work, the resources needed are—artificial neural network, computer vision, and Python. Hardware requirements are 32 GB RAM, 1 TB Hard Disk, Window/Linux Machine. The potential challenges and risks involved are different sources of data which will have different patterns and quality of data which usually lot of efforts to prepare and clean for analysis [2]. Privacy of the data is also one of the challenges in such types of domain. Figure 1 presents the overall flow of the proposed model.

Around 6000 JPEG X-ray images are considered for detecting pneumonia condition. Training, testing, and validation are the stages in which different subfolders of images are distributed [3]. Anterior–posterior chest images are selected from pediatric patients for this study. Clinical and laboratory symptoms are considered which selecting chest images for the investigation [4]. Several chest radiographs are filtered to remove images with noise, poor quality, or unreadable data. Finally, the filtered images are certified by experts before being used for training purpose for our model. In this phase, the grading errors are recorded and discarded from the training database.

5 Methodology

5.1 Dataset Collection

One of the major potential challenges for this work is to obtain relevant medical data. As mentioned earlier, around 6000 JPEG images are been considered from unique patients for this study. Based on the associated radiology reports, the text contents are extracted and used for classification using preprocessing phases of language processing tools [5]. A unique labeling process is adopted to disambiguate and group the images according to the clinical text data as per the proposal in the arti-

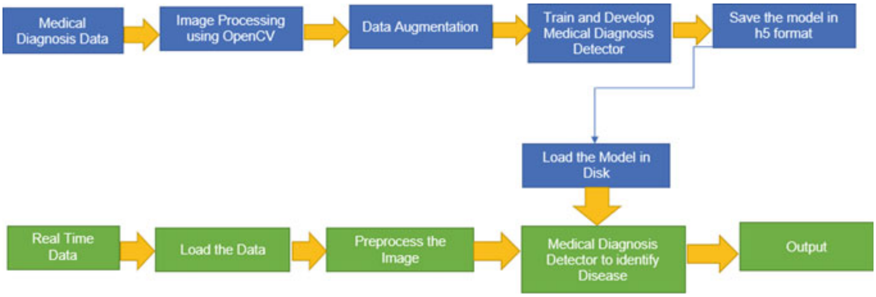

Medical Diagnosis Data
Image Processing using OpenCV
Data Augmentation
Train and Develop Medical Diagnosis Detector
Save the model in h5 format
Load the Model in Disk
Real Time Data
Load the Data
Preprocess the Image
Medical Diagnosis Detector to identify Disease
Output

Fig. 1 Proposed flow diagram indicating all involved modules

cle, “ChestX-ray8: Hospital-scale Chest X-ray database and benchmarks on weakly supervised classification and localization of common thorax diseases” (Wang et al.). This dataset includes 12 zip files, and each of them is of size 2–4 GB. The technologies used are: Keras, Python, Spyder, Jupyter, OpenCv, TensorFlow, and image acquisition through CNN datasets.

The typical text preprocessing steps involved are: removal of white space, expanding the contraction, removing noise, special character, normalizing all text to lower case, finding the maximum length of the text, tokenization, stop word removal, and stemming/lemmatization.

5.2 Preprocessing

Standardization of the features is conducted by standardizing pixel values among the whole database. This action is applied for each column in a tabular database [6]. Feature-centric and feature standard normalization parameters are used to standardize the entire image data generation class. This process is monitored closely in order to avoid multiple arguments with same effect. Otherwise, the redundant entries need to be filtered which will be an added effort. Typical image processing algorithms are applied on these digitized images. There are many applications of digital image processing algorithms as compared to analog processing. Majority of digital image processing algorithms help in enhancing image features by eliminating noise or skewed images. These enhanced image parameters considerably improve in developing artificial intelligent computer models. Typical image processing phases include—reading image, resizing it, de-noise (if any), normalize it, segment, and smooth edges as per the needs.

6 Model Building

The typical neural network model and convolutional neural network model are shown in Figs. 2 and 3. Convolutions are meant to extract key features from the input images. By learning image features, they ensure the relationship among pixels of input images [7]. The two inputs such as image matrix and kernel or filter are considered for a mathematical operation. We all know how to generate the volume dimension as output from an image matrix of dimension using relevant filter [8]. The convolution of image matrix multiplies with filter matrix to generate feature map. Strides in convolution layer are significant.

This section provides a complete overview on the developed model along with the code samples. Figures 4 and 5 show the code for preprocessing steps and visualization steps.

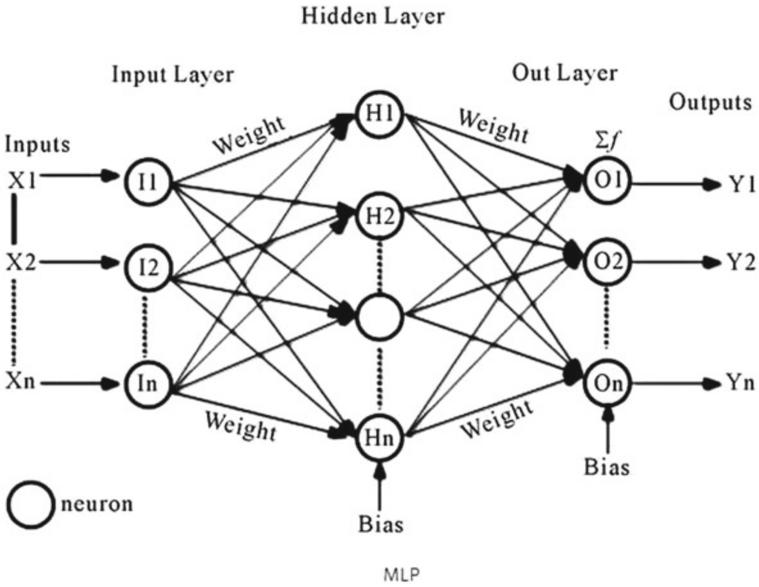


Fig. 2 Simple neural network architecture

It is noticed that the feature map size is smaller than the input size. The feature map has to be avoided from shrinking [9] with the help of padding process. Zero-valued pixels are added around the input in order to avoid shrinking of feature map. This will ensure that the spatial size remains constant. Padding confirms improvement in the performance, and kernel size is constantly maintained. When the input images are large, then the number of arguments will be reduced by pooling. By retaining the key features, dimension reduction happens through downsampling. Various downsampling methods are maxpooling, average pooling, and sum pooling [10]. Suitable bias values are applied to ensure an efficient activation function. When the derivatives are steeper, several neurons will get destroyed which results in a passive network. When the epoch is consuming huge time to run, then it is decomposed into batches. Binary cross-entropy measures are used to average the class-wise errors. Adam optimizer is used to update network weights iterative based on training data. It pursues a single learning rate for all weights updates. The learning rate is undisturbed during training process. Convolution layer is basically a feature detector that automatically tries to learn to filter out the not needed information. Pooling layers reduce the memory size required for processing and also detect object characteristics at some unusual places.

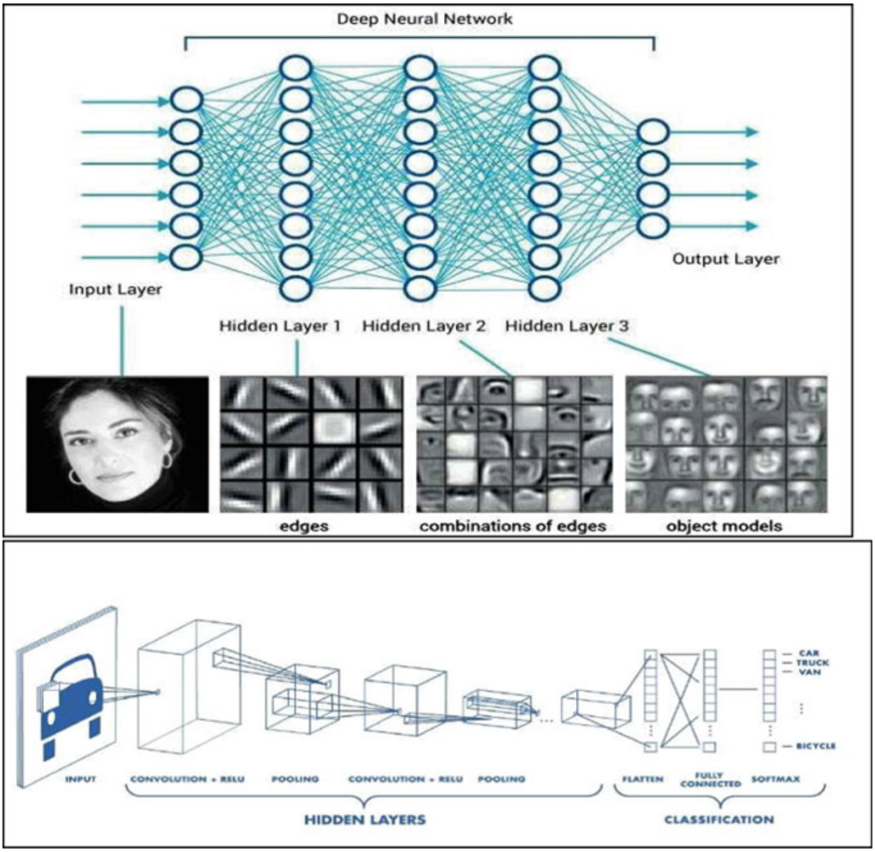
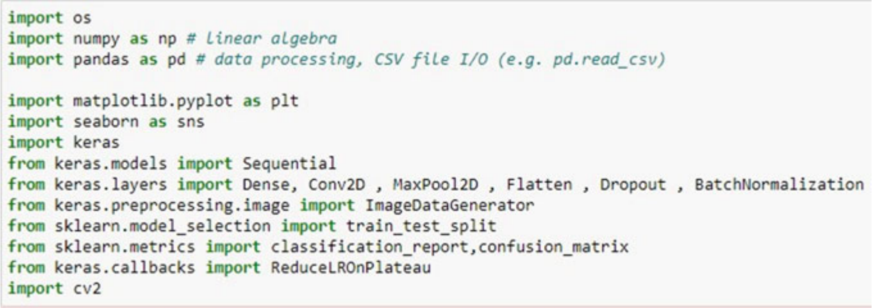


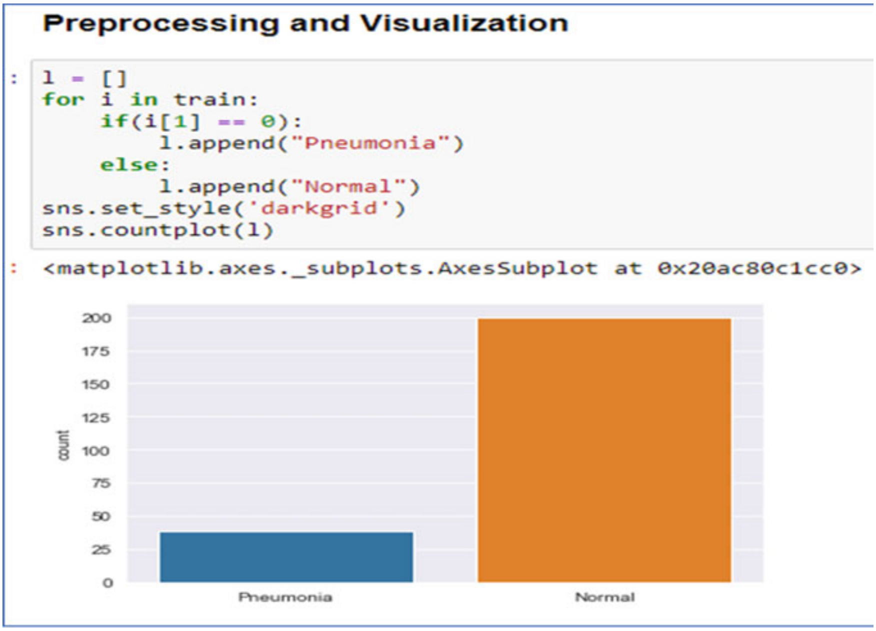
Fig. 3 Convolutional neural network architecture



```
import os
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import matplotlib.pyplot as plt
import seaborn as sns
import keras
from keras.models import Sequential
from keras.layers import Dense, Conv2D , MaxPool2D , Flatten , Dropout , BatchNormalization
from keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,confusion_matrix
from keras.callbacks import ReduceLROnPlateau
import cv2
```

Fig. 4 A sample code showing preprocessing and visualization

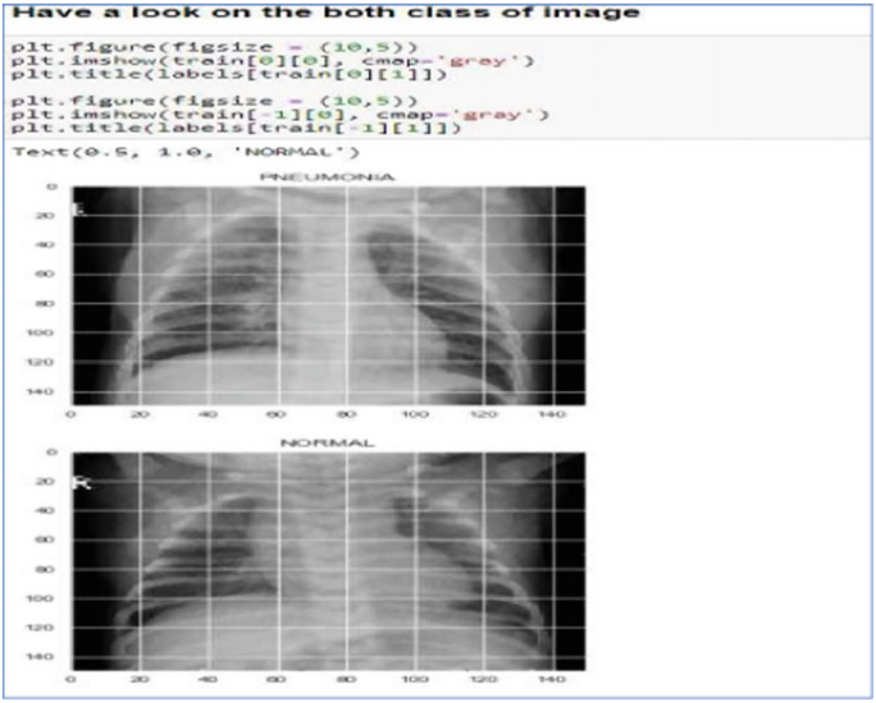


Preprocessing and Visualization

```
l = []
for i in train:
    if(i[1] == 0):
        l.append("Pneumonia")
    else:
        l.append("Normal")
sns.set_style('darkgrid')
sns.countplot(l)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x20ac80c1cc0>
```

Fig. 5 Graph showing the status of pneumonia



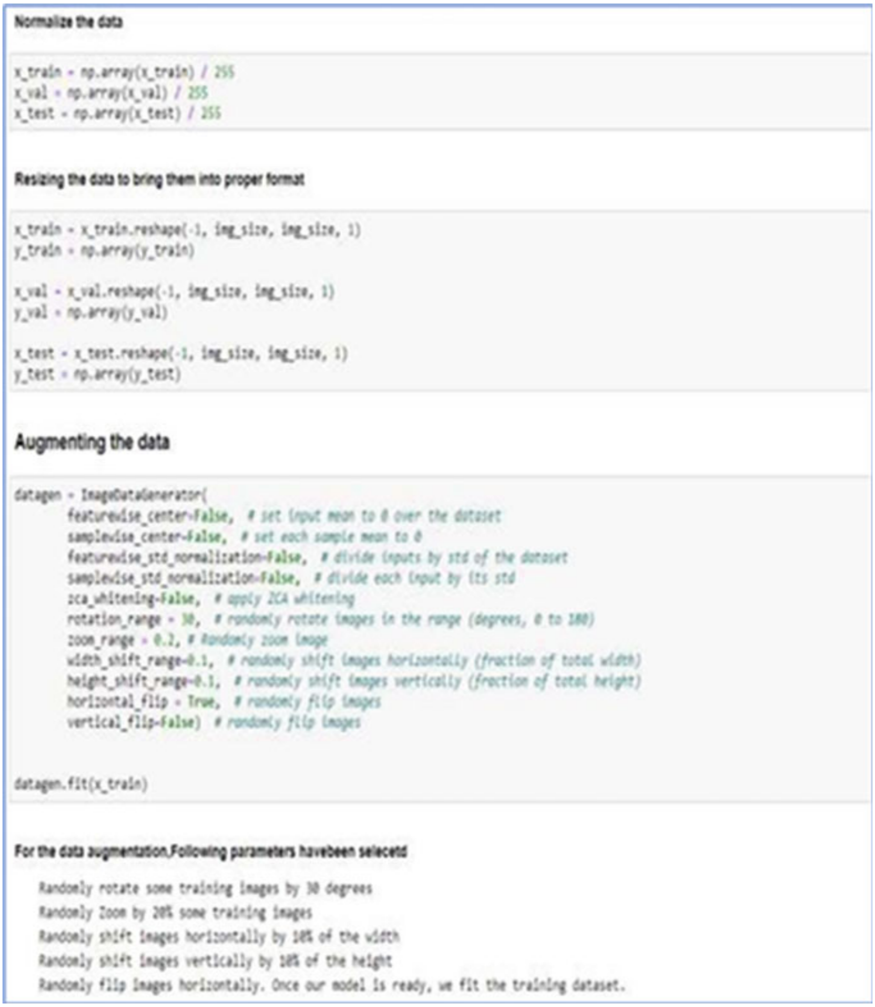
Have a look on the both class of image

```
plt.figure(figsize = (10,5))
plt.imshow(train[0][0], cmap='gray')
plt.title(labels[train[0][1]])

plt.figure(figsize = (10,5))
plt.imshow(train[-1][0], cmap='gray')
plt.title(labels[train[-1][1]])
```

```
Text(0.5, 1.0, 'NORMAL')
```

Fig. 6 Sample images of affected lungs



Normalize the data

```
x_train = np.array(x_train) / 255
x_val = np.array(x_val) / 255
x_test = np.array(x_test) / 255
```

Resizing the data to bring them into proper format

```
x_train = x_train.reshape(-1, img_size, img_size, 1)
y_train = np.array(y_train)

x_val = x_val.reshape(-1, img_size, img_size, 1)
y_val = np.array(y_val)

x_test = x_test.reshape(-1, img_size, img_size, 1)
y_test = np.array(y_test)
```

Augmenting the data

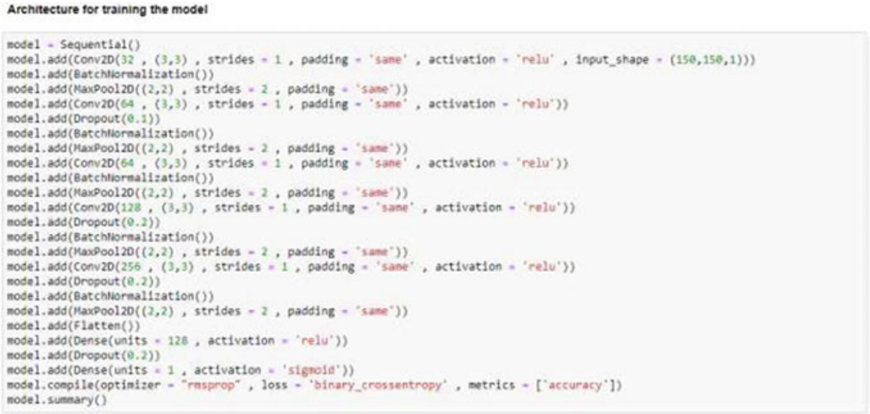
```
datagen = ImageDataGenerator(
        featurewise_center=False,  # set input mean to 0 over the dataset
        samplewise_center=False,  # set each sample mean to 0
        featurewise_std_normalization=False,  # divide inputs by std of the dataset
        samplewise_std_normalization=False,  # divide each input by its std
        zca_whitening=False,  # apply ZCA whitening
        rotation_range = 30,  # randomly rotate images in the range (degrees, 0 to 180)
        zoom_range = 0.2, # Randomly zoom image
        width_shift_range=0.1,  # randomly shift images horizontally (fraction of total width)
        height_shift_range=0.1,  # randomly shift images vertically (fraction of total height)
        horizontal_flip = True,  # randomly flip images
        vertical_flip=False)  # randomly flip images


datagen.fit(x_train)
```

For the data augmentation,Following parameters havebeen selecetd

```
    Randomly rotate some training images by 30 degrees
    Randomly Zoom by 20% some training images
    Randomly shift images horizontally by 10% of the width
    Randomly shift images vertically by 10% of the height
    Randomly flip images horizontally. Once our model is ready, we fit the training dataset.
```

Fig. 7 Sample code showing data normalization, resizing, and augmentation



Architecture for training the model

```
model = Sequential()
model.add(Conv2D(32 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu' , input_shape = (150,150,1)))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(64 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.1))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(64 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(128 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(256 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Flatten())
model.add(Dense(units = 128 , activation = 'relu'))
model.add(Dropout(0.2))
model.add(Dense(units = 1 , activation = 'sigmoid'))
model.compile(optimizer = "rmsprop" , loss = 'binary_crossentropy' , metrics = ['accuracy'])
model.summary()
```

Fig. 8 Code sample to show the process of training the proposed model

7 Code Snippet

All the required class definitions and visualization steps essential for the model are shown. Figure 6 shows images of both normal and pneumonia-affected lungs images. The deviations obtained in these affected images are evident of the percentage of infection through increased number of epochs of CNN. The code for data normalization which is responsible for noise elimination and filtering is shown in Fig. 7. Thus, figure also shows resizing and augmentation code. Figure 8 illustrates the steps followed for training the model. We classified trainable and non-trainable parameters from the input file and identify the percentage of data samples collected. Figure 9 lists the parameters for training the model.

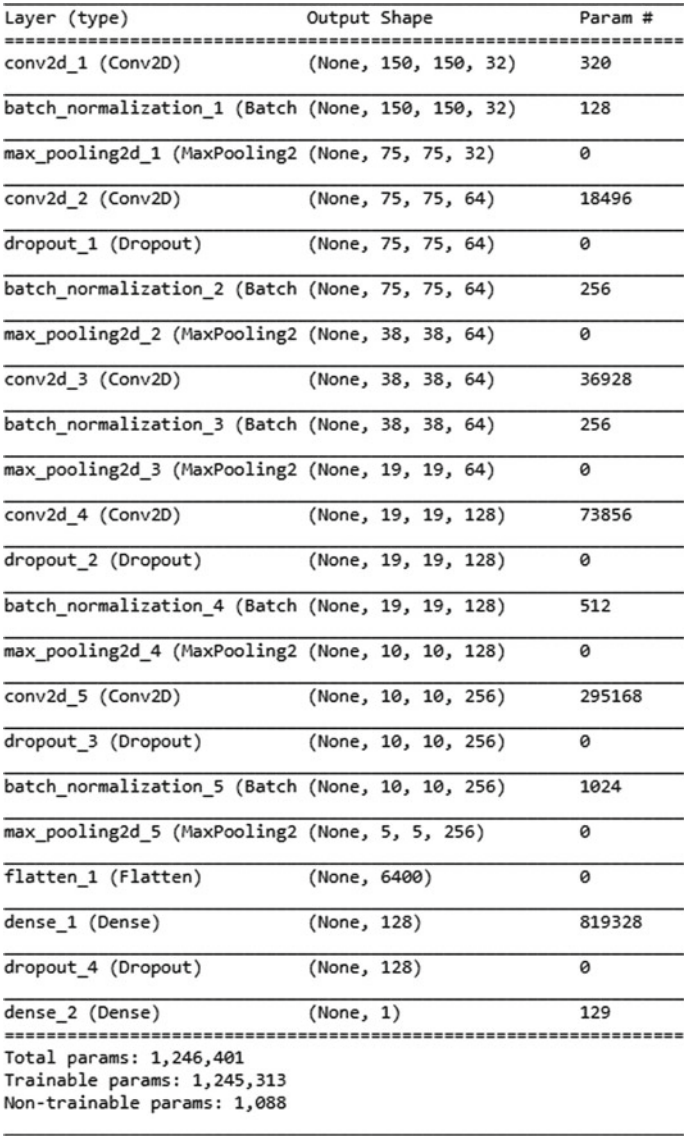
8 Analysis of Model Performance

The proposed work presents the optimal method of analyzing patient's health records in the form of images. Through CNN, the training accuracy and accuracy of the validated results are checked. Figure 10 shows the graph with promising results, and it is evident that the method followed is reliable. Another graph is also shown with very minimal loss rate from the considered datasets.

9 Conclusion and Future Scope

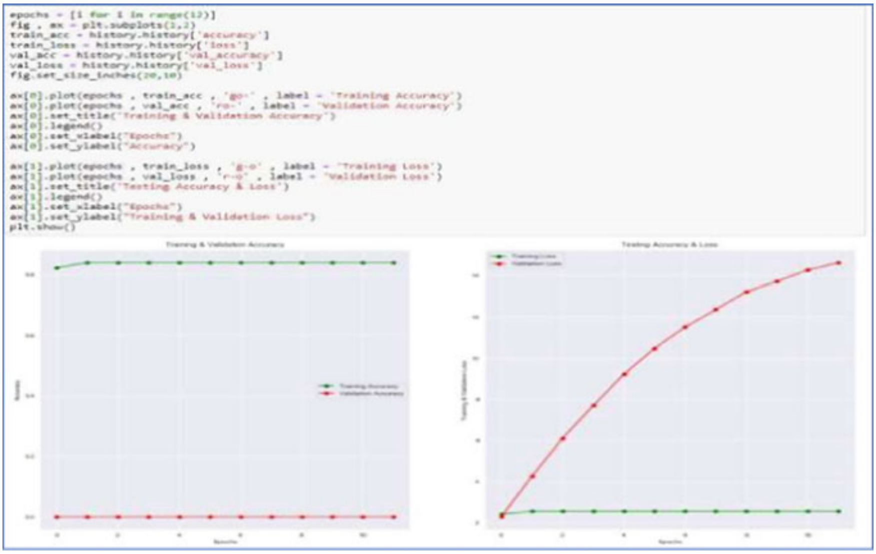
The proposed solution project would be used by the organization/government authorities/medical authorities to reduce the workload of overloaded medical personnel and provide medical facilities to everyone at affordable cost.

Pneumonia is considered to be one of the serious statuses of health which leads to considerable proportion of mortality. This status can be controlled by early diagnosis with some computational techniques. Among various diagnostic procedures, chest X-rays are considered to be a reliable tool for screening and examination. Even though considerable imaging equipments are available, shortage of experts to infer the images is an added challenge. This work facilitates in proposing additional procedure for early detection of the disease through clinical and laboratory evidences of chest X-ray images.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 150, 150, 32)      320
_________________________________________________________________
batch_normalization_1 (Batch (None, 150, 150, 32)      128
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 75, 75, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 75, 75, 64)        18496
_________________________________________________________________
dropout_1 (Dropout)          (None, 75, 75, 64)        0
_________________________________________________________________
batch_normalization_2 (Batch (None, 75, 75, 64)        256
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 38, 38, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 38, 38, 64)        36928
_________________________________________________________________
batch_normalization_3 (Batch (None, 38, 38, 64)        256
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 19, 19, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 19, 19, 128)       73856
_________________________________________________________________
dropout_2 (Dropout)          (None, 19, 19, 128)       0
_________________________________________________________________
batch_normalization_4 (Batch (None, 19, 19, 128)       512
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 10, 10, 128)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 10, 10, 256)       295168
_________________________________________________________________
dropout_3 (Dropout)          (None, 10, 10, 256)       0
_________________________________________________________________
batch_normalization_5 (Batch (None, 10, 10, 256)       1024
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 5, 5, 256)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6400)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               819328
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 129
=================================================================
Total params: 1,246,401
Trainable params: 1,245,313
Non-trainable params: 1,088
_________________________________________________________________
```

Fig. 9 Status of total parameters considered for training the model



```
epochs = [i for i in range(12)]
fig , ax = plt.subplots(1,2)
train_acc = history.history['accuracy']
train_loss = history.history['loss']
val_acc = history.history['val_accuracy']
val_loss = history.history['val_loss']
fig.set_size_inches(20,10)

ax[0].plot(epochs , train_acc , 'go-' , label = 'Training Accuracy')
ax[0].plot(epochs , val_acc , 'ro-' , label = 'Validation Accuracy')
ax[0].set_title('Training & Validation Accuracy')
ax[0].legend()
ax[0].set_xlabel("Epochs")
ax[0].set_ylabel("Accuracy")

ax[1].plot(epochs , train_loss , 'g-o' , label = 'Training Loss')
ax[1].plot(epochs , val_loss , 'r-o' , label = 'Validation Loss')
ax[1].set_title('Testing Accuracy & Loss')
ax[1].legend()
ax[1].set_xlabel("Epochs")
ax[1].set_ylabel("Training & Validation Loss")
plt.show()
```

Fig. 10 Performance of the model developed

References

1. Aydogdu M, Ozyilmaz E, Aksoy H, Gursel G, Ekim N (2010) Mortality prediction in community-acquired pneumonia requiring mechanical ventilation; values of pneumonia and intensive care unit severity scores. Tuberk Toraks 58(1):25–34
2. Berbaum K, Franken EA Jr, Smith WL (1985) The effect of comparison films upon resident interpretation of pediatric chest radiographs. Investig Radiol 20(2):124–128
3. Cherian T, Mulholland EK, Carlin JB, Ostensen H, Amin R, de Campo M, Greenberg D, Lagos R, Lucero M, Madhi SA et al (2005) Standardized interpretation of pediatric chest radiographs for the diagnosis of pneumonia in epidemiological studies. Bull World Health Organ 83(5):353–359
4. Kingma D, Ba JA (2014) A method for stochastic optimization. arXiv preprint arXiv:1412.6980
5. https://data.mendeley.com/datasets/rscbjbr9sj/2
6. Collen MF, Slack WV, Bleich HL (2015) Medical databases and patient record systems. In: Collen M, Ball M (eds) The history of medical informatics in the United States. Health informatics. Springer, London. https://doi.org/10.1007/978-1-4471-6732-7_4
7. Vimalachandran P, Wang H, Zhang Y (2015) Securing electronic medical record and electronic health record systems through an improved access control. In: Yin X, Ho K, Zeng D, Aickelin U, Zhou R, Wang H (eds) Health information science. HIS 2015. Lecture notes in computer science, vol 9085. Springer, Cham. https://doi.org/10.1007/978-3-319-19156-0_3
8. Cahill JE, Gilbert MR, Armstrong TS (2014) Personal health records as portal to the electronic medical record. J Neurooncol 117:1–6. https://doi.org/10.1007/s11060-013-1333-x
9. Kruse CS, Stein A, Thomas H et al (2018) The use of electronic health records to support population health: a systematic review of the literature. J Med Syst 42:214. https://doi.org/10.1007/s10916-018-1075-6
10. Smaradottir BF, Fensli RW (2020) User experiences and satisfaction with an electronic health record system. In: Ahram T, Falcão C (eds) Advances in usability and user experience. AHFE 2019. Advances in intelligent systems and computing, vol 972. Springer, Cham. https://doi.org/10.1007/978-3-030-19135-1_8