LUNG CANCER DETECTION SYSTEM

A PROJECT REPORT

submitted By

SREERAJ V

TVE21MCA053

to

the APJ Abdul Kalam Technological University
in partial fullfilment of the requirements for the award of the degree

of

Master of Computer Applications

Department of Computer Applications

College of Engineering

Trivandrum-695016

NOVEMBER 2022

Declaration

I undersigned hereby declare that the project report titled **"Lung Cancer Detection System"** submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Smt. Sreerekha V K, Asst.Professor. This submission represents my ideas in my words and where ideas or words of others have been included. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity as directed in the ethics policy of the college and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and/or University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title.

Place : Trivandrum **Sreeraj V**
Date : 13/11/2022

DEPARTMENT OF COMPUTER APPLICATIONS

COLLEGE OF ENGINEERING
TRIVANDRUM

CERTIFICATE

This is to certify that the report entitled **Lung Cancer Detection System** submitted by **Sreeraj V** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by him under my guidance and supervision. This report in any form has not been submitted to any University or Institute for any purpose.

Internal Supervisor   External Supervisor

Head of the Dept

Acknowledgement

First and for most I thank **GOD** almighty and to my parents for the success of this project. I owe a sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely thankful to **Dr Suresh Babu V**, Principal, College of Engineering Trivandrum for providing me with the best facilities and atmosphere which was necessary for the successful completion of this project.

I am extremely grateful to **Deepa S S**, HOD, Dept of Computer Applications, for providing me with best facilities and atmosphere for the creative work guidance and encouragement.

I express our sincere thanks to **Smt. Sreerekha V K**, Asst. Professor, Department of Computer Applications, College of Engineering Trivandrum for her valuable guidance, support and advice that aided in the successful completion of my project.

I profusely thank other Asst. Professors in the department and all other staffs of CET, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project. No words can express my humble gratitude to my beloved parents and relatives who have been guiding me in all walks of my journey.

Sreeraj V

Abstract

Pulmonary cancer also known as lung carcinoma is the leading cause for cancer-related death in the world. Early stage detection cancer detection using computed tomography (CT) could save hundreds of thousands of lives every year. However analysing hundreds of thousands of these scans are an enormous burden for radiologists and too often they suffer from observer fatigue which can reduce their performance. Therefore, a need to read, detect and provide an evaluation of CT scans efficiently exists. This system presents an approach which utilizes a Convolutional Neural Network (CNN) to classify the tumors found in lung as malignant or benign. The accuracy obtained by means of CNN is high, which is more efficient when compared to accuracy obtained by the traditional existing systems. This done by applying convolutional neural network technique to a data set of lung cancer CT scans.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Lung cancer is one of the most dreadful diseases in the developing countries and its mortality rate is 19.4Early detection of lung tumor is done by using many imaging techniques such as Computed Tomography (CT), Sputum Cytology, Chest X-ray and Magnetic Resonance Imaging (MRI). Detection means classifying tumor two classes (i)non-cancerous tumor (benign) and (ii)cancerous tumor (malignant). The chance of survival at the advanced stage is less when compared to the treatment and lifestyle to survive cancer therapy when diagnosed at the early stage of the cancer. Manual analysis and diagnosis https://www.overleaf.com/project/637119497d9254c7e5110d81 system can be greatly improved with the implementation of image processing techniques. A number of researches on the image processing techniques to detect the early stage cancer detection are available in the literature. But the hit ratio of early stage detection of cancer is not greatly improved. With the advancement in the machine learning techniques, the early diagnosis of the cancer is attempted by lot of researchers. Neural network plays a key role in the recognition of the cancer cells among the normal tissues, which in turn provides an effective tool for building an assistive AI based cancer detection. The cancer treatment will be effective only when the tumor cells are accurately separated from the normal cellsClassification of the tumor cells and training of the neural network forms the basis for the machine learning based cancer diagnosis.This paper presents a Convolutional Neural Network (CNN) based echnique to classify the lung tumors as malignant or benign.

Chapter 2

Problem Defnition and Motivation

2.1 Existing System

Existing methods use traditional segmentation methods,they are not efficient.Manual segmentation of the lung tumours for cancer diagnosis, from large amount of CT Scan images generated in clinical routine, is a difficult and time consuming task.The existing CAD system used for early detection of lung cancer with the help of CT images has been unsatisfactory because of its low sensitivity and high False Positive Rates (FPR).

2.2 Proposed System

The prorposed system is a deep convolutional neural network architecture for classify Adenocarcinoma,large cell carcinoma,squamous cell carcinoma Non-Small Cell Lung Cancer (NSCLC) with high acccuracy within a short time. n the first stage,lung regions are extracted from CT image and inthat region each slices are segmented to get tumors. The segmented tumor regions are used to train CNN architecture. Then, CNN is used to test the patient images. The main objective of this study is to detect whether the tumor present in a patient's lung is malignant or benign. The proposed architecture is a computationallly lightweight model with a small number of convolutional, max-pooling layers and training iterations.

2.3 Dataset

The dataset that has been used in the experiments and test based on Chest CT-Scan images Dataset from Kaggle. The dataset has been divided into three folders (Training, Testing and Validation),with sub-folders for each class(adenocarcinoma, large.cell.carcinoma, normal, squamous.cell.carcinoma). There are 1000 CT Scan images organised into four classes adenocarcinoma(338), large.cell.carcinoma(187), normal(215), quamous.cell.carcinoma(260).

Chapter 3

Literature Review

The automated evaluation of essays written by the humans is a hot problem since ages. Since essay is the best way of evaluating the academic excellence the major problem which we encountered with essay is the time consumed for the evaluation. As a part of my literature review I went through various papers and presentations on this topic. The quick summary of my findings are specified in this chapter.

3.1 Pankaj Nanglia, Sumit Kumar et all proposed a unique hybrid algorithm called as Kernel Attribute Selected Classifier in which they integrate SVM with Feed-Forward Back Propagation Neural Network, which helps in reducing the computation complexity of the classification.

For the classification they proposed three block mechanisms, pre-process the dataset is the first block. Extract the feature via SURF technique followed by optimization using genetic algorithm is the second block and the third block is classification via FFBPNN.

3.2 Chao Zhang,Xing Sun, Kang Dang et all perform a sensitivity analysis using the multicenter data set

They chosen two categories Diameter and Pathological result. Diameter were divided into three sub groups.0-10mm,10-20mm,20- 30mm. In 0-10mm group sensitivity 85.7 prcentage (95 percentage Cl,70.8 percentage-100.0 percentage) and specificity 91.1 percentage(95 percentage Cl, 86.8 percentage-95.2 percentage) were found. In 10-20mm group sensitivity 85.7 percentage (95 percentage Cl,77.1 percentage-94.3 percentage) and specificity 90.1 percentage (95 percentage Cl, 84.8 percentage-95.4percentage) were found. In 20-30mm group sensitivity 78.9 percentage (95 percentage Cl,66.0percentage-91.8 percentage) and specificity 91.3 percentage (95 percentage Cl, 83.2 percentage-99.4 percentage) were found

3.3 Nidhi S. Nadkarni and Prof. Sangam Borkar focuses their study mainly on the classification of lung images as normal and abnormal

In their proposed method median filter was used to eliminate impulse noise from the images. Mathematical morphological operation enables accurate lung segmentation and detect tumour region. Three geometrical features i.e. Area, perimeter, eccentricity was extracted from segmented region and fed to the SVM classifier for classification

3.4 Ruchita Tekade, Prof. DR. K. Rajeswari studied the concept of lung nodule detection and malignancy level prediction using lung CT scan images

This experiment has conducted using LIDC$_IDRI, LUNA16andDataScienceBowl2017datasetsonCU$
$NETarchitectureforsegmentationoflungnodulefromlungCTscanimagesand3DmultigraphVGGlikea$

3.5 Moffy Vas, Amita Dessai, studied mainly on the classification of lung images cancerous and non-cancerous.

n their proposed method pre-processing was done, in which unwanted portion of the lung CT scan was removed. They used median filter to eliminate salt and pepper noise. Mathematical morphological operation enables accurate lung segmentation and detect tumour region. Seven extracted features i.e. energy, correlation, variance, homogeneity, difference entropy, information measure of correlation and contrast respectively was extracted from segmented region and fed to the feed forward neural network with back propagation algorithm for classification. The algorithm looks for the least of the error function in the weight space gradient descent method.

3.6 Vaishnavi. D1, Arya. K. S2, Devi Abirami. T3 , M. N. Kavitha4, studied on lung cancer detection algorithm.

In pre-processing they used Dual-tree complex wavelet transform (DTCWT)in which the wavelet is discretely sampled. GLCM is second order statistical method for texture analysis which provide a tabulation of how different combination of Gray level co-occur in an image. It measures the variation in intensity at the pixel of interest. They used Probability Neural Network (PNN) classifier evaluated in term of training performance and classification accuracy. It gives fast and accurate classification.

Chapter 4

Requirement Analysis

4.1 Hardware Requirements

- Feature extraction using NLP
- Training and testing of model
- Develope a user interface
- connect the UI with the model.

4.1.1 Software Requirements

- Operating System : Linux/Windows
- Platform : Python
- Librarie used : nltk, pandas, matplotlib, numpy, sklearn,

4.2 Technologies Used

4.2.1 Machine Learning

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.Machine Learning is an essential skill for any aspiring data analyst and data scientist, and also for those who wish to transform a massive amount of raw data into trends and predictions. Learn this skill today with Machine Learning Foundation – Self Paced Course , designed and curated by industry experts having years of expertise in ML and industry-based projects.

4.2.2 Deep Learning

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention.

CNN

Convolutional Neural Network or CNN is a type of artificial neural network, which is widely used for image/object recognition and classification. Deep Learning thus recognizes objects in an image by using a CNN. CNNs are playing a major role in diverse tasks/functions like image processing problems, computer vision tasks like localization and segmentation, video analysis, to recognize obstacles in self-driving cars, as well as speech recognition in natural language processing. As CNNs are playing a significant role in these fast-growing and emerging areas, they are very popular in Deep Learning.

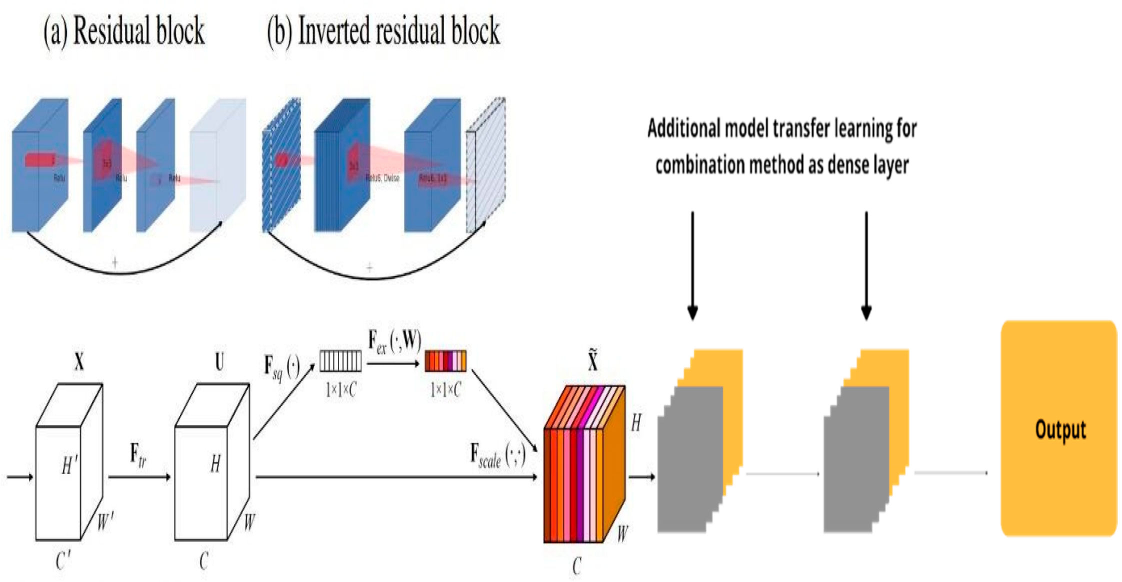


Figure 4.1: Model architecture

4.3 Functional Requirements

The functional requirements includes all the activities or processes that should be achieved by the proposed system. It includes

- **NumPy:** NumPy is a general purpose language. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with python. It has advanced math functions and a rudimentary scientific computing package.

- **pandas:** pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

- **matplotlib:** It’s used for the visualisation of data in python programming language. It’s implemented to work with the wider scipy stack and it’s built on numpy arrays. It’s a multi platform data visualization technique. It was developed in 2002 by John Hunter. Visualization is the most efficient way to understand the data. Using this library, we can represent our data in various plots such as line, bar, histogram, scatter etc.

- **keras:** Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation. Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep-learning models on clusters of Graphics processing units (GPU) and tensor processing units (TPU).

- **tensorflow:**TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. The TensorFlow platform helps you implement best practices for data automation, model tracking, performance monitoring, and model retraining. Using production-level tools to automate and track model training over the lifetime of a product, service, or business process is critical to success.

Chapter 5

Design And Implementation

The proposed system is a deep convolution neural network architecture for classify adenocarcinoma, large.cell.carcinoma, squamous.cell.carcinoma lung cancers.

5.1 Overall Design

The proposed system follows client server architecture. That is the lung cancer detection system has a client part and a server part as well. The client part is used by the user to input the CT Scan image which is to be evaluated. The input is passed to server and the evaluated result is given back to the client. The server side is developed in Python and the client side is built using HTML and Python.

5.1.1 System Design

The system is web based. The input is taken from the user through a web page and the input is passed to the python program running in the server side. The server program perform tasks such as pre processing and feature extraction on the input data. The architecture of the system consist of several steps where the execution starts from taking an input image from the dataset followed by the image pre-processing,image enhancement,and the lung cancer classification using Deep Convolutional Neural Network. Finally the output is observed after all the above mentioned steps are completed. The results of these processes are used to evaluate the input using the pre-trained model.

The model is created using the data obtained from Kaggle.com. The dataset has been divided into three folders (Training,Testing and Validation), with sub-folders for each class(adenocarcinoma, large.cell.carcinoma, normal, squamous.cell.carcinoma). There are 1000 CT Scan images organised into four classes adenocar- cinoma(338), large.cell.carcinoma(187), normal(215), quamous.cell.carcinoma(260).

5.2 Methodology

There are two parts in this project. The first part is the creation of the model and the second one is the creation of user program which will work with the pre-trained model.

The main process of the lung cancer detection system is the creation of the trained model. The major steps in the model creation Feature extraction, training, testing and model evaluation. The major steps in the model creation are mentioned below.

5.2.1 Training

The proposed model consists of a convolution part and a classsifier part. Here we are using Transfer learning in which we use pre-trained model EfficientNetB3 as the starting point of our new model.Classification part has two dense layers and one dropout layer. The training process has 100 epochs with batch size 40. Using abatch size of 40 , means that 40 samples are passed at a time to the trained model until all training data is passed to complete one epoch.

SEQUENTIAL:

- To initialize the neural network, we create an object of the sequential class.
- model = Sequential()

CONVOLUTION:

- The convolutional layer is implemented using the pre-trained model EfficientNetB3 which has more than 400 layers.
- EfficientNet is a convolutional neural network architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient. Unlike conventional practice that arbitrary scales these factors, the EfficientNet scaling method uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients.

- base$_{m}odel = tf.keras.applications.efficientnet.EfficientNetB3()$

BATCH NORMALISATION:

- During the training process of a CNN model, the distribution of input values for a specific layer depends on the previous layers of that model. This variability causes Overfitting and reduces the learning rates.
- Batch Normalization is hired to speed up the training process and decrease the Overfitting issue by standardizing the input vector in a way that eliminates the noisy features, which stabilizes the training process.
- The normalization allows to use lower dropouts rate because it acts as a regularizer and input to this is an vector.
- model.add(BatchNormalization())

POOLING:

- The Pooling layer is responsible for reducing the spatial size of the convolved feature.This is to decrease the computational power required to process the data through dimensionality reduction. Further more , it is useful for extracting dominant features which are rotational and positional invariant , thus maintaining the process of effectively training of the model.
- There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand , Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Here we are using max pooling.
- In this step we reduce the size of the feature map while not losing important image information.
- base$_{m}odel = tf.keras.applications.efficientnet.EfficientNetB3(pooling =' max')$

FLATTENING:

- In this step , all the pooled feature maps are taken and put into a single vector for inputting it to the next layer

- The flatten function flattens all the feature maps into a long column.
- model.add(Flatten())

FULLY CONNECTION:

- The next step is to use the vector we obtained above as the input for the neural network by using the Dense function in keras. The first parameter is output which is the number of nodes in the hidden layer. You can determine the most appropriate number though experimentation. The higher the number of dimensions the more computing resources you will need to fit the model. A common practice is to pick the number of nodes in powers of two.
- model.add(Dense(256,activation='relu'))
- The next layer we have to add is the output layer. In this case , we will use the Softmax function since we expect more than two outcomes. If we expect binary outcome , we would use the sigmoid activation function.

5.2.2 Testing

In testing phase we test the generated model with the remaining portion of the dataset. The data set is fed into the generated model and their results are recorded for the next stage which is the model evaluation and error analysis.

5.2.3 Model evaluation and error analysis:

The results of the testing data along with their original values are used for the error calculation. Various stastical measures can be adopted for calculating the efficiency of the model. The various measures are accuracy , precision , recall etc. If the results of these quantitative analysis are acceptable , then we move forward with the generated model. If the results are poor ,the model is to be regenerated with more feature so that the best is obtained.

The second part is to build the user interface.The user interface is buld using HTML and Python. This is the part of the project which deals with the user. The input is fed into the server through this. And the results returned are also displayed in the user program. The interface is buit in a way such that it is easy and understandable for the person who uses it. For that we uses responsible HTML designs which uses Bootstrap , CSS and JavaScript also to provide the better user experience.

5.3 Data Flow Diagram

DFD is one of the graphical representation techniques used in a project to show the flow of the data through a project. DFD helps us to obtain an idea about the input, output, and process involved. The things absent in a DFD are control flow, decision rules, and loops. It can be described as a representation of functions, processes that capture, manipulate, store, and distribute data between a system and the surrounding and between the components of the system. The visual representation helps for good communication.

It shows the journey of the data and how will it be stored in the last. It does not provide details about the process timings or if the process shall have a parallel or sequential operation. It is very different from a traditional flow chart or a UML that shows the control flow or the data flow.

In level 0 the basic data flow of the application is showcased. It does not show the flow of data much deeper. It will be evaluated in the higher levels of Data Flow Diagram. The Data Flow Diagram of Lung cancer detection system is shown below.

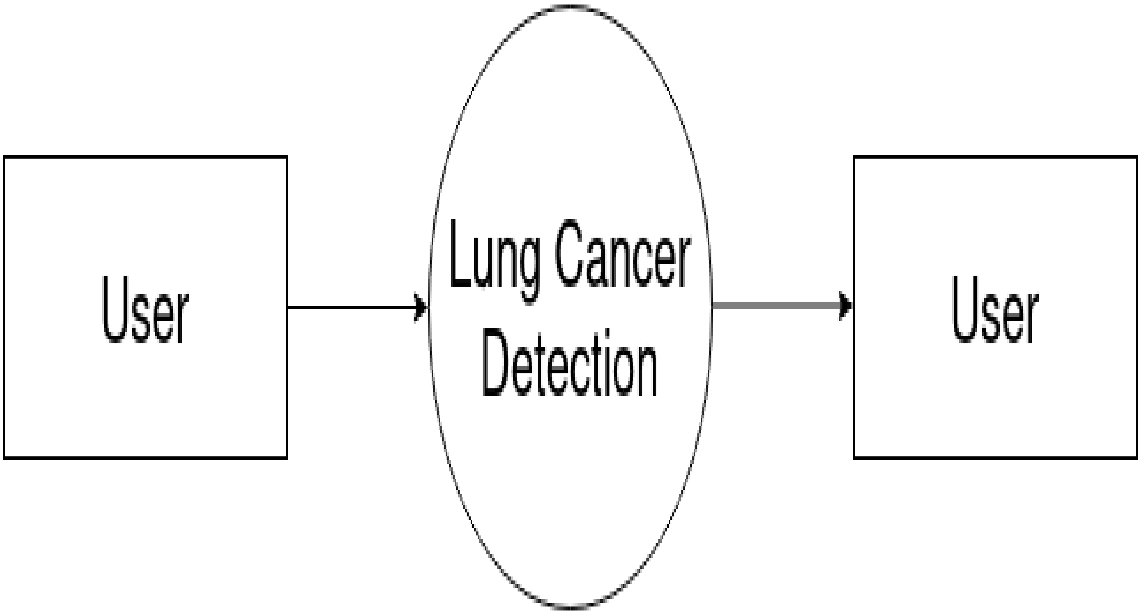
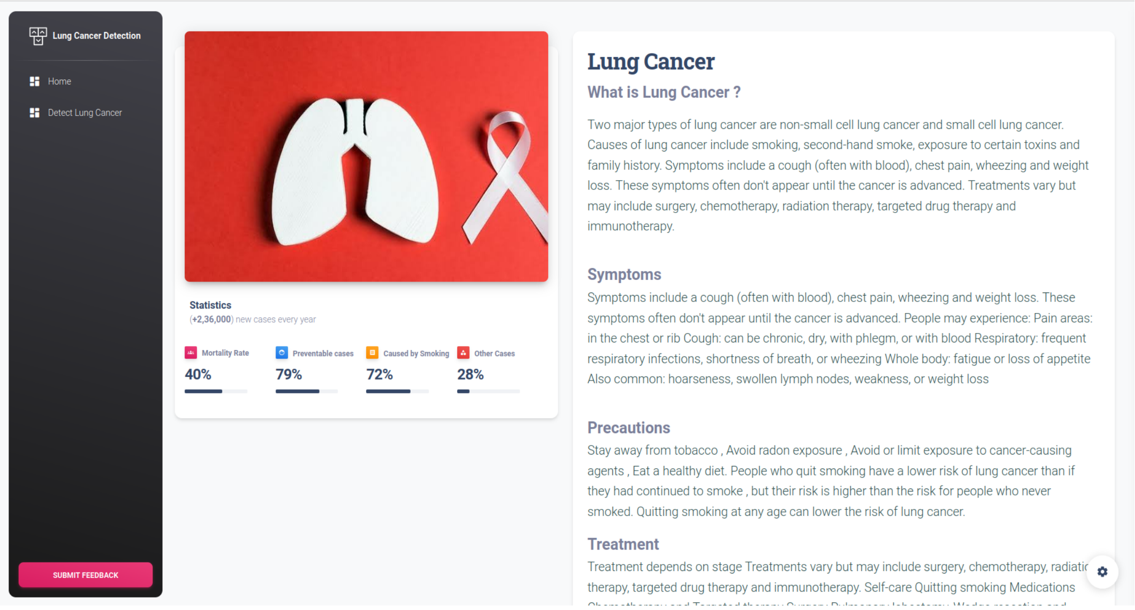


Figure 5.1: Level 0 DFD

The diagram shows Level 0 Data flow diagram of the Lung cancer detection System. As the diagram indicates there is a user part and an admin one. The input of the project is the essay by the user and which is given to the Automated essay scoring system. Then the essay is passed to the admin part for the evaluation of the essay. The evaluation of essay and score calculation is occurred in the admin side. The value of the evaluation is passed back to the user through the application.

5.4 Screenshots of user interface



Statistics

(+2,36,000) new cases every year

Mortality Rate	Preventable cases	Caused by Smoking	Other Cases
40%	79%	72%	28%

Lung Cancer

What is Lung Cancer ?

Two major types of lung cancer are non-small cell lung cancer and small cell lung cancer. Causes of lung cancer include smoking, second-hand smoke, exposure to certain toxins and family history. Symptoms include a cough (often with blood), chest pain, wheezing and weight loss. These symptoms often don't appear until the cancer is advanced. Treatments vary but may include surgery, chemotherapy, radiation therapy, targeted drug therapy and immunotherapy.

Symptoms

Symptoms include a cough (often with blood), chest pain, wheezing and weight loss. These symptoms often don't appear until the cancer is advanced. People may experience: Pain areas: in the chest or rib Cough: can be chronic, dry, with phlegm, or with blood Respiratory: frequent respiratory infections, shortness of breath, or wheezing Whole body: fatigue or loss of appetite Also common: hoarseness, swollen lymph nodes, weakness, or weight loss

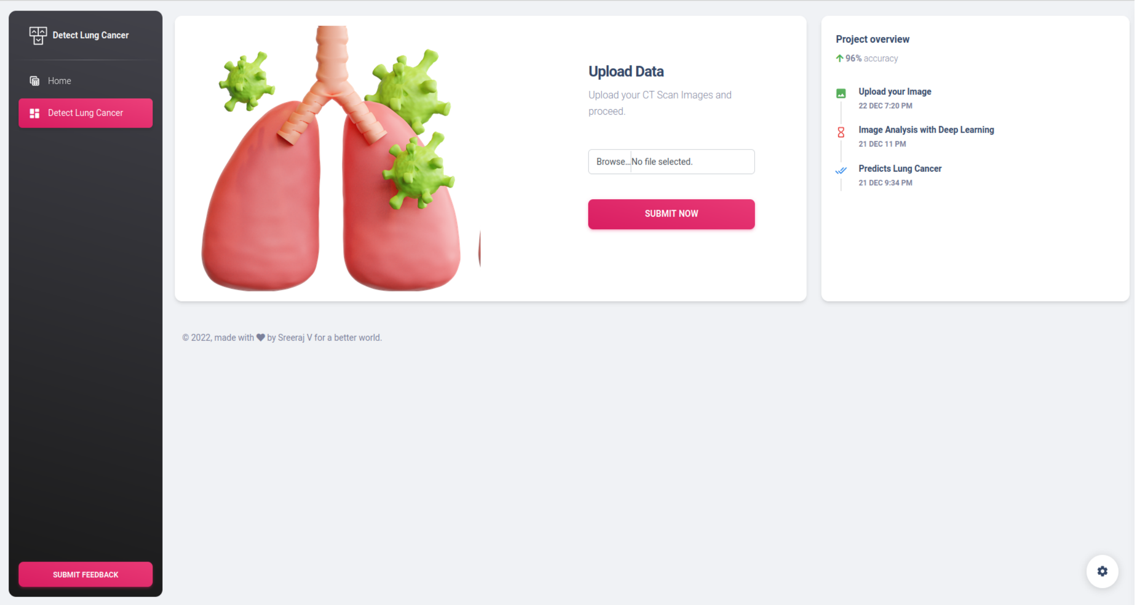
Precautions

Stay away from tobacco , Avoid radon exposure , Avoid or limit exposure to cancer-causing agents , Eat a healthy diet. People who quit smoking have a lower risk of lung cancer than if they had continued to smoke , but their risk is higher than the risk for people who never smoked. Quitting smoking at any age can lower the risk of lung cancer.

Treatment

Treatment depends on stage Treatments vary but may include surgery, chemotherapy, radiatic therapy, targeted drug therapy and immunotherapy. Self-care Quitting smoking Medications Chemotherapy and Targeted therapy Surgery Pulmonary lobectomy Wedge resection and

Figure 5.2: Home page



Upload Data

Upload your CT Scan Images and proceed.

No file selected.

Project overview

96% accuracy

Upload your Image
22 DEC 7:20 PM

Image Analysis with Deep Learning
21 DEC 11 PM

Predicts Lung Cancer
21 DEC 9:34 PM

© 2022, made with ♥ by Sreeraj V for a better world.

Figure 5.3: Detection page

Chapter 6

Sample codes and results

6.1 importing libraries

import os

import cv2

import time

import shutil

import itertools

import numpy as np

import pandas as pd

import seaborn as sns

import tensorflow as tf

sns.set$_styledarkgrid')$

$fromtensorflowimportkeras$

$importmatplotlib.pyplotasplt$

$fromtensorflow.kerasimportregularizers$

$fromtensorflow.keras.optimizersimportAdam, Adamax$

$fromsklearn.model_selectionimporttrain_test_split$

$fromtensorflow.keras.metricsimportcategorical_crossentropy$

$fromtensorflow.keras.modelsimportModel, load_model, Sequential$

$fromsklearn.metricsimportconfusion_matrix, classification_report$

$fromtensorflow.keras.preprocessing.imageimportImageDataGenerator$

$fromtensorflow.keras.layersimportConv2D, MaxPooling2D, Flatten, Dense, Activation, Dropout, B$

6.2 Preprocessing

img$_{s}ize = (224, 224)$

$channels = 3$

$img_{s}hape = (img_{s}ize[0], img_{s}ize[1], channels)$

$ts_{l}ength = len(test_{d}f)$

$test_{b}atch_{s}ize = test_{b}atch_{s}ize = max(sorted([ts_{l}ength//nforninrange(1, ts_{l}ength+1)ifts_{l}engthn ==$

$0andts_{l}ength/n <= 80]))$

$test_{s}teps = ts_{l}ength//test_{b}atch_{s}ize$

$defscalar(img):$

$returnimg$

$tr_{g}en = ImageDataGenerator(preprocessing_{f}unction = scalar, horizontal_{f}lip = True)$

$ts_{g}en = ImageDataGenerator(preprocessing_{f}unction = scalar)$

$train_{g}en = tr_{g}en.flow_{f}rom_{d}ataframe(train_{d}f, x_{c}ol =' filepaths', y_{c}ol =' labels', target_{s}ize = img_{s}ize, class_{m}ode =' categorical', color_{m}ode =' rgb', shuffle = True, batch_{s}ize = batch_{s}ize)$

$valid_{g}en = ts_{g}en.flow_{f}rom_{d}ataframe(valid_{d}f, x_{c}ol =' filepaths', y_{c}ol =' labels', target_{s}ize = img_{s}ize, class_{m}ode =' categorical', color_{m}ode =' rgb', shuffle = True, batch_{s}ize = batch_{s}ize)$

$test_{g}en = ts_{g}en.flow_{f}rom_{d}ataframe(test_{d}f, x_{c}ol =' filepaths', y_{c}ol =' labels', target_{s}ize = img_{s}ize, class_{m}ode =' categorical', color_{m}ode =' rgb', shuffle = False, batch_{s}ize = test_{b}atch_{s}ize)$

$returntrain_{g}en, valid_{g}en, test_{g}en$

6.3 Training

img$_{s}ize = (224, 224)$

$channels = 3$

$img_{s}hape = (img_{s}ize[0], img_{s}ize[1], channels)$

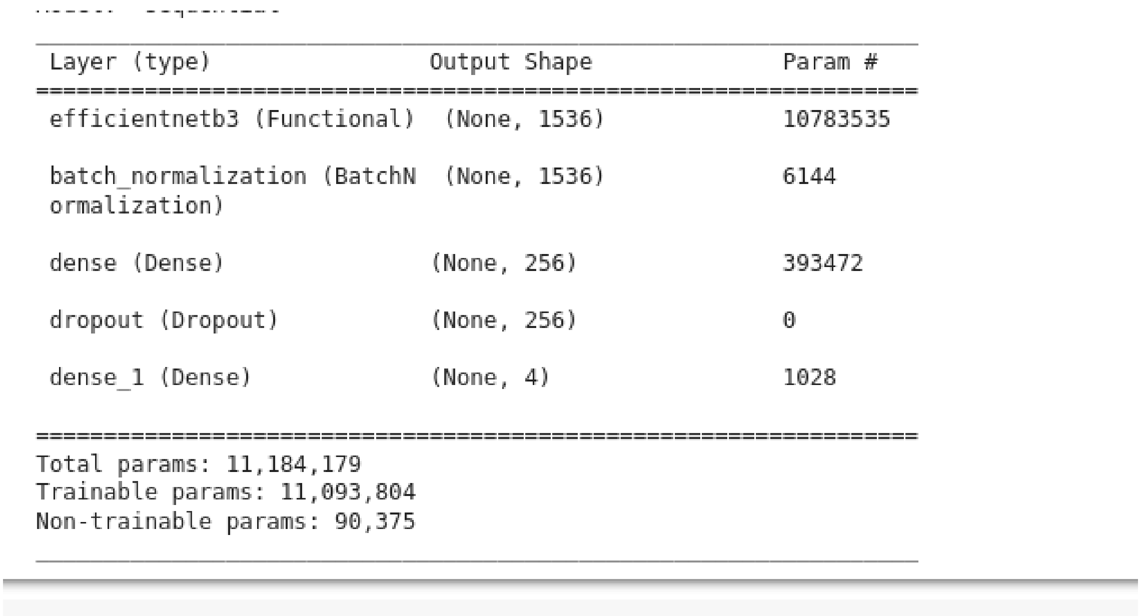
$class_{c}ount = len(list(train_{g}en.class_{i}ndices.keys()))$

$base_{m}odel = tf.keras.applications.efficientnet.EfficientNetB3(include_{t}op = False, weights =$
$"imagenet", input_{s}hape = img_{s}hape, pooling =' max')$

$model = Sequential([base_{m}odel, BatchNormalization(axis = -1, momentum = 0.99, epsilon =$
$0.001), Dense(256, kernel_{r}egularizer = regularizers.l2(l = 0.016), activity_{r}egularizer = regularizers.l$
$regularizers.l1(0.006), activation =' relu'), Dropout(rate = 0.45, seed = 123), Dense(class_{c}ount, activat$
$softmax')$

$model.compile(Adamax(learning_{r}ate = 0.001), loss =' categorical_{c}rossentropy', metrics = ['accuracy'])$

$model.summary()$



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 efficientnetb3 (Functional)  (None, 1536)             10783535

 batch_normalization (BatchN  (None, 1536)             6144
 ormalization)

 dense (Dense)               (None, 256)               393472

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 4)                 1028

=================================================================
Total params: 11,184,179
Trainable params: 11,093,804
Non-trainable params: 90,375
_________________________________________________________________
```

Figure 6.1: model summary

batch$_{s}ize = 40$

$epochs = 100$

$patience = 1$

$stop_{p}atience = 3$

$threshold = 0.9$

$factor = 0.5$

$dwell = True$

$freeze = False$

$ask_{e}poch = 5$

$batches = int(np.ceil(len(train_{g}en.labels)/batch_{s}ize))$

$callbacks = [MyCallback(model = model, base_{m}odel = base_{m}odel, patience = patience, stop_{p}atience =$

$stop_patience, threshold = threshold, factor = factor, dwell = dwell, batches = batches, initial_epoch =$
$0, epochs = epochs, ask_epoch = ask_epoch)]$
$history = model.fit(x = train_gen, epochs = epochs, verbose = 0, callbacks = callbacks, validation_data =$
$valid_gen, validation_steps = None, shuffle = False, initial_epoch = 0)$
$plot_training(history)$

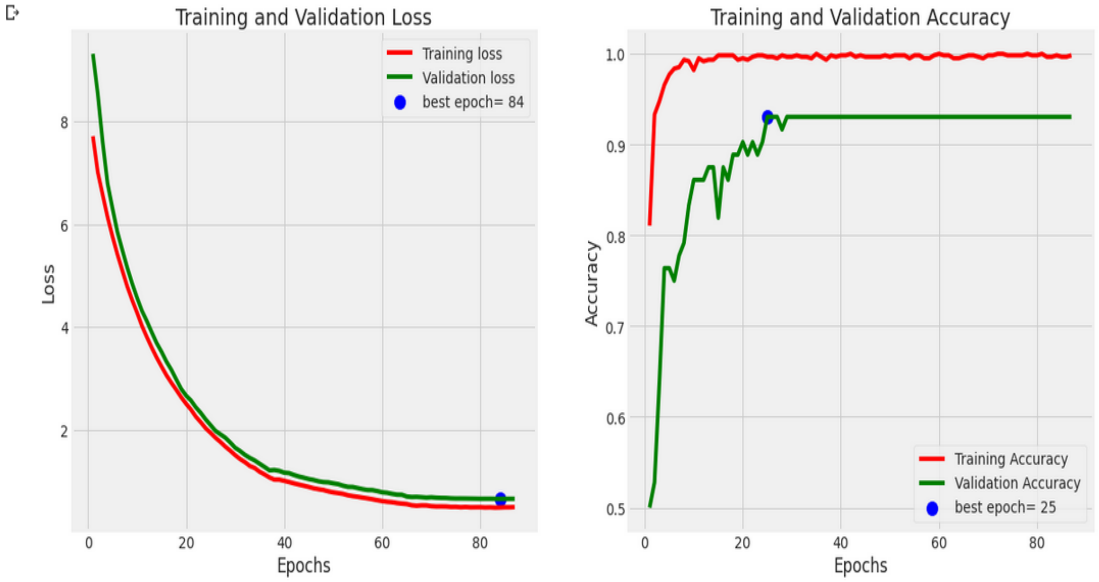


Figure 6.2: training progress-loss and accuracy

Chapter 7

Testing and Implementation

7.1 Testing and various types of testing used.

Once a software is developed, the major activity is to test whether the actual results match with the experimental results. This process is called testing. It's used to make sure that the developed system is defect free. The main aim of testing is to find the errors and missing operations by executing the program. It also ensure that all of the objectivs of the project are met by the developer. The objective of testing is not only to evaluate the bugs in the created software but also finding the ways to improve the efficiency, usability and accuracy of it. It aims to measure the functionality, specification and performance of a software program. Tests are performed on the created software and their results are compared with the expected documentation. When there are too much errors occurred, debugging is performed. And the result after debugging is tested again to make sure that the software is error free. The major testing processes applied to this project are unit testing, integration testing and system testing. In unit testing, our aim is to test all individual units of the software. It makes sure that all of the units of the software works as it intended. In integration testing, the combined individual units are tested to check whether it met the intended function or not. It helps us to find out the faults that may arise when the units are combined. In system testing the entire software is tested to make sure that it satisfies all of the requirements. The tables shown below describes the testing process occurred during the development of this project "Automated essay scoring". This defines the various steps took to create the project error free.

7.1.1 Unit Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	selected CT Scan image	Same as expected	Pass	
2	pre-processing of images	Each image pre-processed and corresponding file created.	same as expected	Pass
3	Deteection of images	Train data and detect it. Correct output obtained	csv file generated	Pass

Table 7.1: Unit test cases and results

7.1.2 Integration Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	Integration of images and Deep learning modules	The recorded data should be directly feeded in to the tensorflow module.	Same as expected	Pass
2	The command , context and values should be forwarded to the module to insert code in the project.	Respective code should be inserted into the project to implement the desired feature.	Same as expected	Pass

Table 7.2: Integration cases and result

7.1.3 System Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	Model outputs the correct result based on the input.	The output of the model should be based on the image detection which is collected from dataset.	Same as expected	Pass

Table 7.3: System test cases and results

Chapter 8

Results and Discussion

The main aim of the project was to develop an automated system for classification of lung cancers. The system can be used by onchologists and health care specialists. And it is observed that the system performs all the functionalities as expected.

8.1 Advantages and Limitations

The proposed system is a deep neural network model to predict lung cancers. The proposed system save a huge amount of time. Like every other system , this system also have it's own disadvantages. But they are negligible while comparing with the advantages and they can be overcome in future.

8.1.1 Advantages

- Can save the time needed for the prediction of cancer. And this can be spend more pructive by the doctors.
- The Human resource needed for the prediction can be saved.
- The mood or mental situation of the doctor no longer depend the accuracy of the prediction he done.
- The system predict which type of cancer is present, that is adenocarcinoma, large.cell.carcinoma, normal,squamous.cell.carcinoma.

- Since the process of evaluation is automatic, the patients can check their result soon after finishing.
- prediction with high accuracy.

8.1.2 Limitations

- The current dataset is comparatively small. It can be improved by improving dataset
- The feature extraction process can be found as slow. but it can be improved using high performing systems.

Chapter 9

Conclusion and Future Scope

Our system architecture is proposed for adenocarcinoma, large cell carcinoma and squamous cell carcinoma lung cancers detection with an objective of high classification accuracy within a short time. First, a proper lung cancer dataset for efficiently performing the training and testing process. Second a training strategy includes training our model on the desirable pattern from scratch. Third we hired our model to extract CT Scan images features and efficiently classify them. We evaluate the proposed model on dataset with 400 CT Scan images. The proposed model accomplished high accuracy.

In the future , we are going to increase CT Scan images in the used dataset to improve the accuracy of the proposed model. The overall accuracy of the system can be improved using 3D convolutional neural network and also by improving the hidden neurons with deep network.

Bibliography

[1] [1412.6980v8] Adam: A Method for Stochastic Optimization, . URL https://arxiv.org/abs/1412. 6980v8.

[2] Analyzing The Papers Behind Facebook's Computer Vision Approach Adit Deshpande CS Undergrad at UCLA ('19), . URL https://adeshpande3.github.io/Analyzing-the-Papers-Behind-Facebook27s-Computer-Vision-Approach/.

[3] Computer Vision - deeplearning.ai, . URL https://www.coursera.org/learn/ convolutional-neural-networks/lecture/Ob1nR/computer-vision.

[4] Data Science Bowl 2017 — Kaggle, . URL https://www.kaggle.com/c/data-science-bowl-2017/ kernels.

[5] FloydHub - Deep Learning Platform - Cloud GPU, . URL https://www.floydhub.com/.

[6] Neural Network Foundations, Explained: Activation Function, . URL https://www.kdnuggets.com/ 2017/09/neural-network-foundations-explained-activation-function.html.

[7] M.S. Al-Tarawneh, "Lung cancer detection using image processing techniques," Leonardo Electronic Journal of Practices and Technologies, vol. 20, pp. 147– 58, May 2012.