Project Report

On

Pneumonia detection using Convolutional Neural Network

Towards partial fulfillment of the requirement for the award of degree of

Bachelor of Computer Applications

from

Babu Banarasi Das University Lucknow

Developed and submitted by

Pawan Kumar

Under Guidance of

Mr. Sarfaraz Alam

Academic session (2019-20)

School of Computer Application

Babu Banarasi Das University
Lucknow

CERTIFICATE

This is to certify that Project Report entitled

Pneumonia detection using CNN

being submitted by

Pawan Kumar

towards the partial fulfillment of the requirement

for the award of the degree of

Bachelor of Computer Applications

to

Babu Banarasi Das University Lucknow

Prabhash Ch. Pathak

Head (School of computer application)

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **FreeCodeCamp** and **FastAi** for their courses and library, and friendly forum as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my friends & members of **KG Brothers E commerce pvt ltd** for their kind cooperation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to Professors of my college for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Thank you

DECLARATION

I **PAWAN KUMAR** hereby declare that this Project Report entitled **PNEUMONIA DETECTION USING CONVOLUTIONAL NEURAL NETWORK** , submitted by me**,** **As final semester project** is my own and has not been submitted to any other University or Institute or published earlier.

Signature of Student :

Pawan Kumar
BCA
VI Semester
(1170211200)
Date: 18-May-2020

ABSTRACT

Artificial Intelligence is already playing a vital role in healthcare with use into various fields like medical imaging and surgery to robots making the crucial and time taking tasks perform such actions. It is now widely used in detecting the diseases and discovering the new drugs.

Just like current applications into healthcare and medical sub-fields, there is a huge scope of AI in this sector to contribute to making the medical care and treatment process more expedient and trouble-free while improving the predictions given by machine learning models with more improved accuracy level helping doctors get quicker decisions.

Using the high-quality machine learning training data enabled systems can easier detect such maladies helping people die due to such critical maladies. Healthcare companies and medical -care organizations apply the machine learning technology into their subfields to diagnose the diseases initially and provide the timely treatment.
In this project, we used the fast and in memory computation framework 'Pytorch Python ' to train our model on real life data., and perform prediction of X-Ray images.

The primary aim is to provide a method for detecting Pneumonic cases from reports in order to save time for medical workers.

This page intentionally left blank

TABLE OF CONTENT

Introduction of the Project

1 Introduction

The risk of pneumonia is immense for many, especially in developing nations where billions face energy poverty and rely on polluting forms of energy. The WHO estimates that over 4 million premature deaths occur annually from household air pollution-related diseases including pneumonia. **Over 150 million people get infected with pneumonia on an annual basis** especially children under 5 years old . In such regions, the problem can be further aggravated due to the dearth of medical resources and personnel. For example, in Africa's 57 nations, a gap of 2.3million doctors and nurses exists. For these populations, accurate and fast diagnosis means everything. **It can guarantee timely access to treatment and save much needed time and money for those already experiencing poverty.**

In recent times, CNN-motivated deep learning algorithms have become the standard choice for medical image classifications. For example U-Net, SegNet , and Car-diacNet are some of the prominent architectures for medical image examination.

Algorithm we develop in this project will provide reliable **Detection of Pneumonia** to users and can be used as base for other related project
implementing or using our model is as simple as calling our **REST API**. There are no servers to setup or settings to configure.

Objectives and scope

2.1 Objective

- To train a convolutional neural network that can identify pneumonia cases with reasonable accuracy.
- Establishing an automatic retraining of the model based on feedback collected.
- Provide an easy to use interface so that any non-technical individual can use.
- Provide an API so that existing or new systems can use it with ease.
- Develop a documentation for the project

2.2 Scope

- The scope of the project is to provide a user friendly software that any person can use to detect pneumonia.
- Create an anonymous dataset based on users feedback to improve upon accuracy and to extract more features from data.
- This model can be embedded into an x-Ray imaging machine to automatically classify chest x-rays on the basis of the presence of pneumonia.

PROBLEM STATEMENT

3.1Problem statement

- Create an Artificial intelligence software that can identify pneumonia with standard accuracy.
- Create an easy to use interface for the software.
- Evolve it as a self improving software.

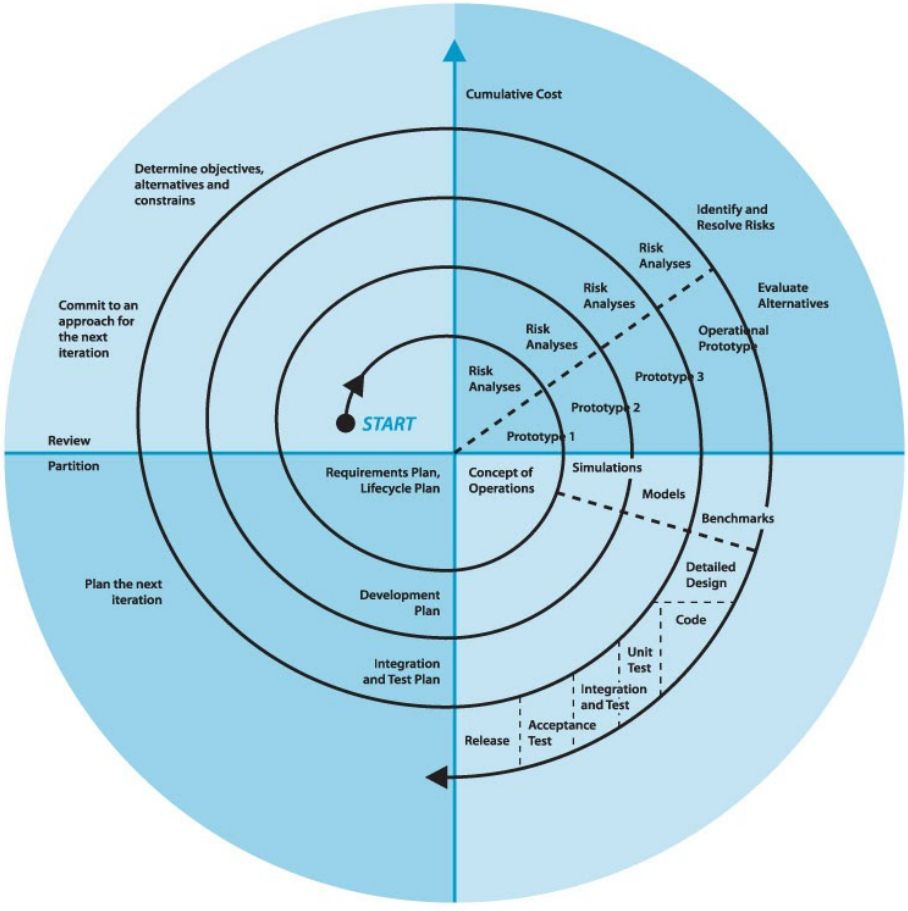
System Analysis

&

Software Requirement Specification

4.1 Spiral design

Spiral model is a combination of sequential and prototype model. This model is best used for large projects which involves continuous enhancements. There are specific activities which are done in one iteration (spiral) where the output is a small prototype of the large software. The same activities are then repeated for all the spirals till the
the entire software is built.



4.2 Modules of the project

4.2.1 Backend

- Data set collection
- Cleaning dataset
- Data augmentations
- Training existing CNN model
- Testing
- Creating REST API

4.2.2 Frontend

- Creating Image upload portal
- Result display portal
- Feedback portal
- Documentation

4.2.3 Deployment

- Software and Hardware requirements calculation (run-time).
- Project optimisation (for production).
- Pipeline setup for automation. (automatic deployment on change).
- Control source code and attach pipeline using VCS (Github).

4.3 Gantt Chart

Task	4Jan-30Jan	31Jan-9Feb	10Feb-12Mar	13Mar-16Apr	17Apr-22Apr	23Apr-28Apr
Develop project proposal	27 days					
Analysis		10 days				
Designing			30 days			
Coding				34days		
Unit Testing					5 days	
Implementation						5 days

4.4 Requirements

4.4.1 Software requirements

Client Side

- Web Browser (Mozilla Firefox, Google chrome). Any browser that supports HTML5, Javascript, CSS3
- Any modern operating system (Windows, Gnu/Linux, Macos etc)

Server Side

- **Libraries:** IPython, JupyterLab, Pytorch, FastAi, Flask, Numpy, Scipy, Pandas etc.
- **Interpreter**: Python3,

4.4.2 Hardware requirements

Client Side

- **Processor:** 1Ghz or more,
- **Ram**: At least 2GB
- **Any working network device either embedded.**

Server Side

- **Processor:** High performance Intel core cpu (i7,i5), Base speed 2.5 or more,
- **Ram**: 16 GB
- **GPU**: Nvidia Tesla p1

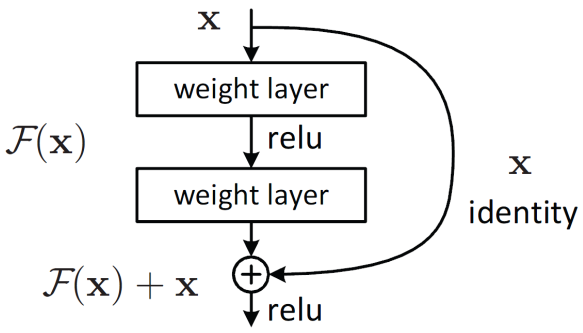
MODEL SELECTION

5.1 Model Selection

Model selection is the process of selecting one final machine learning model from among a collection of candidate machine learning models for a training dataset. In this project we will be using ResNet-50.

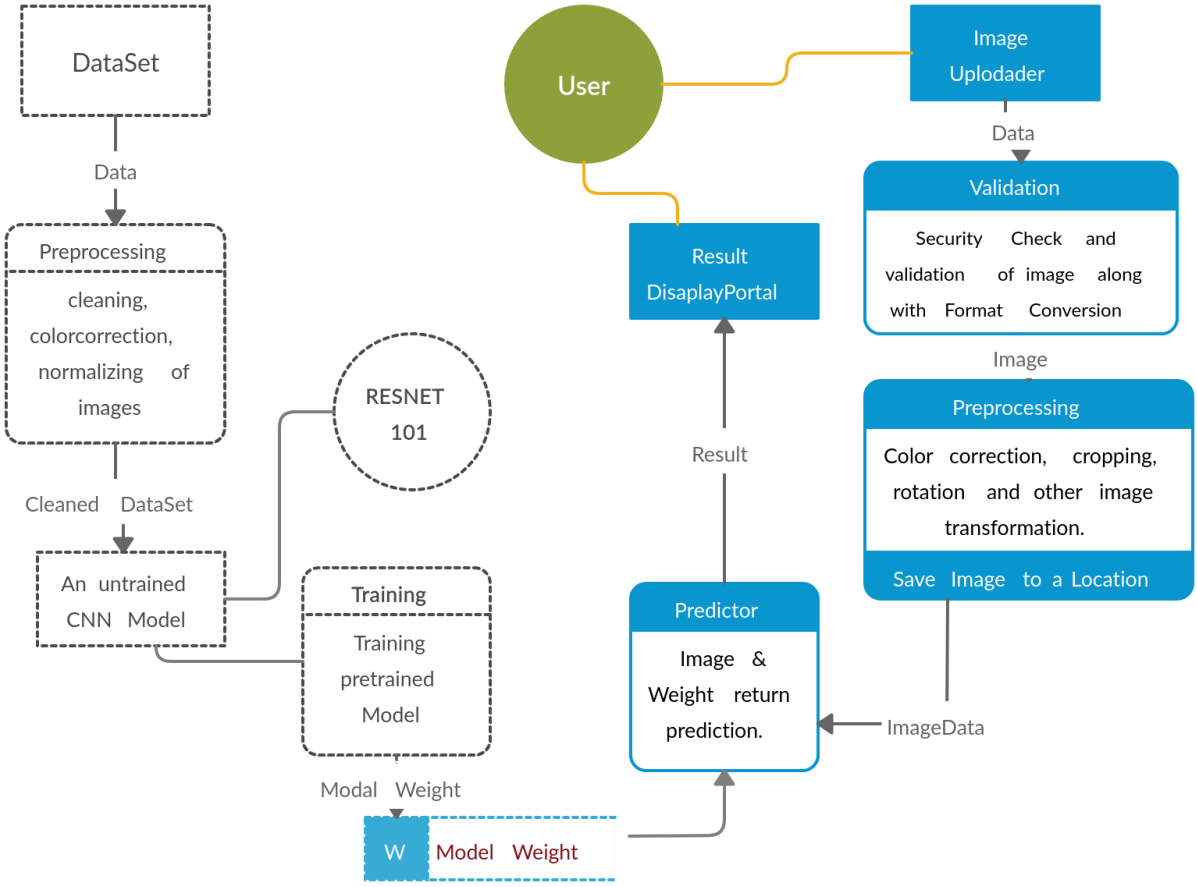
5.2 ResNet-50

ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pre-trained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.
ResNet-50 is a deep residual network. The "50" refers to the number of layers it has. It's a subclass of convolutional neural networks, with ResNet most popularly used for image classification.
The main innovation of ResNet is the skip connection. As you know, without adjustments, deep networks often suffer from vanishing gradients, as the model backpropagation, the gradient gets smaller and smaller. This makes retraining easier.



HIGH LEVEL DESIGN

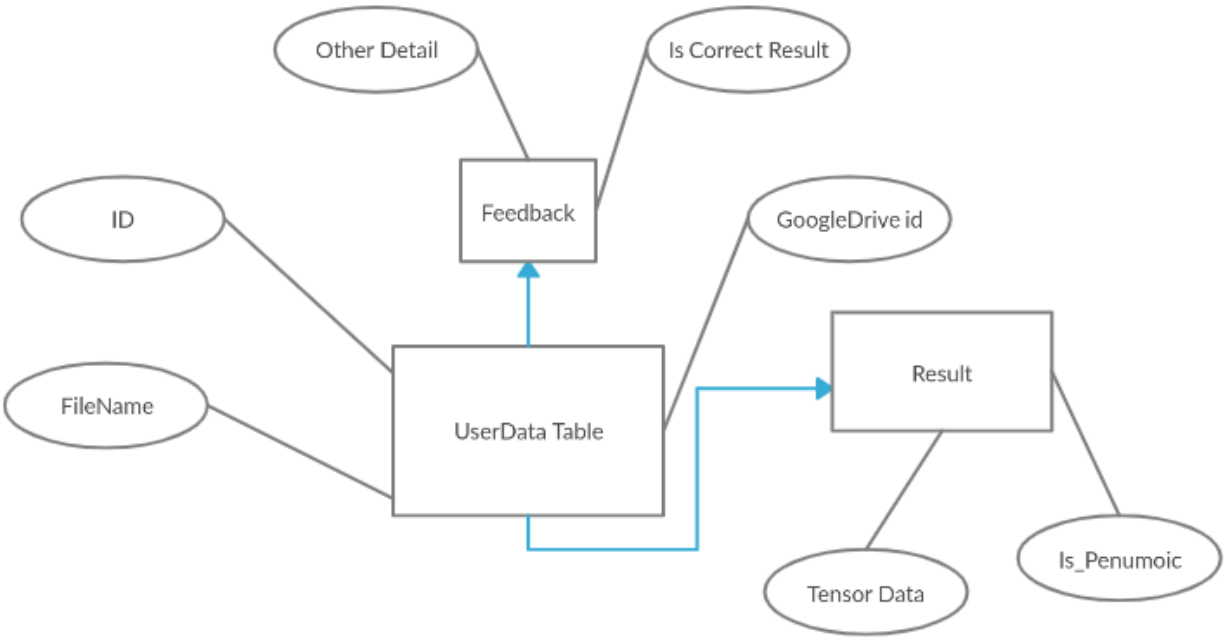
6.1 Data Flow Diagram



White part is one time operation only

Only required during training model

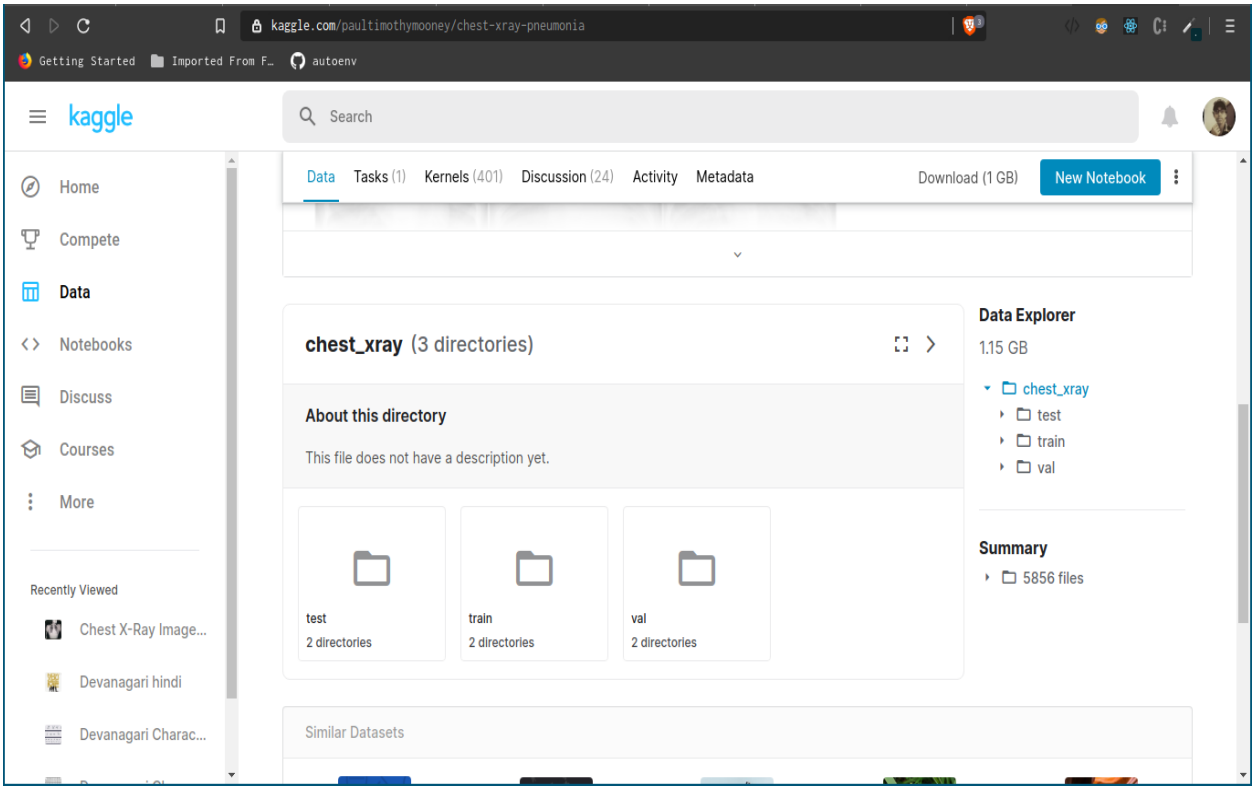
6.2 ER Diagram



- **UserData**
  - **Feedback**
  - **Result**

Images
&
ScreenShots

7.1 DataSet



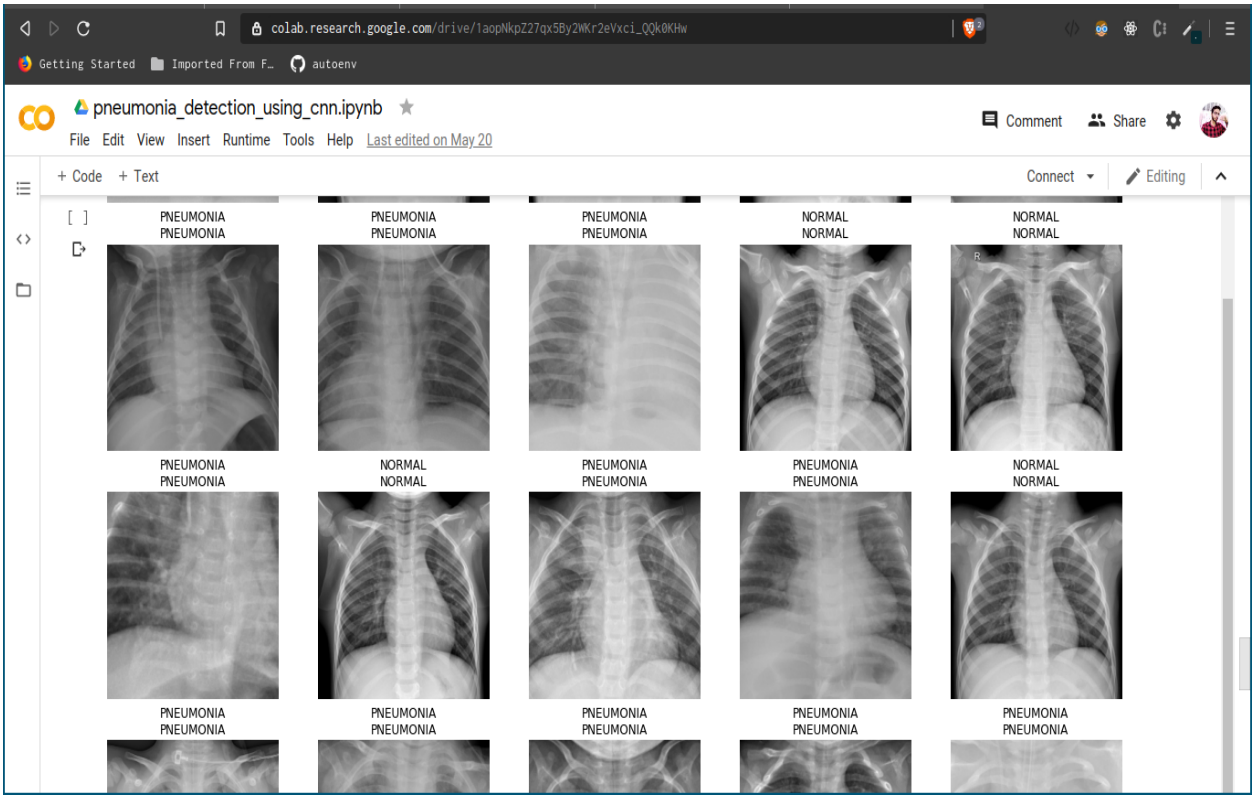
7.2 MetaData

Training Set	Test Set	Validation Set
1314 Normal, 3875 Pneumonic	334 Normal, 390 Pneumonic	8 Normal, 8 Pneumonic

7.2.1 DataSize.

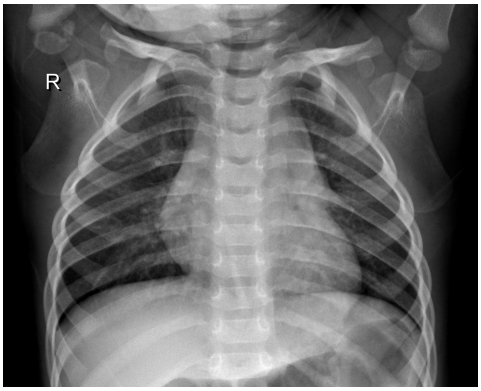
1.4 GB

7.2.2 Sample Data

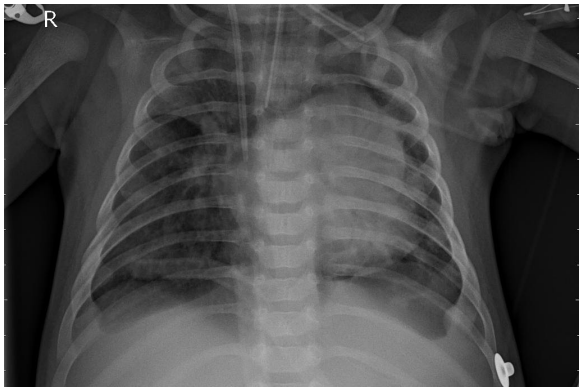


7.2.3 Data Classes

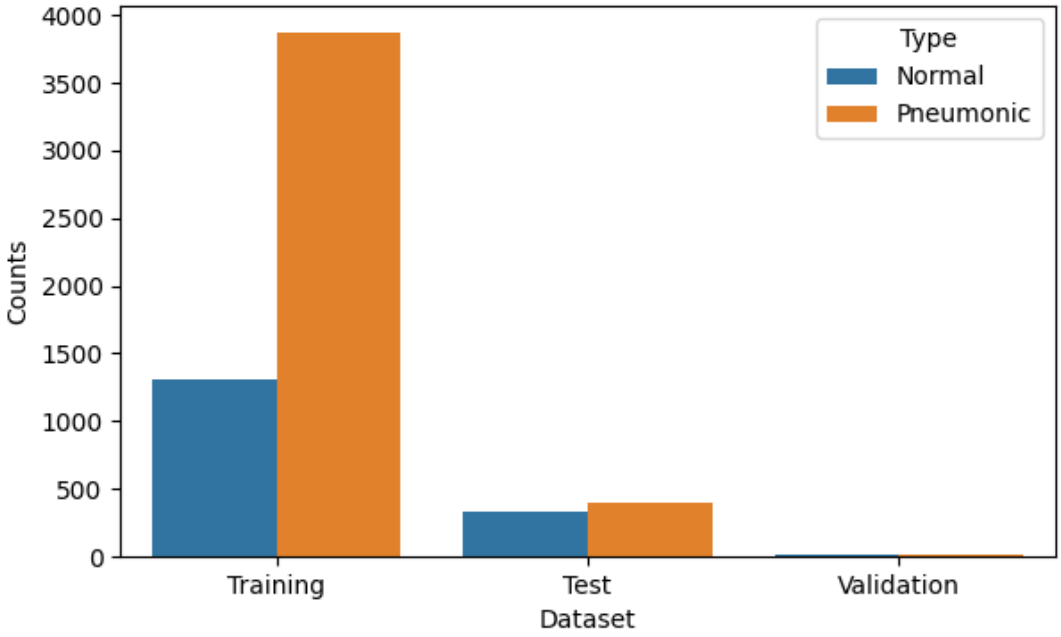
Normal,



Pneumonic



7.3 Data Distribution visualization



Observation:

By above Bar plot we can see that data is not uniformly distributed; it is heavily unbalanced that is common with most medical datasets.

Training Model

&

Metrics

8.1 Epoch Cycle and accuracy metrics

pneumonia_detection_using_cnn.ipynb

File Edit View Insert Runtime Tools Help Last edited on May 20

+ Code + Text

```
['NORMAL', 'PNEUMONIA'], 2, 9370, 2342)
```

```
learner = cnn_learner(data, models.resnet50, metrics=[error_rate, accuracy])
```

```
Downloading: "https://download.pytorch.org/models/resnet50-19c8e357.pth" to /root/.cach
100% 97.8M/97.8M [41:35<00:00, 41.1kB/s]
```

```
learner.fit_one_cycle(4)
```

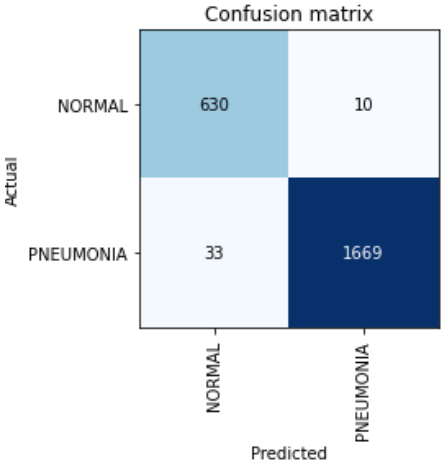
epoch	train_loss	valid_loss	error_rate	accuracy	time
0	0.343232	0.185680	0.062340	0.937660	10:31
1	0.183993	0.190342	0.060632	0.939368	10:11
2	0.132823	0.122265	0.034159	0.965841	10:15
3	0.118773	0.119570	0.040137	0.959863	10:18

```
learner.save("stage-1")
```

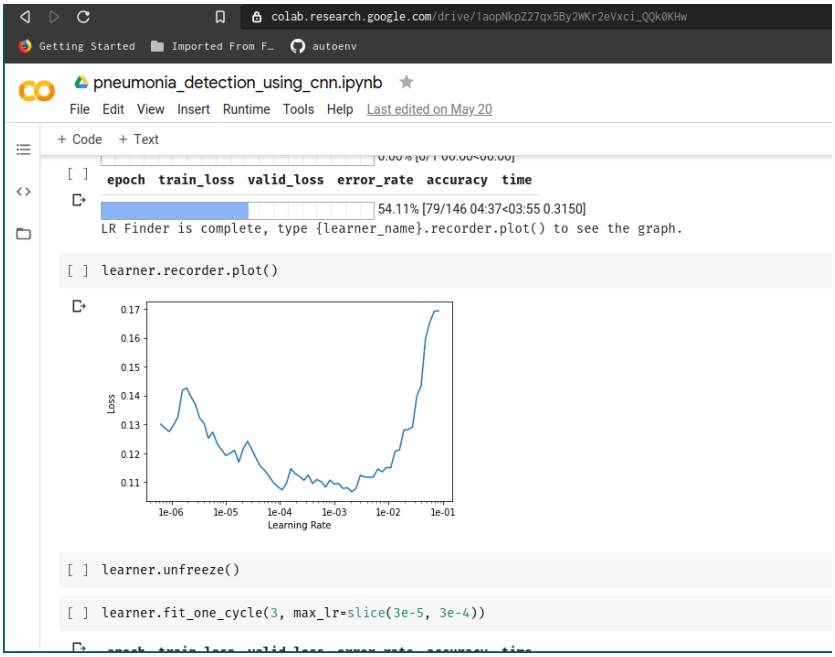
```
ClassificationInterpretation.from_learner(learner).plot_confusion_matrix()
```

Confusion matrix

8.2 Confusion Matrix (After Training)



8.3 Learning Rate



8.4 Validation on separate dataset.

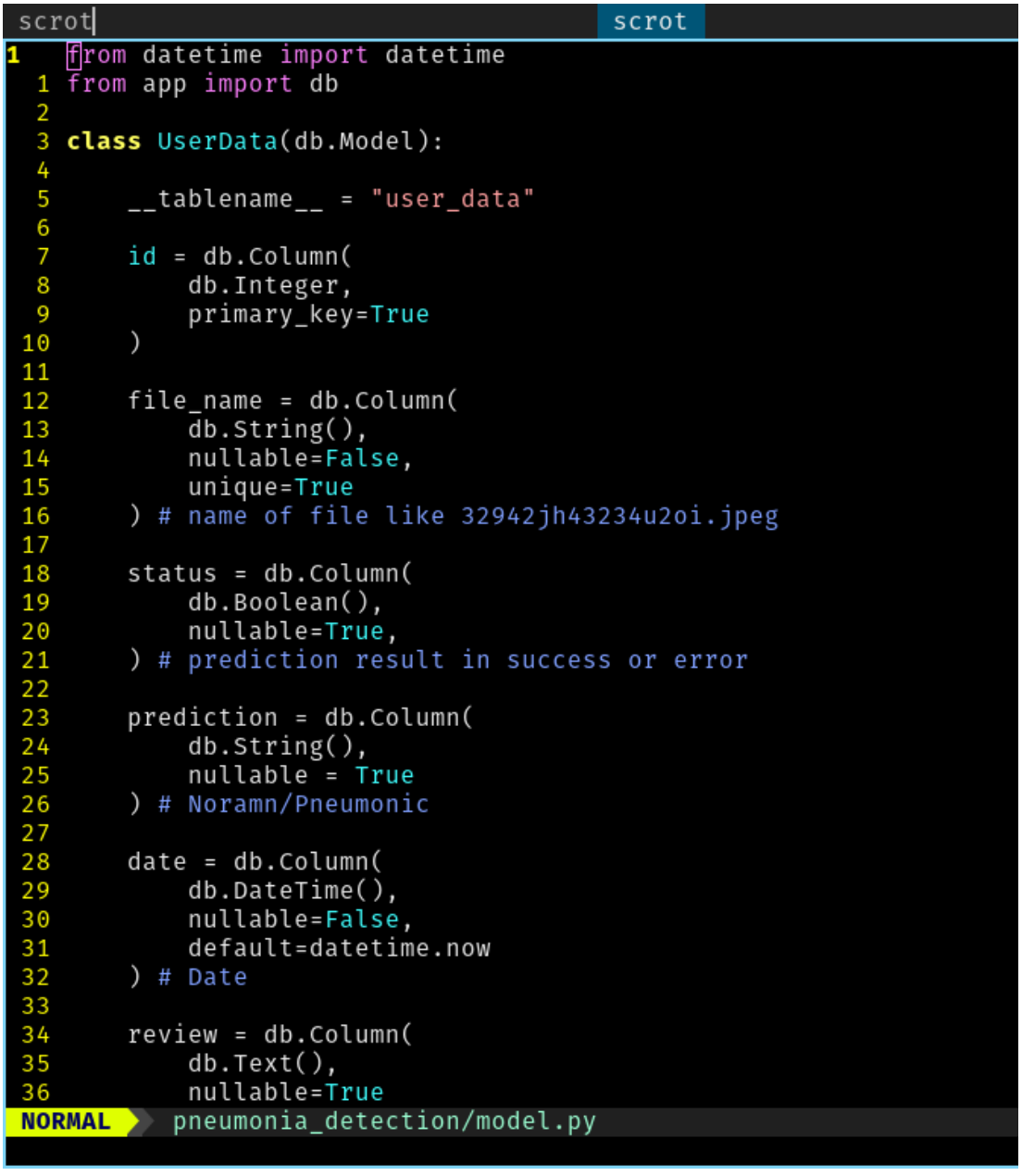
```
[ ] data2 = ImageDataBunch.from_folder(path=DATASET_PATH ,valid="test",ds_tfms=tfms, size=256, bs=64 ).normalize(imagenet_stats)
```

```
[ ] learner.validate(data2.valid_dl, metrics=[accuracy])
```

```
[0.064038604, tensor(0.9744)]
```

Implementation, Working & Deployment

9.1.1 DataBase Table (ORM Model)



9.1.2 Web Interface and Api Creation

```
scrot  3  4  5  6  7  8  9  [1]  nvim  scrot
 17 import os
 16 from uuid import uuid4
 15 from flask import request, abort, send_from_directory
 14 from app import app, db, UserData
 13 from config import UPLOAD_ALLOWED_EXTENSION, UPLOAD_FOLDER
 12
 11
 10 def allowed_file(filename):
  9     return '.' in filename and filename.rsplit('.', 1)[1].lower() in UPLOAD_ALLOWED_EXTENSION
  8
  7
  6 @app.route('/api/upload', methods=['POST'])
  5 def upload_file():
  4
  3     if 'file' not in request.files:
  2         return abort(400)
  1
 18     file = request.files['file']
  1     if file.filename == '':
  2         return abort(400)
  3
  4     if file and allowed_file(file.filename):
  5         file_extension = file.filename.rsplit('.', 1)[1].lower()
  6         file_name = str(uuid4())
  7
  8         file.save(
  9             os.path.join(
 10                 UPLOAD_FOLDER, file_name + "." + file_extension
 11             )
 12         )
 13
 14         data = UserData(file_name + "." + file_extension)
 15         db.session.add(data)
 16         db.session.commit()
 17         return {
 18             "id": file_name,
 19             "message": "Use this secret id for querying on your image."
NORMAL  pneumonia_detection/apis.py                                  python  utf
```

9.2.1 Querying on image using REST API

```
http -f https://pneumonia-detection-using-cnn.herokuapp.com/api/upload file@somfile.jpg
```

Rest API (Request parameter and Response Format)

1. Uploading image for querying

Method: POST
Parameter: <File Input> with name as file.
Endpoint: https://pneumonia-detection-using-cnn.herokuapps.com/api/upload

Sample Response

```
{
    "Id": "bb5c85e3-d539-4660-8e03-3149701b074e",
    "Message": "Use this secret id for query."
}
```

This id will be used to refer to your uploaded images in other requests.

2. Getting detection on the image.

Method: GET
Parameter: <Json request with id of your image> and Action set to "pred"
Endpoint: https://pneumonia-detection-using-cnn.herokuapp.com/api/action/

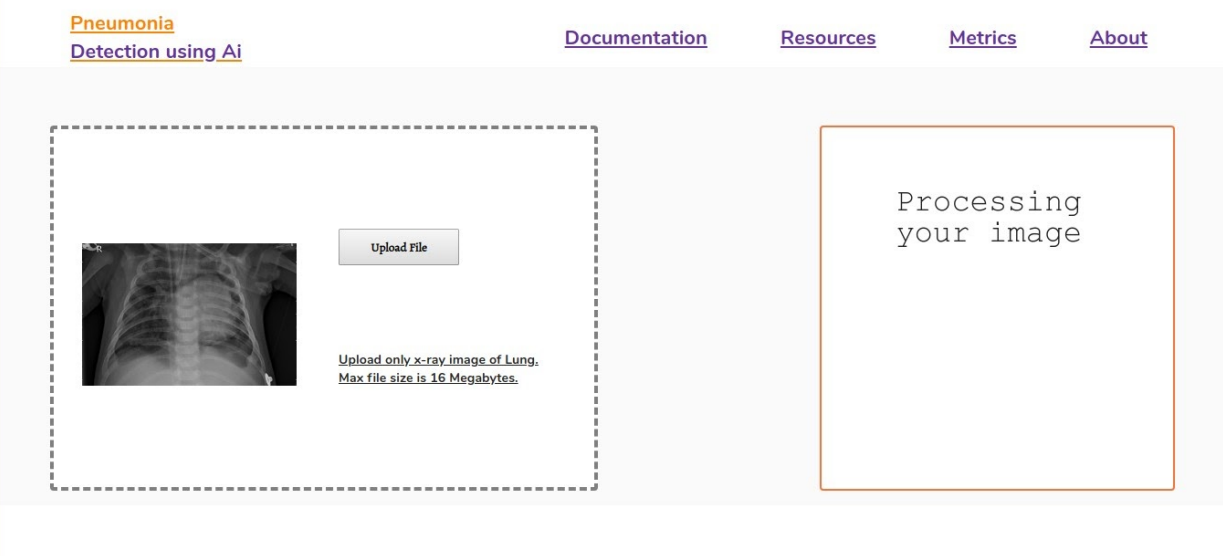
Sample Request

```
{
    "Id": "bb5c85e3-d539-4660-8e03-3149701b074e",
    "Action": "pred"
}
```

Sample Response

```
{
    "Id" : "bb5c85e3-d539-4660-8e03-3149701b074e",
    "Status": "success/error",
    "Result/error":"[Normal/Pneumonic] /Error log",
    "Additional-detail": "Any other detail",
    "Feedback-url": "URL to submit feedback ,
}
```

9.2.1 Querying using a web interface.

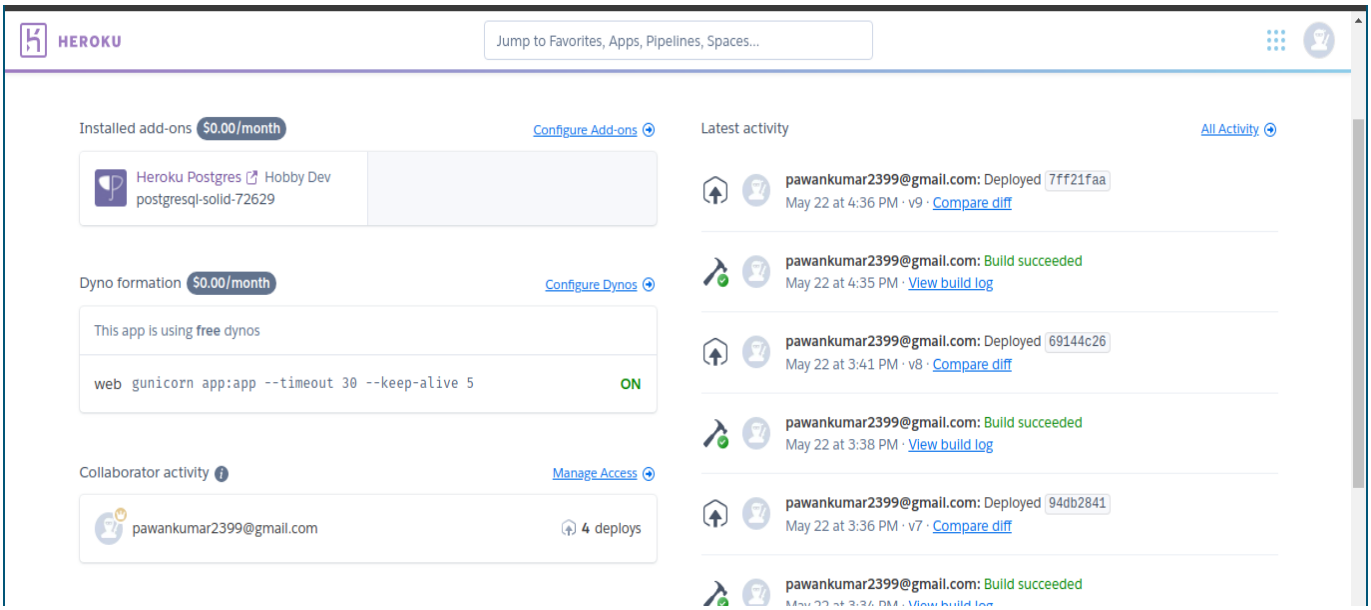


9.3.1 Deployment

Project is deployed on *Heroku*.
Plan: Free dynos.
Add-on: Heroku Postgres.
Url: *https://pneumonia-detection-using-cnn.herokuapps.com*
Github:
https://github.com/bellrd/pneumonia-detection-using-cnn.git

Screenshots

9.3.2 AppName: *pneumonia-detection-using-cnn*



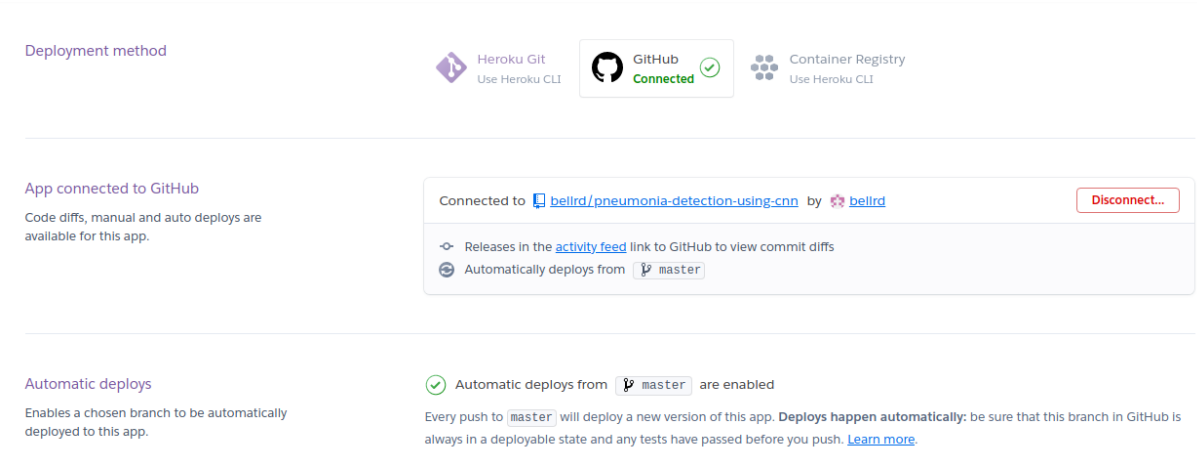
9.3.3 Build Logs:

Activity Feed > Build Log

```
-----> Python app detected
 !     Python has released a security update! Please consider upgrading to python-3.8.3
       Learn More: https://devcenter.heroku.com/articles/python-runtimes
-----> No change in requirements detected, installing from cache
-----> Installing SQLite3
-----> Installing requirements with pip
-----> Discovering process types
       Procfile declares types -> web
-----> Compressing...
       Done: 266.4M
-----> Launching...
       Released v9
       https://pneumonia-detection-using-cnn.herokuapp.com/ deployed to Heroku
```

Build finished

9.3.4 Resources & Add-ons:



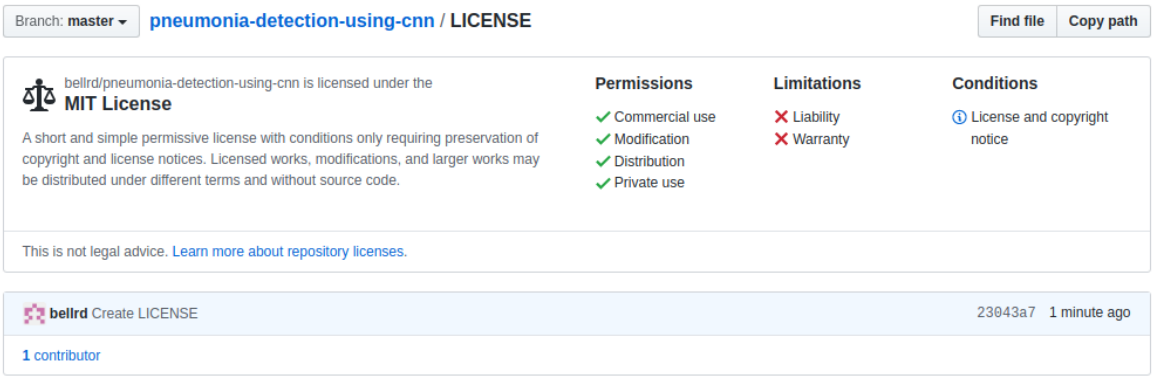
9.3.5 Source code and Permissive License:

This project is licensed under.

MIT LICENSE

https://github.com/belllrd/pneumonia-detection-using-cnn/LICENSE

Feel free to contribute code and send pull requests to improve upon this project.



Conclusion

We developed a model to detect and classify pneumonia from chest X-ray images taken from frontal views at high validation accuracy. e algorithm begins by transforming chest X-ray images into sizes smaller than the original. The next step involves the identification and classification of images by the convolutional neural network framework,which extracts features from the images and classies them.

Future Scope

We have demonstrated how to classify positive and negative pneumonia data from a collection of X-ray images. We build our model from scratch using deep learning, which separates it from other methods that rely heavily on Machine learning approach. In the future, this work may be extended to detect and classify X-ray images consisting of lung cancer and pneumonia.Distinguishing X-ray images that contain lung cancer and pneumonia has been a big issue in recent times.

Citations and References

https://www.researchgate.net/publication/332049903_An_Efficient_Deep_Learning_Approach_to_Pneumonia_Classification_in_Healthcare

https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

https://arxiv.org/abs/1512.03385

Source Code
https://github.com/bellrd/pneumonia-detection-using-cnn.git

Project url
https://penumonia-detection-using-cnn.herokuapps.com

Jupyter Notebook

https://colab.research.google.com/drive/1aopNkpZ27qx5By2WKr2eVxci_QQk0KHw?usp=sharing