CIS-579-001: ARTIFICIAL INTELLIGENCE

WINTER 2024

PROJECT REPORT

SEMESTER - 4

SECTION - 001

PNEUMONIA DETECTION USING CONVOLUTIONAL NEURAL NETWORK (CNN)

TEAM – 9

Ravi Sekhar Gajula

TABLE OF CONTENTS

ABSTRACT

Humans who have pneumonia, a potentially fatal bacterial condition affecting one or both lungs, are frequently infected with the Streptococcus pneumonia bacteria. According to the World Health Organization, pneumonia is to blame for one in three deaths in India (WHO). Radiotherapists with advanced training are required to evaluate chest X-rays used to diagnose pneumonia.

Creating an automatic system for diagnosing pneumonia might help treat the illness quickly, especially in distant places. Convolutional Neural Networks (CNN) have attracted a lot of attention for illness categorization as a result of deep learning algorithms' success in evaluating medical imagery. Moreover, pre-trained CNN models' features from huge data sets are quite helpful in picture classification applications. In this study, we evaluate how well pre-trained CNN models perform as feature-extractors used in conjunction with various classifiers to identify abnormal and normal chest X-rays.

We use analysis to choose the best CNN model for the job. According to statistical findings, pretrained CNN models combined with supervised classifier algorithms can be highly helpful for assessing chest X-ray pictures, particularly when trying to find pneumonia.

PROBLEM STATEMENT

The objective of this project is to develop a convolutional neural network (CNN) model capable of effectively distinguishing between normal, viral pneumonia, and bacterial pneumonia by using chest X-ray images. By using deep learning techniques, this model aims to provide results with a reliable tool for early pneumonia detection, facilitating prompt intervention and ultimately improving patient diagnosis.

INTRODUCTION

Pneumonia, an inflammation of the lungs primarily affecting the alveoli, presents with symptoms such as cough, chest pain, fever, and breathing difficulties. It can vary in severity and is commonly caused by viral or bacterial infections, though other factors like autoimmune diseases or certain medications can also contribute. Risk factors include conditions like Cystic Fibrosis, COPD, asthma, diabetes, and a weakened immune system, among others. Diagnosis typically involves physical examination, symptom assessment, blood tests, sputum cultures, and chest X-rays.

Globally, pneumonia claimed over a million children's lives in 2018, remaining a significant threat if not promptly treated. Radiography, CT scans, or MRI are commonly used to detect pneumonia, with chest X-rays being a primary method. These X-rays produce images where hard tissues appear bright, while soft tissues appear darker. Pneumonia manifests as brightness due to fluid filling the air sacs in the lungs. However, other conditions like cancer, enlarged blood vessels, or heart issues may also present as bright areas on the X-ray, complicating diagnosis.

In rural areas, timely pneumonia detection can be challenging, but recent research has shown promise in utilizing advanced techniques like CNN models to enhance detection accuracy. By training and evaluating these models using various classifiers, researchers have aimed to improve pneumonia diagnosis, even in settings with limited resources. However, the early stages of the disease can still be difficult to discern, potentially leading to misdiagnosis.

pneumonia is a serious lung condition with varied causes and risk factors, often diagnosed through a combination of physical exams, symptoms, and imaging like chest X-rays. Despite its lethality, advancements in technology, such as CNN models, offer hope for improved detection, particularly in underserved rural areas, though challenges remain in accurately identifying the disease's onset.

DATA DESCRIPTION

There are sub folders for each image category (Pneumonia/Normal) in the data set, which is divided into 2 folders (train, test). With 2 categories (Pneumonia/Normal), there are 5,841 X-Ray images in JPEG format. The size of our data is 1.2 GB. Anterior-posterior chest X-rays of pediatric patients aged one to five from the Guangzhou Women and Children's Medical Center, Guangzhou, were chosen from retrospective cohorts. The normal clinical treatment provided to patients included all chest X-ray imaging. All chest radiographs were originally inspected for quality control prior to the interpretation of the x-ray pictures, and any scans that were of poor quality or were impossible to read were removed. Before the AI system could be trained, the diagnosis for the photos were rated by two experienced doctors.

The data can be accessed from this link:
https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

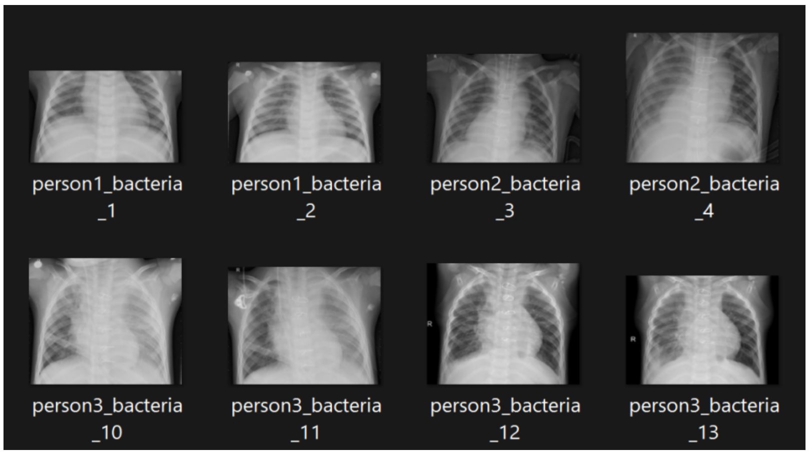


Fig 1. X-ray images of Pneumonia disease

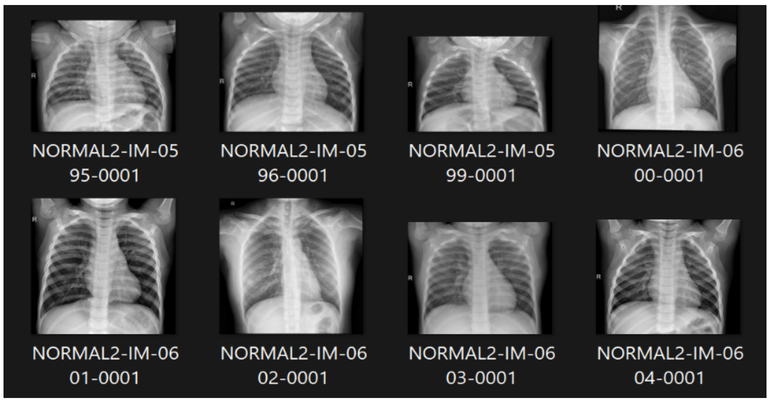


Fig 2. X-ray images of Normal person

The quality and amount of the data is one of the key problems in gathering data for a machine learning project. The project's effectiveness in the case of pneumonia identification with CNN depends on having a sizable and varied dataset of chest X-ray pictures that faithfully depict both healthy and sick lungs.Due to privacy issues, ethical considerations, and legal limitations, it can be difficult to gather a large enough dataset for various medical imaging applications. Medical picture labeling is also complicated, time-consuming, and sometimes expensive.

The variation in image resolution and quality poses another difficulty in gathering data for medical imaging jobs. To enhance the performance of the model in the pneumonia detection project, the scientists had to downsize, normalize, and reduce the pixel intensity of the photos.The success of a machine learning project depends on getting high-quality data, which necessitates careful evaluation of the data sources, labeling techniques, and preprocessing methods.

ALGORITHM USED

CONVOLUTIONAL NEURAL NETWORK:

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning technique that can take an input image, assign different elements and objects in the image importance (learnable weights and biases), and be able to differentiate between them. In comparison to other classification methods, a ConvNet requires significantly less pre-processing. Contrary to earlier approaches, where filters must be hand-engineered, ConvNets are capable of learning these filters and their attributes. The architecture of a ConvNet is similar to the connecting network of neurons in the human brain and was influenced by the organization of the visual cortex. Individual neurons only respond to stimuli in this restricted region of the visual field, known as the Receptive Field. There are several overlapping fields like this that make up the total visual field.

METHODOLOGY

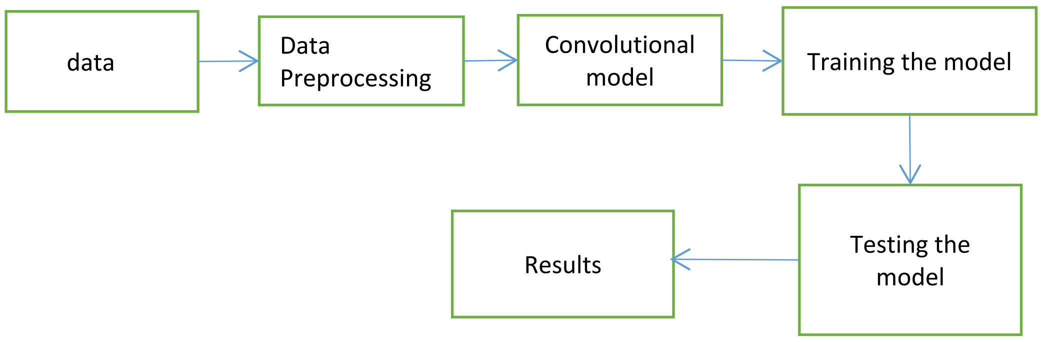


Fig 3. Flowchart of Project Implementation

Convolution block diagram:

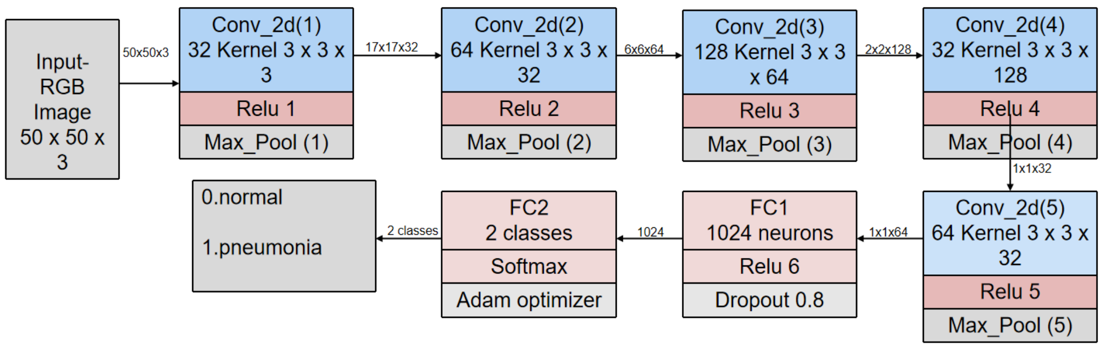
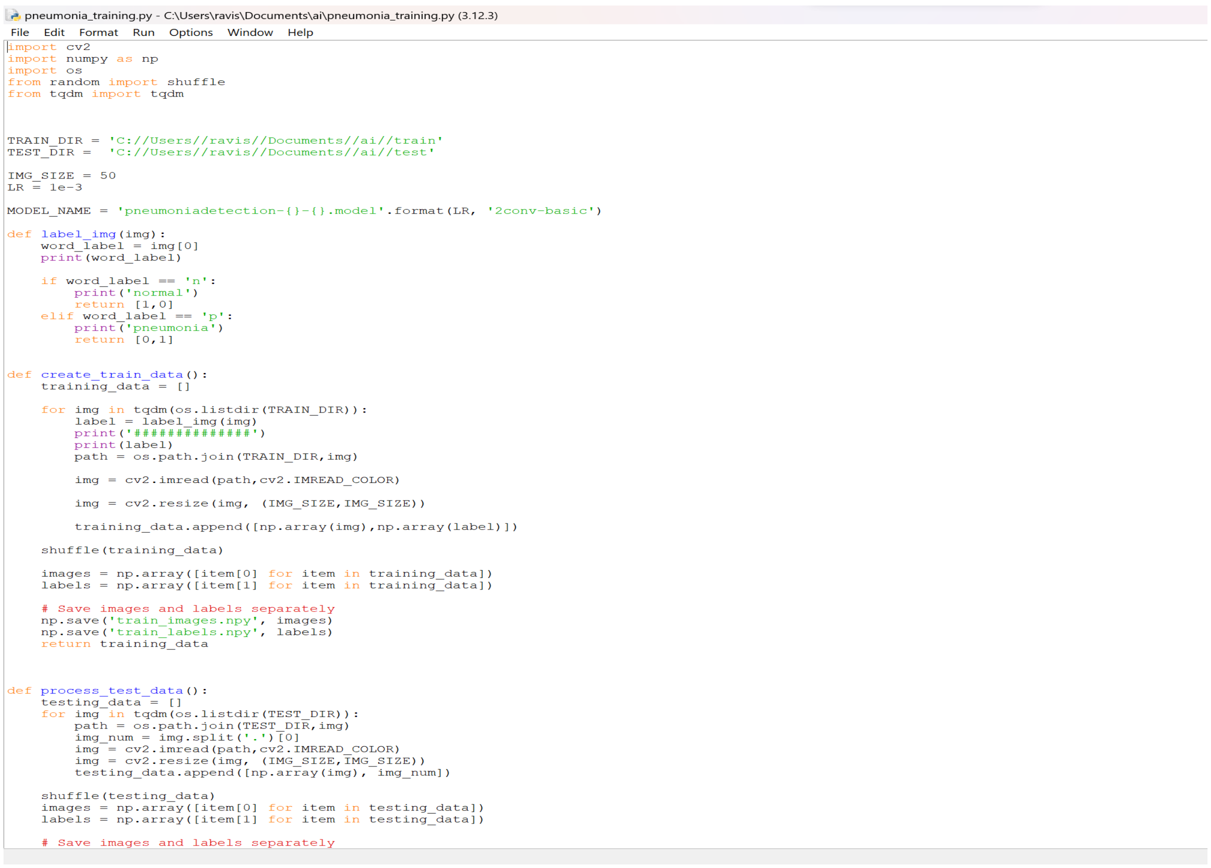


Fig 4. Convolution Model developed

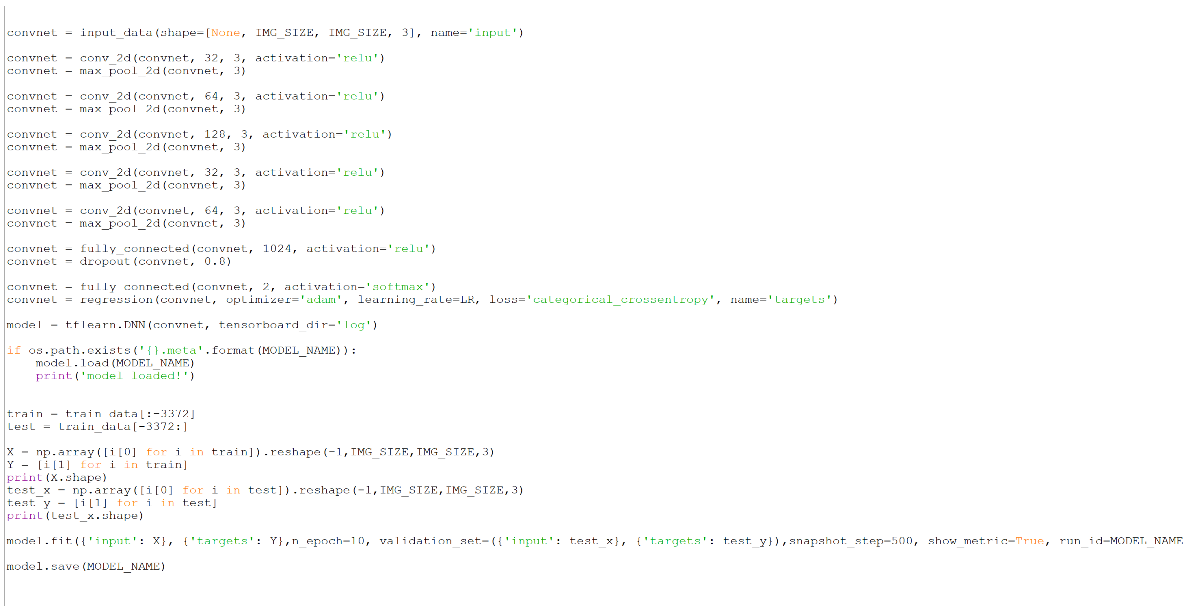
EXPERIMENTATION

The experimental setup detailed the construction of a convolutional neural network (CNN) for the purpose of pneumonia detection from X-ray images. The CNN architecture consisted of several layers: convolutional layers, activation layers (ReLU), max pooling layers, and fully connected layers (FC1 and FC2). The input layer accepted 50x50x3 dimensional X-ray images, which were sequentially processed through the CNN. Convolutional layers applied learnable filters to extract features, followed by ReLU activation to introduce non-linearity. Max pooling layers downsampled the feature maps to reduce dimensionality. The flattened output from these layers fed into fully connected layers, with FC1 containing 1024 neurons and utilizing a dropout ratio of 0.8 for regularization. The Adam Optimizer was employed to optimize model parameters during training. Finally, the softmax activation function was applied in the last fully connected layer (FC2) to generate class probabilities for pneumonia detection. This experimental design aimed to develop an efficient and accurate model for pneumonia detection from X-ray images, utilizing techniques such as dropout for regularization and softmax for classification. The overarching goal was to enhance pneumonia detection accuracy, particularly in medical emergencies in rural areas where access to specialized healthcare may be limited. Through rigorous experimentation and optimization of the CNN architecture, the study sought to contribute to advancements in medical imaging technology, ultimately improving patient outcomes and reducing mortality rates associated with pneumonia.


pneumonia_training.py - C:\Users\ravis\Documents\ai\pneumonia_training.py (3.12.3)
File Edit Format Run Options Window Help

```python
import cv2
import numpy as np
import os
from random import shuffle
from tqdm import tqdm


TRAIN_DIR = 'C://Users//ravis//Documents//ai//train'
TEST_DIR =  'C://Users//ravis//Documents//ai//test'

IMG_SIZE = 50
LR = 1e-3

MODEL_NAME = 'pneumoniadetection-{}-{}.model'.format(LR, '2conv-basic')

def label_img(img):
    word_label = img[0]
    print(word_label)

    if word_label == 'n':
        print('normal')
        return [1,0]
    elif word_label == 'p':
        print('pneumonia')
        return [0,1]


def create_train_data():
    training_data = []

    for img in tqdm(os.listdir(TRAIN_DIR)):
        label = label_img(img)
        print('##############')
        print(label)
        path = os.path.join(TRAIN_DIR,img)

        img = cv2.imread(path,cv2.IMREAD_COLOR)

        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))

        training_data.append([np.array(img),np.array(label)])

    shuffle(training_data)

    images = np.array([item[0] for item in training_data])
    labels = np.array([item[1] for item in training_data])

    # Save images and labels separately
    np.save('train_images.npy', images)
    np.save('train_labels.npy', labels)
    return training_data


def process_test_data():
    testing_data = []
    for img in tqdm(os.listdir(TEST_DIR)):
        path = os.path.join(TEST_DIR,img)
        img_num = img.split('.')[0]
        img = cv2.imread(path,cv2.IMREAD_COLOR)
        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
        testing_data.append([np.array(img), img_num])

    shuffle(testing_data)
    images = np.array([item[0] for item in testing_data])
    labels = np.array([item[1] for item in testing_data])

    # Save images and labels separately
```

Fig 5.code snippet for training the mode



```
convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 3], name='input')

convnet = conv_2d(convnet, 32, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

convnet = conv_2d(convnet, 64, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

convnet = conv_2d(convnet, 128, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

convnet = conv_2d(convnet, 32, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

convnet = conv_2d(convnet, 64, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)

convnet = fully_connected(convnet, 2, activation='softmax')
convnet = regression(convnet, optimizer='adam', learning_rate=LR, loss='categorical_crossentropy', name='targets')

model = tflearn.DNN(convnet, tensorboard_dir='log')

if os.path.exists('{}.meta'.format(MODEL_NAME)):
    model.load(MODEL_NAME)
    print('model loaded!')


train = train_data[:-3372]
test = train_data[-3372:]

X = np.array([i[0] for i in train]).reshape(-1,IMG_SIZE,IMG_SIZE,3)
Y = [i[1] for i in train]
print(X.shape)
test_x = np.array([i[0] for i in test]).reshape(-1,IMG_SIZE,IMG_SIZE,3)
test_y = [i[1] for i in test]
print(test_x.shape)

model.fit({'input': X}, {'targets': Y},n_epoch=10, validation_set=({'input': test_x}, {'targets': test_y}),snapshot_step=500, show_metric=True, run_id=MODEL_NAME

model.save(MODEL_NAME)
```

Fig6.code snippet

We have used training code from GeeksforGeeks which is mentioned below and we have used our own testing code.

https://www.geeksforgeeks.org/image-classifier-using-cnn/

TOOLS USED

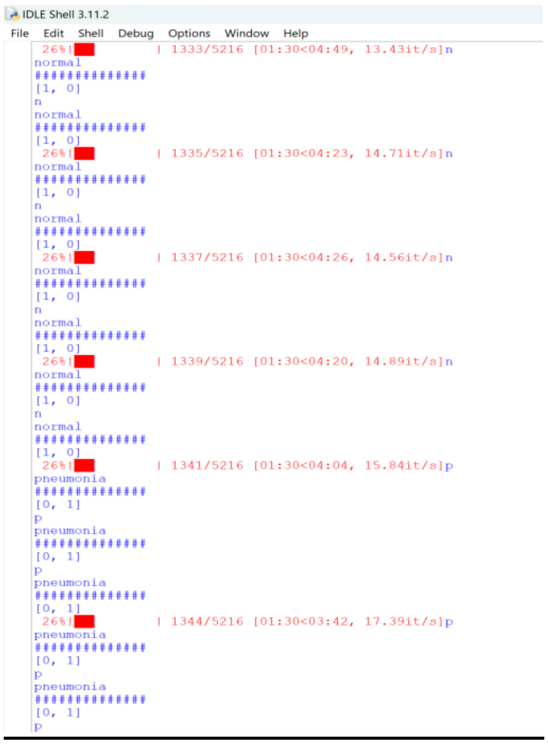
Hardware:

- Min 4GB Ram.
- Processor better than i5.

Software:

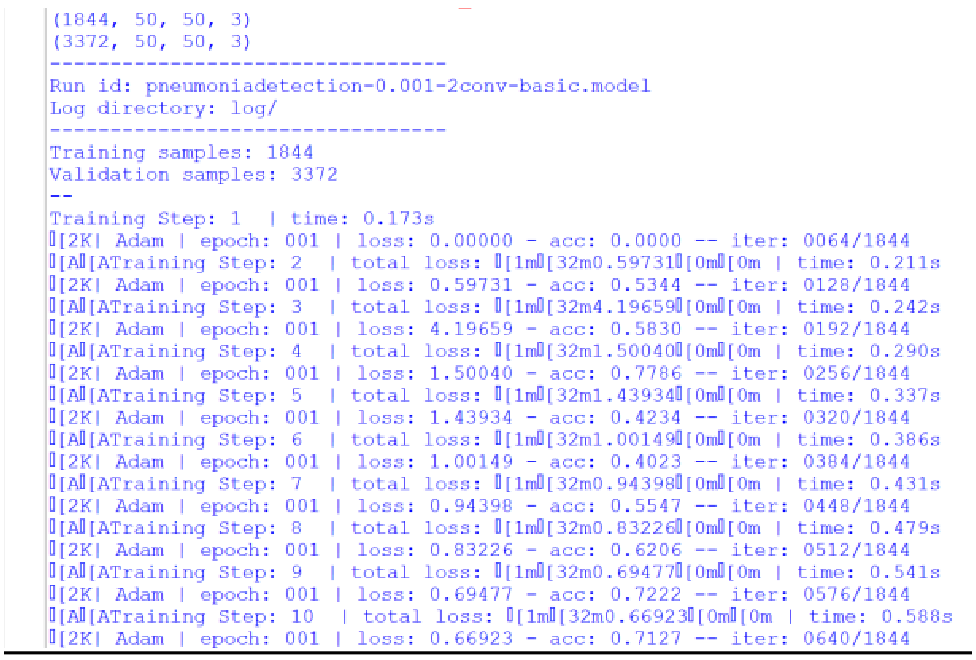
- We used python IDE to build the model.
- Some of the python libraries that are used in this project are tensorflow, opencv, os, numpy, tkinter.

RESULTS:



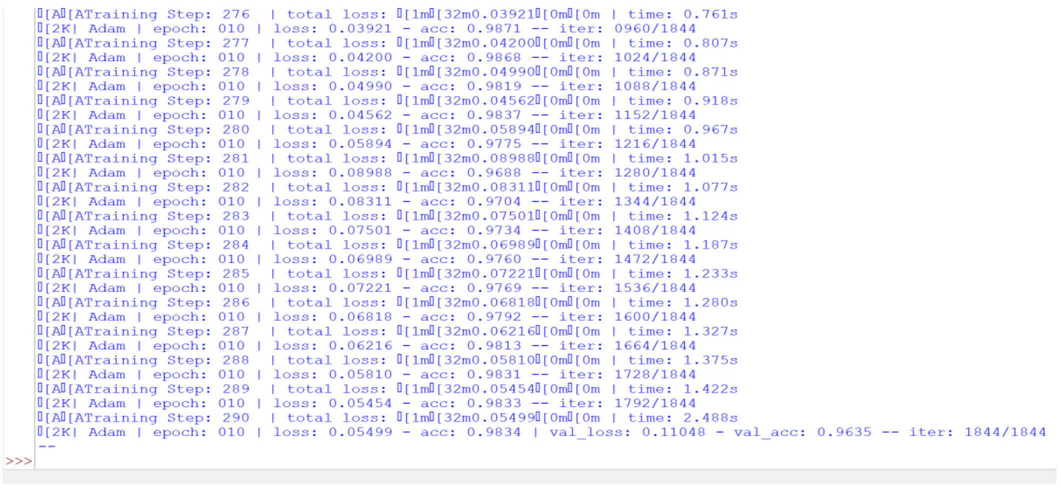
```
 26%|██         | 1333/5216 [01:30<04:49, 13.43it/s]n
normal
##############
[1, 0]
n
normal
##############
[1, 0]
 26%|██         | 1335/5216 [01:30<04:23, 14.71it/s]n
normal
##############
[1, 0]
n
normal
##############
[1, 0]
 26%|██         | 1337/5216 [01:30<04:26, 14.56it/s]n
normal
##############
[1, 0]
n
normal
##############
[1, 0]
 26%|██         | 1339/5216 [01:30<04:20, 14.89it/s]n
normal
##############
[1, 0]
n
normal
##############
[1, 0]
 26%|██         | 1341/5216 [01:30<04:04, 15.84it/s]p
pneumonia
##############
[0, 1]
p
pneumonia
##############
[0, 1]
p
pneumonia
##############
[0, 1]
 26%|██         | 1344/5216 [01:30<03:42, 17.39it/s]p
pneumonia
##############
[0, 1]
p
pneumonia
##############
[0, 1]
p
```

Fig 7. Training the images for the model



```
(1844, 50, 50, 3)
(3372, 50, 50, 3)
---------------------------------
Run id: pneumoniadetection-0.001-2conv-basic.model
Log directory: log/
---------------------------------
Training samples: 1844
Validation samples: 3372
--
Training Step: 1  | time: 0.173s
▯[2K| Adam | epoch: 001 | loss: 0.00000 - acc: 0.0000 -- iter: 0064/1844
▯[A▯[ATraining Step: 2  | total loss: ▯[1m▯[32m0.59731▯[0m▯[0m | time: 0.211s
▯[2K| Adam | epoch: 001 | loss: 0.59731 - acc: 0.5344 -- iter: 0128/1844
▯[A▯[ATraining Step: 3  | total loss: ▯[1m▯[32m4.19659▯[0m▯[0m | time: 0.242s
▯[2K| Adam | epoch: 001 | loss: 4.19659 - acc: 0.5830 -- iter: 0192/1844
▯[A▯[ATraining Step: 4  | total loss: ▯[1m▯[32m1.50040▯[0m▯[0m | time: 0.290s
▯[2K| Adam | epoch: 001 | loss: 1.50040 - acc: 0.7786 -- iter: 0256/1844
▯[A▯[ATraining Step: 5  | total loss: ▯[1m▯[32m1.43934▯[0m▯[0m | time: 0.337s
▯[2K| Adam | epoch: 001 | loss: 1.43934 - acc: 0.4234 -- iter: 0320/1844
▯[A▯[ATraining Step: 6  | total loss: ▯[1m▯[32m1.00149▯[0m▯[0m | time: 0.386s
▯[2K| Adam | epoch: 001 | loss: 1.00149 - acc: 0.4023 -- iter: 0384/1844
▯[A▯[ATraining Step: 7  | total loss: ▯[1m▯[32m0.94398▯[0m▯[0m | time: 0.431s
▯[2K| Adam | epoch: 001 | loss: 0.94398 - acc: 0.5547 -- iter: 0448/1844
▯[A▯[ATraining Step: 8  | total loss: ▯[1m▯[32m0.83226▯[0m▯[0m | time: 0.479s
▯[2K| Adam | epoch: 001 | loss: 0.83226 - acc: 0.6206 -- iter: 0512/1844
▯[A▯[ATraining Step: 9  | total loss: ▯[1m▯[32m0.69477▯[0m▯[0m | time: 0.541s
▯[2K| Adam | epoch: 001 | loss: 0.69477 - acc: 0.7222 -- iter: 0576/1844
▯[A▯[ATraining Step: 10  | total loss: ▯[1m▯[32m0.66923▯[0m▯[0m | time: 0.588s
▯[2K| Adam | epoch: 001 | loss: 0.66923 - acc: 0.7127 -- iter: 0640/1844
```

Fig 8. Model Validation



```
▯[A▯[ATraining Step: 276  | total loss: ▯[1m▯[32m0.03921▯[0m▯[0m | time: 0.761s
▯[2K| Adam | epoch: 010 | loss: 0.03921 - acc: 0.9871 -- iter: 0960/1844
▯[A▯[ATraining Step: 277  | total loss: ▯[1m▯[32m0.04200▯[0m▯[0m | time: 0.807s
▯[2K| Adam | epoch: 010 | loss: 0.04200 - acc: 0.9868 -- iter: 1024/1844
▯[A▯[ATraining Step: 278  | total loss: ▯[1m▯[32m0.04990▯[0m▯[0m | time: 0.871s
▯[2K| Adam | epoch: 010 | loss: 0.04990 - acc: 0.9819 -- iter: 1088/1844
▯[A▯[ATraining Step: 279  | total loss: ▯[1m▯[32m0.04562▯[0m▯[0m | time: 0.918s
▯[2K| Adam | epoch: 010 | loss: 0.04562 - acc: 0.9837 -- iter: 1152/1844
▯[A▯[ATraining Step: 280  | total loss: ▯[1m▯[32m0.05894▯[0m▯[0m | time: 0.967s
▯[2K| Adam | epoch: 010 | loss: 0.05894 - acc: 0.9775 -- iter: 1216/1844
▯[A▯[ATraining Step: 281  | total loss: ▯[1m▯[32m0.08988▯[0m▯[0m | time: 1.015s
▯[2K| Adam | epoch: 010 | loss: 0.08988 - acc: 0.9688 -- iter: 1280/1844
▯[A▯[ATraining Step: 282  | total loss: ▯[1m▯[32m0.08311▯[0m▯[0m | time: 1.077s
▯[2K| Adam | epoch: 010 | loss: 0.08311 - acc: 0.9704 -- iter: 1344/1844
▯[A▯[ATraining Step: 283  | total loss: ▯[1m▯[32m0.07501▯[0m▯[0m | time: 1.124s
▯[2K| Adam | epoch: 010 | loss: 0.07501 - acc: 0.9734 -- iter: 1408/1844
▯[A▯[ATraining Step: 284  | total loss: ▯[1m▯[32m0.06989▯[0m▯[0m | time: 1.187s
▯[2K| Adam | epoch: 010 | loss: 0.06989 - acc: 0.9760 -- iter: 1472/1844
▯[A▯[ATraining Step: 285  | total loss: ▯[1m▯[32m0.07221▯[0m▯[0m | time: 1.233s
▯[2K| Adam | epoch: 010 | loss: 0.07221 - acc: 0.9769 -- iter: 1536/1844
▯[A▯[ATraining Step: 286  | total loss: ▯[1m▯[32m0.06818▯[0m▯[0m | time: 1.280s
▯[2K| Adam | epoch: 010 | loss: 0.06818 - acc: 0.9792 -- iter: 1600/1844
▯[A▯[ATraining Step: 287  | total loss: ▯[1m▯[32m0.06216▯[0m▯[0m | time: 1.327s
▯[2K| Adam | epoch: 010 | loss: 0.06216 - acc: 0.9813 -- iter: 1664/1844
▯[A▯[ATraining Step: 288  | total loss: ▯[1m▯[32m0.05810▯[0m▯[0m | time: 1.375s
▯[2K| Adam | epoch: 010 | loss: 0.05810 - acc: 0.9831 -- iter: 1728/1844
▯[A▯[ATraining Step: 289  | total loss: ▯[1m▯[32m0.05454▯[0m▯[0m | time: 1.422s
▯[2K| Adam | epoch: 010 | loss: 0.05454 - acc: 0.9833 -- iter: 1792/1844
▯[A▯[ATraining Step: 290  | total loss: ▯[1m▯[32m0.05499▯[0m▯[0m | time: 2.488s
▯[2K| Adam | epoch: 010 | loss: 0.05499 - acc: 0.9834 | val_loss: 0.11048 - val_acc: 0.9635 -- iter: 1844/1844
--
>>>
```

Fig 9. Model trained with 10 epochs

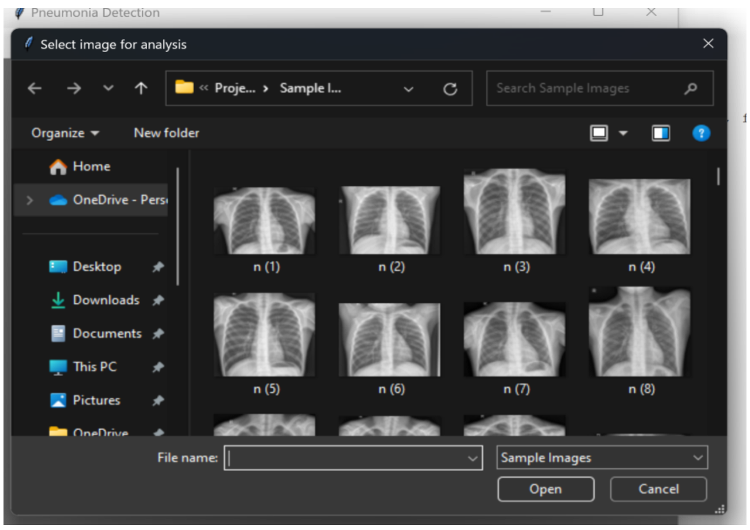
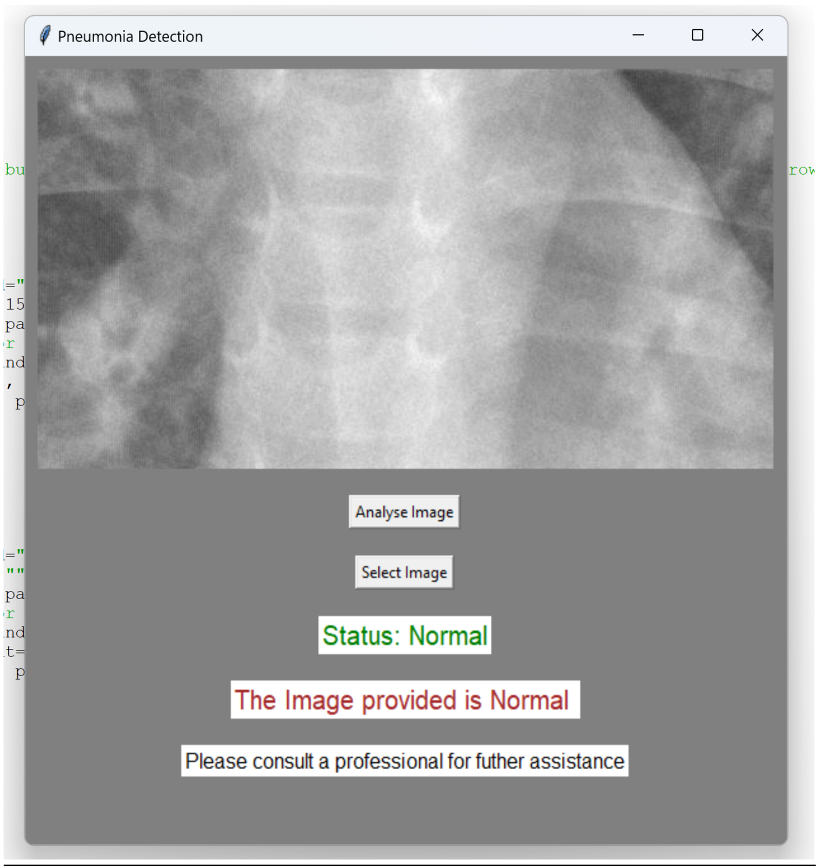


Fig 10. Images used for analysis

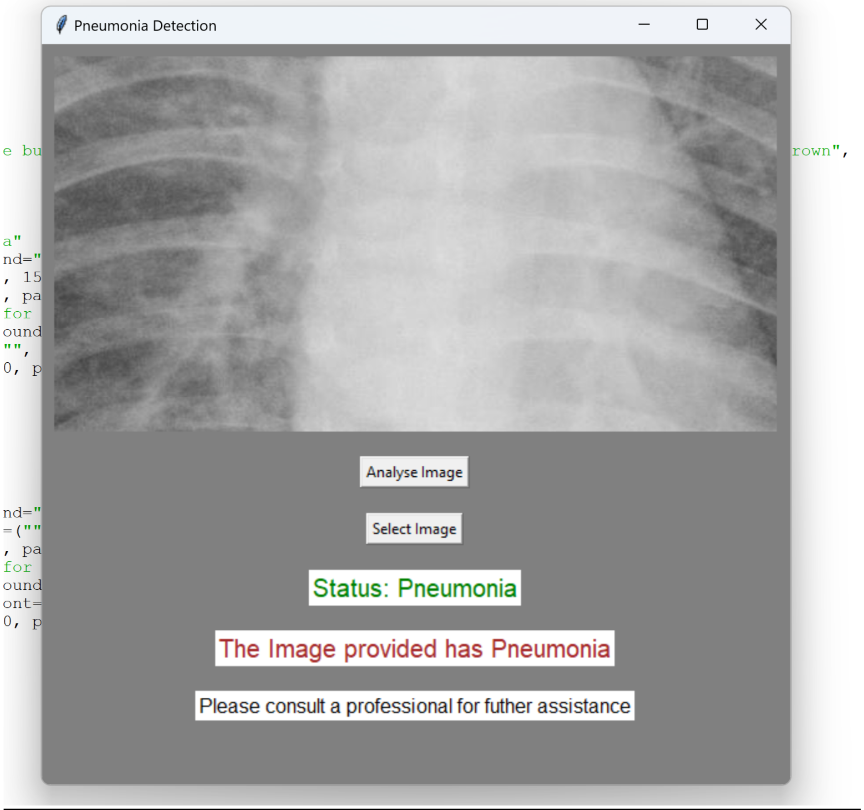


Status: Normal

The Image provided is Normal

Please consult a professional for futher assistance

Fig 11. Image tested as normal



Status: Pneumonia

The Image provided has Pneumonia

Please consult a professional for futher assistance

Fig 12. Image with Pneumonia

CONCLUSION

Our research focused on pneumonia detection with modern technologies to overcome the gap in thoracic illness diagnosis in areas lacking trained radiologists. Our study found that using convolutional neural networks (CNNs), it was possible to diagnose pneumonia from chest X-ray pictures with an astounding 96% accuracy rate. Our CNN model's excellent sensitivity and specificity point to its potential as a trustworthy diagnostic aid, particularly in places with limited access to professional medical care. Creating an intuitive graphical user interface improves accessibility even more and makes it possible to analyse and analyse pictures quickly, which helps with prompt illness treatment. This study emphasises how important it is for technology to advance healthcare delivery, especially in isolated areas with limited access to healthcare services.

FUTURE PLANS

We can explore various applications of our CNN model beyond pneumonia detection. By adapting our approach, we can extend its use to diagnose other lung diseases like tuberculosis or lung cancer, and even expand into medical image classification tasks involving mammograms, MRI scans, or CT scans. Additionally, we can utilize the model for identifying specific features within medical photos, such as tumors or nodules, through further customization. Furthermore, employing transfer learning techniques allows us to leverage the pre-trained model for other computer vision tasks, potentially enhancing performance while overcoming limitations such as the need for extensive annotated datasets and concerns regarding bias and overfitting. Although a deeper exploration of these applications was limited by time constraints, addressing challenges like dataset size and potential biases remains crucial for advancing the application of CNNs in medical imaging.

SUMMARY

The significance of employing CNNs to diagnose pneumonia lies in its potential impact on the medical industry, particularly in addressing the widespread prevalence and severity of the illness globally. Pneumonia affects millions worldwide, and accurate and timely diagnosis is paramount for improving patient outcomes and potentially saving lives. Automating the diagnostic process through CNNs offers a faster and more precise approach, highlighting the promise of technology in revolutionizing healthcare practices.

The remarkable efficacy of CNNs in diagnosing pneumonia stands out as a notable outcome of this experiment. Despite being primarily designed for image recognition tasks, the ability of convolutional neural networks to discern between healthy and diseased lungs is truly impressive, underscoring their versatility and utility in medical imaging applications.

However, the challenges encountered in developing a reliable diagnostic model for pneumonia cannot be overlooked. Securing a large and diverse dataset of X-ray images with accurately labeled classifications poses a significant hurdle. Additionally, ensuring the model remains unbiased requires careful consideration of the dataset's representation across various demographics affected by pneumonia. Despite these challenges, the model's precise identification of pneumonia in X-ray images is a source of pride, with its high accuracy rates offering the potential to significantly enhance patient outcomes through early detection.

In this project, a CNN model was specifically trained to recognize pneumonia in chest X-ray images. Leveraging images from the dataset, the model successfully classified lung images as healthy or infected, achieving impressive levels of accuracy. This development marks a promising advancement towards automating the diagnosis of pneumonia, offering a potential solution to the challenges posed by the illness's widespread impact and the limitations of traditional diagnostic methods.

REFERENCES

1. Dimpy Varshni, Kartik Thakral, Lucky Agarwal, Rahul Nijhawan, Ankush Mittal. Pneumonia Detection Using CNN based Feature Extraction. IEEE 2019.

2. Sammy V. Militante, Brandon G. Sibbaluca. Pneumonia Detection Using Convolutional Neural Networks. IJSTR 2020.

3. V. Sirish Kaushik, Anand Nayyar, Gaurav Kataria, Rachna Jain. Pneumonia Detection Using CNNs. Research gate 2020.

4. **https://www.geeksforgeeks.org/image-classifier-using-cnn/**