Pathology Image Analysis For Lung Cancer Classification Using IBM Watson

1.INTRODUCTION:

1.1 Overview:

The main objective of this project is to detect whether the tumor present in a patient's lung is malignant or benign using Convolution Neural Network (CNN) .In a study published in Nature Medicine, researchers said that lung cancer caused an estimated 160,000 deaths in 2018, making it the most common cause of cancer death in the US. Lung cancer screenings that use low-dose tomography have been shown to reduce mortality by 20-43 percent, but there are still challenges that result in unclear diagnoses, subsequent unnecessary procedures, and high costs. Radiologists also usually have to look through dozens of 2D images within a single CT scan, and cancer can be hard to spot. Deep learning can offer a viable solution to these problems.

1.2 Purpose:

- Tumor region detection in pathology images allows us tocharacterize tumor shape accurately and extract tumor shape-based features.

- In this study, we developed a deep CNN model to automatically recognize tumor regions for lung ADC from H&E pathology images.

2. LITERATURE SURVEY :

2.1 Existing problem:

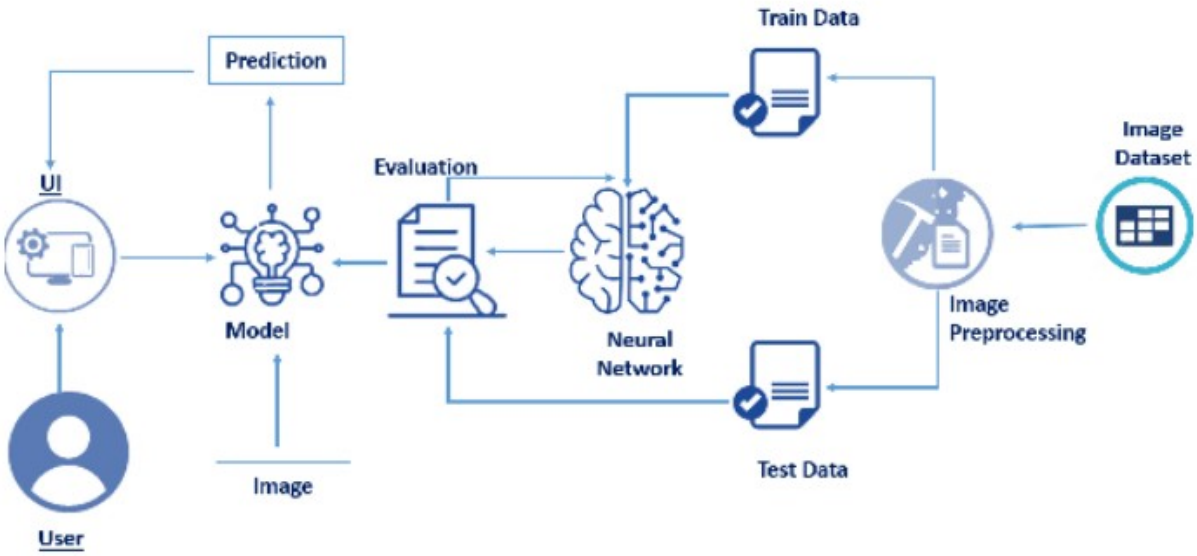
In existing system we use KNN to detect whether the tumor present in patient's lung is malignant or benign.

2.2 Proposed system:

In proposed system we use CNN to detect whether the tumor present in patient's lung in malignant or benign. CNN has more accuracy than KNN.

3. THEORITICAL ANALYSIS:

3.1 Block diagram:



3.2 HARDWARE / SOFTWARE DESIGNING:

Hardware requirements:

- Processor-i5
- RAM-8GB

Software requirements:

- Anaconda Navigator :

  Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed onWindows, Linux, and macOS.Conda is an open-source, cross-platform,package management system.Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For thisproject, we will be using Jupiter notebookand spyder.

- Jupyter Notebook
- Browsers
- Languages – Python, HTML5, CSS3, JSS

4. EXPERIMENTAL INVESTIGATIONS:

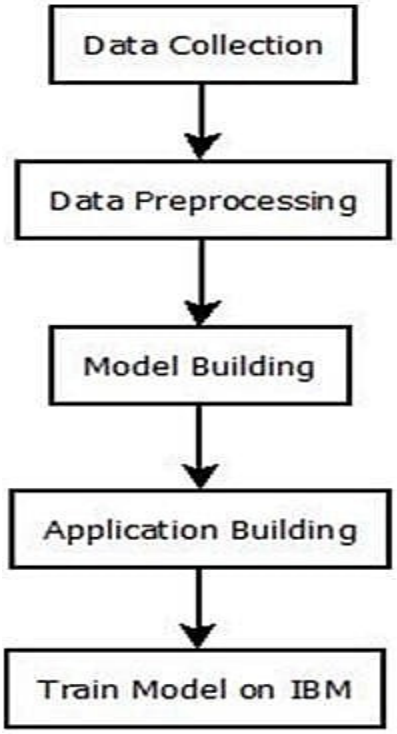
1.Dataset collection:

Deep Learning depends heavily on data, without data, a machine can't learn. It is the most crucial aspect that makes neural network training possible.In Deep Learning projects, we need a training dataset. It is the actual data set used to train the model for performing variousactions.We can collect datasets from different open sources like kaggle.com,data.gov, UCI machine learning repository, etc. The datasetused for this project was obtained from Kaggle. Please refer to the link to download the data set and to know about the dataset.This dataset contains two columns, cancer and Non Cancer.we can collect datasetfrom different open sources like kaggle.com Kaggle is the world's largestdata science communitywith powerful toolsand resources to help you achieve your data science goals.

2.Data preprocessing:

- Import the model building Libraries
- Initializing the model
- Adding CNN Layers
- Adding Hidden Layer
- Adding Output Layer
- Configure the Learning Process
- Training and testing the model
- Saving the model

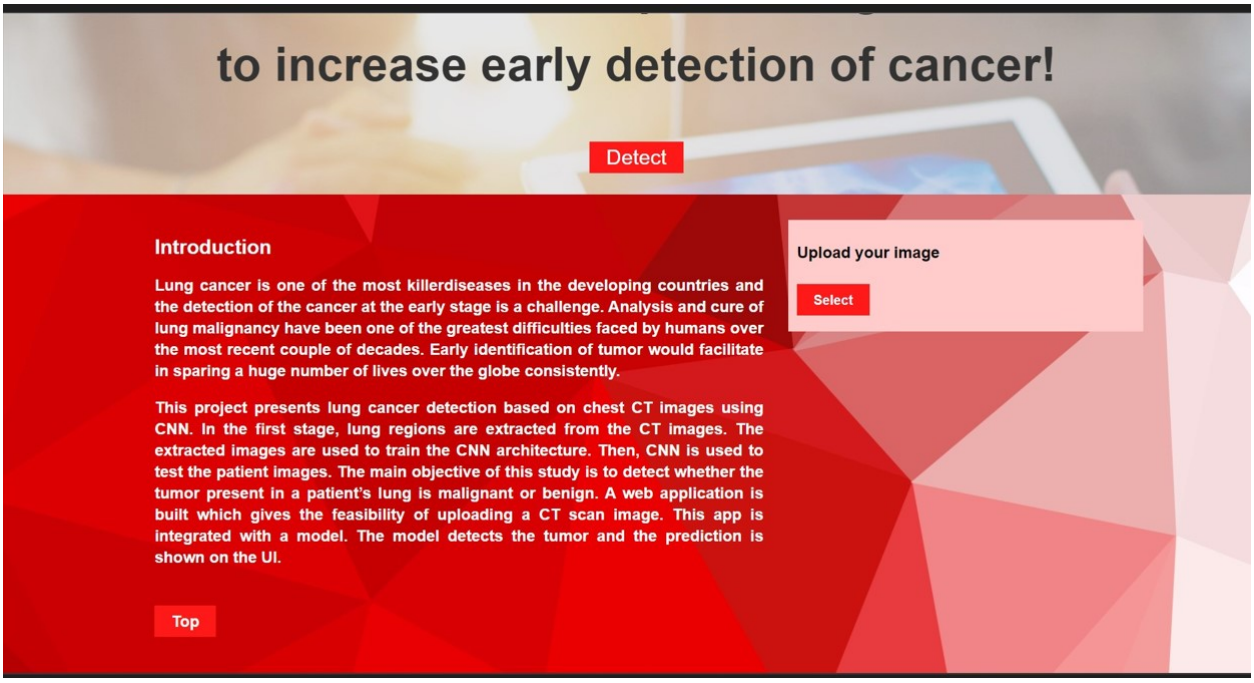
5. FLOW CHART:



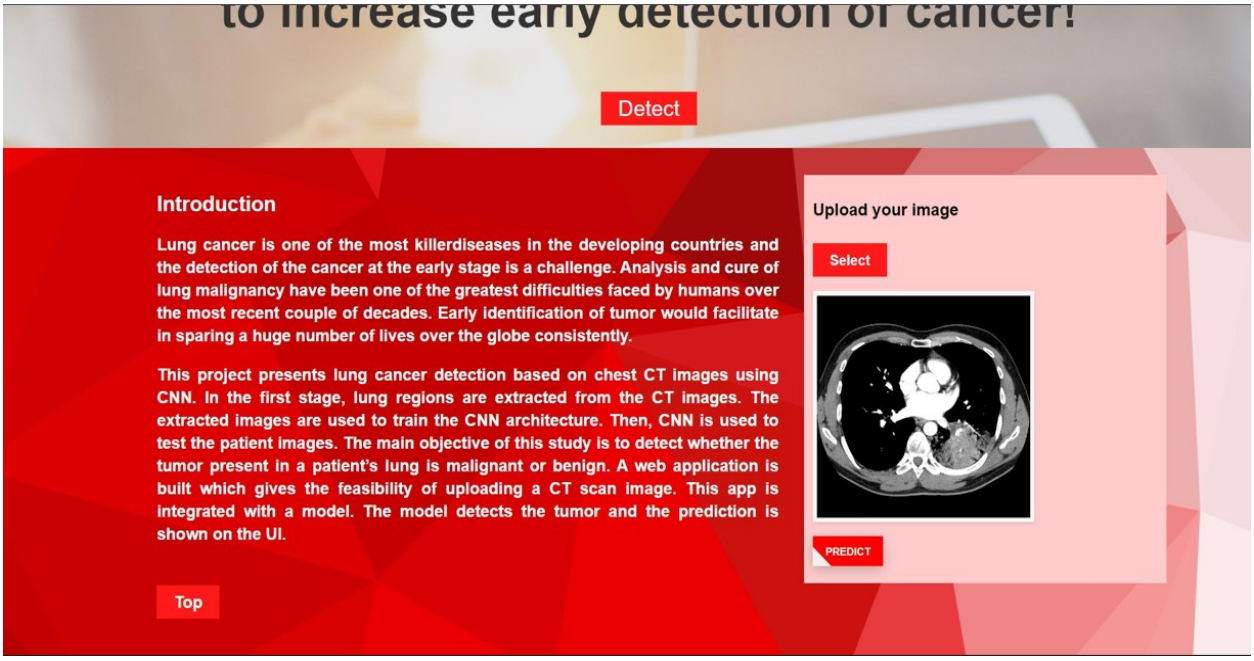
6. RESULT:

This is our home page. At the bottom we click on detect.

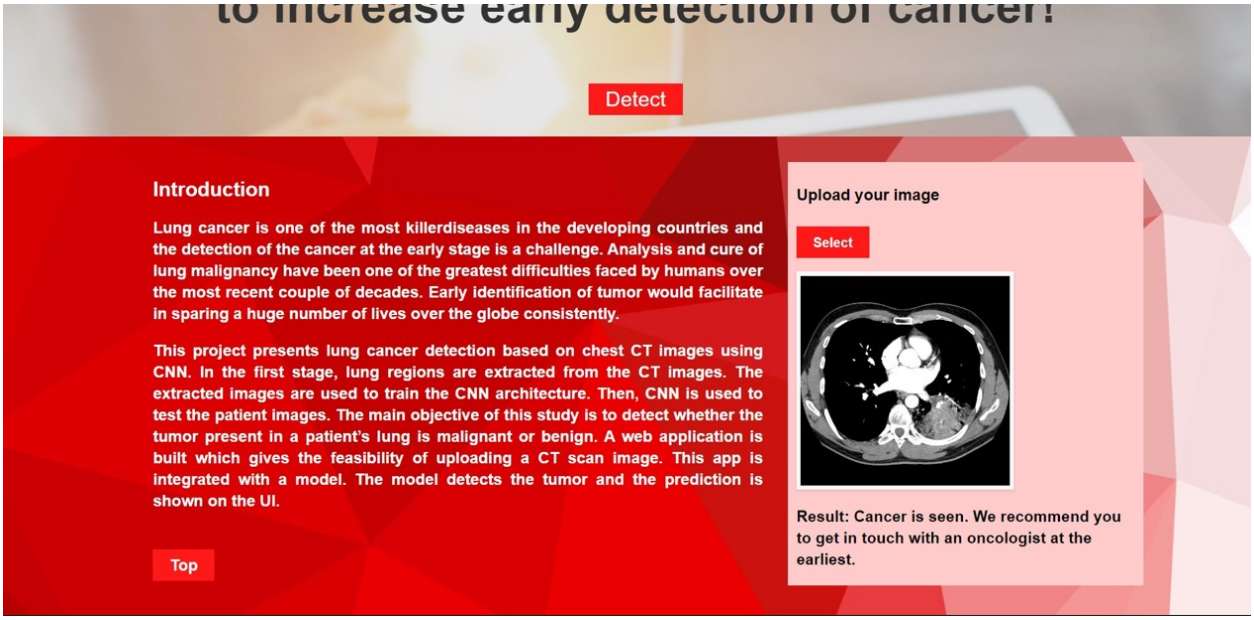
- After clicking detect we get a page where we can upload a image.



- Now we select a image from the given dataset and upload the image.



- After uploading the image click on predict button we get the result whether it is cancer or not.



7. ADVANTAGES AND DISADVANTAGES:

- **ADVANTAGES:**

- Automatic tumor region detection in pathology images allows us to better characterize tumor region boundaries and extract tumor shape and boundary-based features.

- **DISADVANTAGES:**

- More time consuming.

8. APPLICATIONS:

- Pathology image analysis that tackles this issue would be helpful for **diagnoses and subtyping of lung carcinoma**.

9. CONCLUSION:

Our pipeline for tumor region recognition and risk- score prediction based on tumor shape features serves as an objective prognostic method independent of other clinical variables, including age, gender, smoking status and stage. The tumor region heatmaps generated by our model can help pathologists locate tumor regions in pathology tissue images swiftly and accurately. The model development pipeline can also be used in other cancer types, such as breast and kidney cancer. We point out some promising future directions for lung cancer pathology image analysis, including multi-task learning, transfer learning, and model interpretation.

10. FUTURE SCOPE:

- Interpreting Deep Learning Models and Mining Knowledge from Trained Neural Networks.
- Comprehensive Lung Cancer Diagnosis and Prognosis through Multi-Task Learning.
- Utilization and Integrating Multiple Methods of Medical Imaging.

11.BIBILOGRAPHY:

1. Dela Cruz C.S., Tanoue L.T., Matthay R.A. Lung cancer: Epidemiology, etiology, and prevention. Clin. Chest Med. 2011;32:605–644. doi: 10.1016/j.ccm.2011.09.001. - DOI - PMC - PubMed
2. de Groot P.M., Wu C.C., Carter B.W., Munden R.F. The epidemiology of lung cancer. Transl. Lung Cancer Res. 2018;7:220–233. doi:

10.21037/tlcr.2018.05.06. - DOI - PMC - PubMed

3. Barta J.A., Powell C.A., Wisnivesky J.P. Global Epidemiology of Lung Cancer. Ann. Glob. Health. 2019;85:e8. doi: 10.5334/aogh.2419. - DOI - PMC - PubMed
4. van den Bent M.J. Interobserver variation of the histopathological diagnosis in clinical trials on glioma: A clinician's perspective. Acta Neuropathol. 2010;120:297–304. doi: 10.1007/s00401-010-0725-7. - DOI - PMC - PubMed
5. Cooper L.A., Kong J., Gutman D.A., Dunn W.D., Nalisnik M., Brat D.J. Novel genotype-phenotype associations in human cancers enabled by advanced molecular platforms and computational analysis of whole slide images. Lab. Investig. A J. Tech. Methods Pathol. 2015;95:366–376. doi: 10.1038/labinvest.2014.153. - DOI - PMC - PubMed

12. APPENDIX:

Source Code :

```
from keras.models import Sequential

from keras.layers import Dense

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten


from keras.preprocessing.image import ImageDataGenerator

train_datagen=ImageDataGenerator(rescale=1./255,shear_range = 0.2,zoom_range = 0.2,horizontal_flip = True)


test_datagen = ImageDataGenerator(rescale = 1./255)
```

```
x_train=train_datagen.flow_from_directory(
r'D:\InternshipProject\Lungcancer\trainset'
 ,target_size = (64,64),batch_size = 32 ,
    class_mode = 'binary',color_mode="grayscale")


x_test = test_datagen.flow_from_directory(
    r'D:\InternshipProject\Lungcancer\testset',
    target_size = (64,64),batch_size = 32 ,
    class_mode = 'binary',color_mode="grayscale")


print(x_train.class_indices)


model = Sequential()


model.add(Convolution2D(32,(3,3),input_shape=
(64,64,1),activation = "relu"))


model.add(MaxPooling2D(pool_size = (2,2)))


model.add(Flatten())


model.add(Dense(units=128,kernel_initializer=
"random_uniform",activation = "relu"))
```

```
model.add(Dense(units=1,kernel_initializer=
"random_uniform",activation = "sigmoid"))


model.compile(loss="binary_crossentropy",optimizer=
"adam",metrics = ["accuracy"])


model.fit_generator(x_train , steps_per_epoch = 4,
                    epochs = 100, validation_data = x_test,
                    validation_steps = 2)


model.save("LungCancer.h5")
```

app.py :

```
import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import tensorflow as tf
global graph
graph = tf.compat.v1.get_default_graph()
from flask import Flask , request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
import cv2
```

```
app = Flask(__name__)
model = load_model("Lungcancer.h5")

@app.route('/')
def index():
    return render_template('base.html')

@app.route('/predict',methods = ['GET','POST'])
def upload():
    if request.method == 'POST':
        f = request.files['image']
        print("current path")
        basepath = os.path.dirname(__file__)
        print("current path", basepath)
        filepath = os.path.join(basepath,'uploads',f.filename)
        print("upload folder is ", filepath)
        f.save(filepath)

        img = image.load_img(filepath,target_size = (64,64))
    gray = cv2.cvtColor(np.float32(img), cv2.COLOR_BGR2GRAY)
        print(gray.shape)
        x = image.img_to_array(gray)
        x = np.expand_dims(x,axis =0)
```

```
        preds = np.argmax(model.predict(x))


        print("prediction",preds)


        index = ['Cancer','NonCancer']
        text = str(index[preds])


        if (text == "Cancer"):
            text = "Cancer is seen. We recommend you to get in touch with an oncologist at the earliest."
        else:
            text = "Cancer not seen. Stay safe and healthy."


    return render_template("result.html",predict_text=text)
if __name__ == '__main__':
    app.run(debug = False, threaded = False)
```