Lung Cancer Detection Using Deep Learning

Azib Farooq

18-EE-43

Sabahat Tabbassum

18-EE-01

Syed Irtiza Hussain

18-EE-46

Javeria Noor Tariq

18-EE-53

Dr. Gulistan Raja

Professor

DEPARTMENT OF ELECTRICAL ENGINEERING

FACULTY OF ELECTRONICS & ELECTRICAL ENGINEERING

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

TAXILA

July 2022

Lung Cancer Detection Using Deep Learning

Azib Farooq

18-EE-43

Sabahat Tabbassum

18-EE-01

Syed Irtiza Hussain

18-EE-46

Javeria Noor Tariq

18-EE-53

A thesis submitted in partial fulfillment of the requirements for the degree of

B.Sc. Electrical Engineering

Supervisor:

Dr. Gulistan Raja

Professor, EED

External Examiner Signature:____

Thesis Supervisor Signature: ____

DEPARTMENT OF ELECTRICAL ENGINEERING

FACULTY OF ELECTRONICS & ELECTRICAL ENGINEERING

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

July 2022

Undertaking

I certify that SDP work titled "*Lung Cancer Detection using Deep Learning*" is my own work. The work has not, in whole or in part, been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged/referred.

Signature of Student

Azib Farooq

18-EE-43

Signature of Student

Sabahat Tabbassum

18-EE-01

Signature of Student

Syed Irtaza Hussain

18-EE-46

Signature of Student

Javeria Noor Tariq

18-EE-53

Dedication

The work is dedicated to Electrical Engineering Department of UET, Taxila.

Abstract

Lung Cancer Detection Using Deep Learning

Azib Farooq, Sabahat Tabassum, Syed Irtiza, Javeria Noor Tariq

18-EE-43, 18-EE-01, 18-EE-46, 18-EE-53

Thesis Supervisor: Dr. Gulistan Raja

Professor, UET Taxila

Lung cancer is one of the leading cause of deaths around the globe. Early detection and treatment is crucial for patient recovery. Lung cancer detection and classification is both challenging and prone to human error. This project has designed and implemented double stand approach for classification and detection of lung cancer. For classification task, pre-trained 50 layer Residual Network is used, fine-tuned for three class classification, and the model is trained on LC25000 dataset for lung cancer images, containing three classes i.e. Adenocarcinoma, Squamous Cell Carcinoma and Benign. Classification model is able to achieve 98% accuracy on test data. The lung cancer detection is achieved by semantic segmentation using U-Net and ACDC Lung Cancer Grand Challenge dataset is used to train the model. Detection model achieves Mean IOU score of 33% on training data and 31% on validation data. Trained model integrated Graphical User Interface is delivered in the form of mobile and web application.

Keywords: Histopathology, Classification, Detection, Lung Cancer, Big Data.

Acknowledgements

This works acknowledges the immense and sincere efforts of:

- *Dr. Gulistan Raja, Professor UET Taxila.*
- *Dr. Furqan shoukat, Assistant Professor UET Taxila.*
- *Dr. Junaid Mir, Assistant Professor UET Taxila.*
- *Maj. Mansoor, PhD student UET Taxila.*
- *Maj. Haider, PhD student NUST Islamabad.*
- *Muhammad Shaban, PostDoc Fellow Harvard USA.*

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENT

CHAPTER 1
Introduction

Lung cancer is the eminent origin of mortality for men as well as women across the globe. According to Worldwide Cancer Research Fund statistics, in 2018 lung cancer is detected in about 2.07 million people out of which 1.86 million people have died [1]. The key to the survival rate is its early detection. If cancer is identified in its initial stages, death rate can be reduced. But the problem is that the most of the patients are explicitly diagnosed during the intervening or final stage of lung cancer, making the risk of deaths increased. In the process of diagnosing cancerous lung lumps, the pathologists have to detect an abnormal growth of lung cells and categorize its types among Benign, Squamous Cell Carcinoma and Adenocarcinoma. Diagnosing lung cancer is an onerous process, but pathologists bear a high level of reliability in diagnosis. According to studies, the detection accuracy rate of the pathologists in the identification and classification of lung cancer using histopathological images divulged to about 0.75%. As a result, doctors face difficulties in diagnosing cancer.

1.0. Global Statistics

According to GLOBOCAN 2020, it was estimated that among 2.2 million new cancer cases and 1.8 million deaths, lung cancer is the second most diagnosed cancer. It is the leading cause of cancer death in 2020, representing approximately one in 10 (11.4%) cancers diagnosed and one in 5 (18.0%) deaths.

According to GLOBOCAN 2020, Lung cancer remained the leading cause of cancer death, with an estimated 1.8 million deaths (18%). Early detection and treatment are crucial for patient recovery. Smoking is a known cause of lung cancer. Next are indirect smoking, radon gas, air pollution, and genetics.

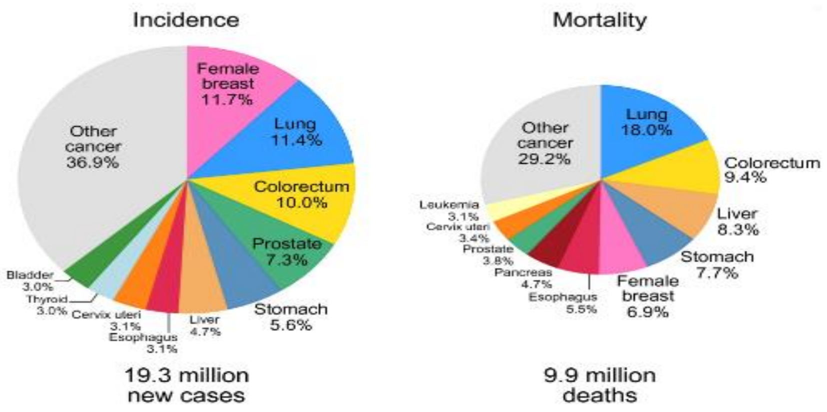


Figure 1.0.1 Division of death and events among the Most Common Cancers in 2020 *[1]*

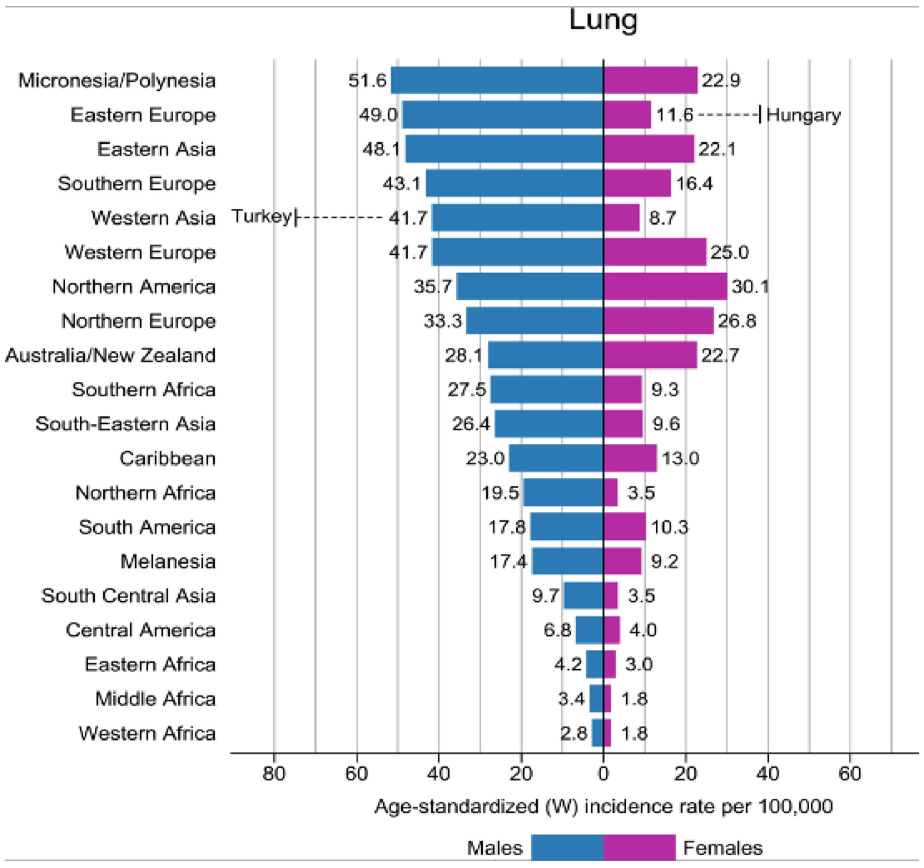


Figure 1.0.2.: Gender Wise Cancer distribution around globe *[1]*

Our body is made of millions of cells. These cells, which are basic unit of human life develop in our body and after specific time these cells destroy and new cells are formed within our body.

Cancer occurs when this normal process doesn't happen and instead of it cells start to grow in abnormal way. As a result, outgrowth of cells occur in body which is called as tumors. These tumors can infect the tissues close to them and can spread within the body either through the lymph node or blood, if not treated timely. That's why tumors are called malignant. Benign is the mildest among all the types of cancer as it's not malignant. It doesn't spread through the body and doesn't attack the nearby tissues. When Benign is treated by surgery or any other medical procedure, it doesn't grow back again. But when a malignant tumor is treated, there are chances that it can develop again. Normally Benign tumors are not dangerous excluding the benign brain tumor. It is life-threatening and can lead to death.

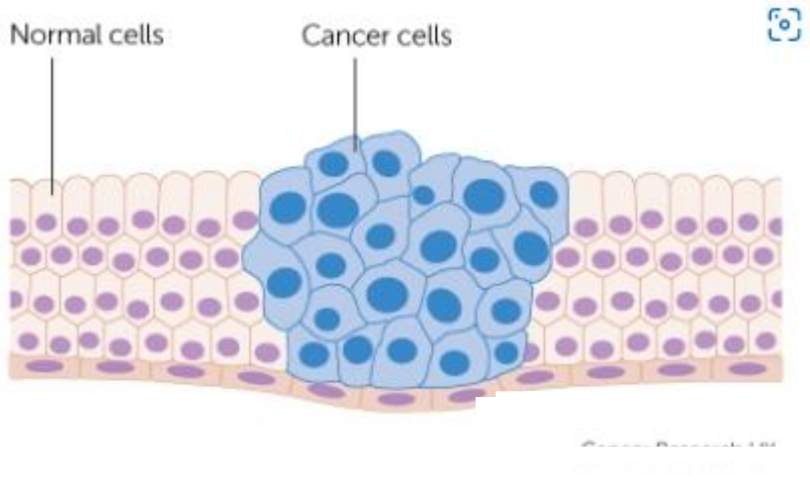


Figure 1.0.3.: Abnormal Cell Leading to Cancer *[2]*

Lung Cancer occurs when there is abnormal development of cells lungs. These unusual cells do not function properly and do not mature into normal cells as healthy cells are expected to do. As we know, the pivotal function of the lungs is to endow oxygen to the entire body via the blood. But when cancer occurs in the lungs, the supply of oxygen gets disturbed. Although the medical field has made many advances in treatment procedures still lung cancer which is at the early stages or final stages is not easily treatable.

1.1. Lung Cancer Classification

Lung Cancer is classified into two categories according to its visual appearance in the microscopic image.

1. Small Cell Lung Cancer (SCLC)
2. Non-Small Cell Lung Cancer (NSCLC)

Both of these types differ from each other on the basis of their growth and their treatment procedures also vary, that's why it is of crucial importance to correctly identify the category of Lung cancer among the two mentioned above.

1.2. Project Overview

1.2.1. Lung Cancer Applications

We have integrated the deep learning model into mobile app as well as web app for the detection as well as classification purpose. For the purpose of the mobile app we used software development kit named flutter using Android Studio in which dart language is used for coding. For the development of web app we used Flask in PyCharm using Python language. In these applications, the user needs to upload histopathological image. The uploaded image is then inferred to the lung cancer detection deep learning model trained on dataset of Kaggle LC25000, which analyze the image, identify the type and displays the results on the screen.

1.2.2. Dataset

This thesis report contains many different factors that assists deep learning's ability to find lung cancer inside Histopathological images. The data used to train the model for the classification was

Kaggle LC25000 dataset. This dataset contains 25000 Histopathological images of size 768x768 pixels in jpeg. There are a total of 750 images of lung tissues (250 of each type). Out of 25000 images, 10000 images are of colon cancer and 15000 of lung cancer. While training the model, first we removed the colon images and used only lung cancer images

For localization, we used ACDC (Automated Cancer Detection and Classification) lung cancer grand challenge dataset which are histopathological images that were annotated by pathologists. The pathologists determine the type of cancer. Research on Deep Learning research was done to choose the correct architecture for the training of the model. For classification, RESNET 50 and for localization U-Net architecture are being used.

1.3. Challenges

In this section, we will discuss a few challenges that were faced during this project.

1.3.1. Large Dataset Management

Kaggle LC25000 dataset was easy to deal with as it was approximately 2GB so the processing and training process was not time taking but the ACDC dataset is very large (about 200 GB) which is difficult to manage and the training process is the time taken. A useful solution to this is to relocate data to a cloud service for efficient workflow.

1.3.2. Choosing the Neural Network Architecture

There are a lot of different Neural Network Architecture out there. The main challenge is to choose the architecture which is best suited according to the dataset being used and according to the desired results. To deal with this challenge, it is important that we choose the correct model to implement before training.

1.3.3. Model Training

Kaggle LC25000 dataset is trained using Google Colab as Fast GPU is available on it and training does not consume much time. As the ACDC dataset is large, training its deep learning model consumes a lot of time. It took almost one week for the training of the dataset in our model and an extra hard drive to process the dataset. Careful measures before training can greatly reduce the problem.

1.3.4. Model Deployment

For integration in a mobile app, the TensorFlow Lite model is required. But the issue was flite model command went deprecated in the new version of Android Studio and the alternate method was difficult to implement so we use another method. At first, we created a Web App on python using PyCharm, integrated that into the azure portal, got the website URL, and used this URL link in the flutter mobile app. This app works by taking histopathological images from the user and then uploading it to the server, and in the end, the results are displayed.

1.4. Aims and Objectives

The Aims and Objectives of our thesis are:

- Identifying the type of lung cancer among Benign, Squamous Cell Carcinoma, and Adenocarcinoma using Reset 50 Architecture.
- Identifying the location of the cancerous area in the histopathological image using U-Net Architecture.

The thesis aims at:

1. The development of a deep learning model capable of classification and localization of the tumor in lungs.

2. To integrate the model into mobile apps and web apps.
3. To provide Graphical User Interface Applications so that doctors can easily access the application and can diagnose lung cancer in the minimum possible time.

1.5. Ethical Issues

Medical ethics has a closed relationship with law. Ethical principles such as respect for the persons, informed consent and confidentiality are basic to the patient-physician relationship.

Autonomy:

Patient has freedom of thought, intention and action. Patient should know all the risks, benefits of the procedure and likelihood of success before making the decision.

Beneficence:

The main aim of the procedure is to do good to the patient. Considerate the patient's welfare.

Confidentiality:

Personal, medical and treatment information should be kept confidential. If the information is crucial for the patient, only in that case it can be revealed.

Non-Maleficence:

Making sure that the procedure doesn't harm the patient or others in the society.

Justice and Equity:

Fair and equal distribution of scarce health resources and decision of who gets what treatment.

1.5.1. Scientific digital image acquisition and manipulation guidelines

1. Scientific digital images are data that can be compromised by inappropriate deceptions or manipulations.

2. Digital image manipulation should always be done in the data copy of the unprocessed image.
3. Simple adjustments throughout an image are generally acceptable.
4. Use of software filters to improve image quality is usually not recommended for biological images.
5. Manipulations that are specific to one area of an image and are not performed on other areas are questionable.
6. Comparable digital images should be obtained under the same conditions, and any post-acquisition image processing should be the same.
7. Cloning or copying objects into a digital image from other parts of the same image or from a different image, is very questionable.
8. Avoiding use of lossy compression.
9. Intensity measurements should be performed on uniformly processed image data, and the data should be calibrated to a known standard.
10. Image cropping is acceptable.

1.7. Thesis Organization

Chapter 1 of the thesis contains a basic introduction to lung cancer and its difficulty in diagnosis, a brief overview of cancer, lung cancer, and its types. In this chapter, we also presented a brief overview of our project, the applications (mobile and web) that we developed, the dataset we used, and the challenges that we had to face during developing the deep learning model. Chapter 2 contains the literature review and all the relevant theoretical concepts that are necessary to understand for this project. In Chapter 3, we discussed the methodology that we followed for developing the model, the dataset description, the description of architectures that we used, and the whole procedure of flutter mobile app development as well as web app. In Chapter 4, we presented the results and relevant case studies. Chapter 5 contains the conclusion of our work. In Chapter 6, we presented the suggestions for the future work that can be further done on this project.

CHAPTER 2

Literature Review

2.0. Research epitome

In this chapter, the author describes the research work carried out. This chapter summarizes related studies from a variety of sources, demonstrating knowledge in the medical field, especially in the detection of lung cancer. This chapter also includes research on the data set we used and technology.

2.1. Technical Modality

In this part, we will discuss about the technologies we used in our project. We will discuss that why these tools are fundamental for our project.

2.1.1. Python

Python is the fastest going language in terms developers, libraries and applications where it can be used, it could be machine learning, artificial intelligence and web development. It is the general-purpose high-level programming language which is easy to learn and dynamically initialized. Its easy syntax and high-level features makes it best programming language to get started off. It is an open-source programming language and its free to use. Python has a great community who constantly make libraries. Python is best suited for machine learning algorithms and here in our project we will be using Python.

2.1.2. Tensor Flow

Google's Tensor Flow ease the process of acquiring data, training models, solving predictions and refining future results. It is an open source google library for numerical computation and large-scale machine learning. Tensor flow bundles together a study of machine learning and deep learning models to make them useful. It offers multiple level of abstractions and train models by using the high level Kira's API.

2.1.3. Keras

It is python-based deep learning framework that is in actual a high level api of tensor flow. It is an open source and actively developed by all the contributors across the globe. It runs on the top of Tensor flow, Theano and CNTK. Its building models are as simple as stacking layers and later connecting these graphs. It is an API which is actually used to specify and train differentiable programs high performance naturally flows through it.

2.1.4. Anaconda

Anaconda is an open source dispensation of python and R. It is used for machine learning, deep learning, data science and many more. With over three hundred data science libraries, it becomes fairly optimal for any programmer to work on data science. Anaconda also helps in simplified package management and deployment. Anaconda comes with wide variety of tools, it easily collects data from various sources using machine learning/AI algorithm. It also helps in getting an easy manageable environment setup which can deploy on any project with a click of single button.

2.1.5. NumPy

NumPy is basically a module or pith library that is available in python for scientific computing. It contains a powerful n dimensional array object, tools for integrating with C++. It is very useful to perform Fourier transform, random number capabilities and linear algebra. NumPy can also be used as a coherent multi-dimensional container for generic data.

In our project we are using Numpy instead of lists because of the following reasons:

- It occupies less memory than lists.

- It is relatively faster than the lists.
- It is very easy to work with NumPy as compared to lists.

2.1.6. Data Visualization

It is easy for human brain to process the information in pictorial or graphical form. It helps us to understand the trend better and help in taking better decision. Data visualization allows us to immediately explicate the data just by looking at the trend of the graph. Graphical representation will give us a very a good idea that which variables are useful and which variables are useless. For efficient and quick results of an experiment we use data visualization.

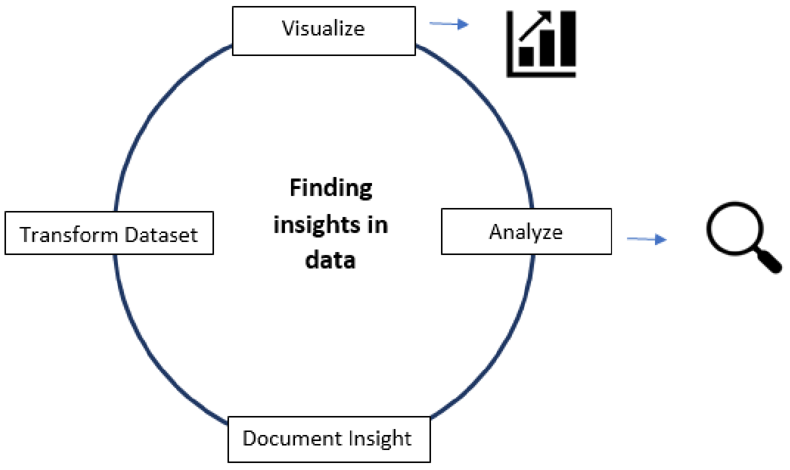


Figure 2.2.7.1.: Data Visualization Flow

2.1.7. Matlplotlib

Python has a plotting library named as Matplotlib, it allows the programmers to design a large category of graphs and visualization. Its biggest advantage is that it is easy to use. Also its major aspect is that it can integrate with Jupyter notebook and simplified visualization is created. It also works great with Numpy.

2.1.8. Jupyter Notebook

Jupyter is a modern d2 that allows data scientists to record their complete analysis process much in the same way other scientists used a lab notebook. Jupyter product was originally developed as a part of the ipython project. Ipython project was used to provide interactive online access to python over the time it became useful to interact with other data analysis tools such as R. Ipython is still an active tool that's available for use the name, jupyter itself is derived from the combination of Giulia, Python and R. Jupyter runs code in many programming languages. Python is the requirement for installing the jupyter notebook itself

2.1.9. Azure Web Services

Azure basically is a cloud computing platform provided by Microsoft. Primarily it is an online portal through which you can access, manage resources and services. We can store our data and can transform the data using services that Microsoft provides. Azure is free to start and has a pay-per-use model (need to pay the services we use through azure). Azure has 42 regions which is more than any cloud service provider has. Azure offers services that cover computation, machine learning, integration, artificial intelligence, Iot, security, developer tools and databases. It helps to create a virtual machine of linux or window operating system. Its very easily configurable. We can add RAM or storage and also we can remove RAM or storage.

By using the cloud service we can create an application within the cloud and all of the work after we deployed is taken care of by azure. It includes provisioning the application, load balancing, ensuring that the application is in good health and all of the other things are handled by azure.

2.1.10. File Handling

There is a group of data and information in a file. Data can be updated and shared through files with multiple stakeholders. File handling in python performs the functions which includes creating, reading, updating and deleting. By using the file handling operators and functions, we can perform general crud operations. File operation in Python takes place in the following order

- Foremost step is to open a file
- Perform Write or Read operation
- Close the file

There are two files in Python, one is text file and other is binary file. Text file is used for storing data in the form of characters and string, whereas binary file is used for storing data in the form of bytes. Some of the basic advantages of file handling are as follows:

- Stored data is permanent, primarily it resides in the file and then it is up to us how do we save the data in the operating system.
- Stored data can be shared, certainly we can mail the data, we can share it in our hard drive or over a network.
- We can update the data in the file. It means we can override, append, remove and delete the file.

2.1.11. Open slide

Whole slide image is a large image that contains hundreds of thousands of pixels in x and y direction. We cannot load that image into our memory and process it. So, for that we will be using open slide library. It reads virtual slides in several formats and svs is the most common format for it. Other formats includes tip, leica mira, hamamtsu and x Philips.

2.1.12. Exception Handling

Exception Handling is actually a process of respond to the occurrence of exception. It happens during the process of compilation of exceptional statements that requires special processing. This is basically done because of the change in the normal flow of program execution. For the process of exception handling, we will use the following keywords:

- Try: it is basically a keyword that is used to keep the code segments under check. So, the code we put under try is code under scrutiny.
- Except: Except is a segment which is actually used to handle the exception.
- Else: It runs when no exception exists.
- Finally: It will run either if there is exception or not.

2.1.13. Patchify

Patchify is the library that can be used to cut down the images into smaller patches and then it stored those patched/cropped images in NumPy array or our drive. Patchify is instated through pip install patchify. Normally image and their corresponding mask that we have annotated is very large about 1k by 1k. Its very difficult for the programmer to train them on smaller gpu's or regular system. So, we need to cut them down into 256 by 256 patches. This library can also do unpatchify, means transformed or morphed images can get back into an original image.

2.1.14. Flutter

Flutter is a mobile UI framework. It is used for designing native apps for android and IOS devices. These apps can have access to platform AP, including camera, microphone etc. Flutter makes it very easier for the programmer to design an app as it allows only single code base for both the operating devices. This single code base is created using a programming language called Dart. Dart is very easy language to learn. It uses good layout approach and make the apps responsive in terms of speed. While running the apps it will give immediate and smooth experience. It uses material design out of the box which will make our apps look appealing.

2.1.17. OpenCv

OpenCv stands for Open Source Computer Vision. It is a universal computer vision library. It is written in C++ but it can interfaced with other programming languages as well like Python, C++ and Java. This tool shows best results when working with the images. It has assorted of tools for performing various functions including image manipulation, feature extraction, sharpening of the edges etc.

2.2. Literature Review

2.2.1. Conventional Methods

Following are conventional methods used for detecting the lung cancer

a. Imaging tests: An x-ray image can be used to detect cancerous cells in lungs. CT scan can be used to detect details which is not possible with X -ray imaging.

b. Sputum cytology: Sputum produced in coughing can be analyzed under microscope to confirm lung cancer.

c. Tissue sample (biopsy): Cells sample taken in process called biopsy is analyzed under microscope. Methods for biopsy can include bronchoscopy, Mediastinoscopy, needle biopsy. Analysis of cell will confirm what type of cancer does patient have. Which will help prognosis and guide doctor to adopt the treatment.

Bronchoscopy can be used for obtaining tissue sample from the body. A lightning tube is inserted in into lungs passing form throat into lungs. The tube is used to study abnormal cells in lungs. In mediastinoscopy, a cut is made at the base of the neck, surgical tools are inserted at the back of breastbone and sample is taken from lymph nodes. In Needle Biopsy, X-ray and CT guide needle through chest wall to take suspicious cell from the lungs and analyzed under microscope.

After detection of lung cancer, we have to determine the stage of cancer that vary from 0-IV stage. Staging tests include MRI, Bone Scan, CT and PET. These tests will help the doctors to determine spread of cancerous cell and guide them to adopt method for treatment [3].

Evaluation of histopathological slides by pathologists is necessary for diagnosis and classify the type of lung cancer. For pathologist and other medical professional diagnosing process is time consuming. Due to cumbersome process, the cancer type can be detected wrongly and directing toward wrong treatment can cause risking of patient's life. To reduce the risk, machine learning algorithms can be used to detect lung cancer [4]. Some methods used to detect lung cancer are being reviewed in this section. Although we have to take the sample same as conventional methods and take images of those histopathological slides. New methods involve the analysis of these histopathological slides using machine learning algorithm.

2.2.2. Modern Approaches for detecting Lung cancer

Number of techniques are being deployed in the literature for the classification of the lung cancer through application of algorithms on images obtained from different sources like X-RAY and histopathology images. In this section, we will discuss techniques and approaches on both type of images, and also outlined the results being obtained from the techniques.

Prediction of lung cancer using chest x-ray through transfer learning is being applied by the W. Ausawalaithong et.al.(2018) Images of size 224x224 with 121-layers densely connected CNN (DenseNet-121) is used. Single sigmoid node is applied in FC layer. The result obtained by this techniques had 74.43% mean accuracy with tolerance of almost 6%. The model has the same mean sensitivity for the dataset of the different images [5]. T. Atsushi et.al. (2017) used DCNN (Deep Convolutional Neural Networks) to automate lung cancer classification on cytological images. The primary structure of the DCNN include 3 convolutional and pooling layers with 2 fully connected layers, with 0.5 dropout for the DCNN. The model has accuracy of 71.1% [6].

W. Rahane et.al. (2018) detected lung cancer on CT images using image processing and SVM (support vector machine). Primary image processing techniques applied were grayscale conversion, noise reduction and conversion to binary image. Followed by the feature extraction, which in turn fed to the SVM. The features obtained was area, perimeter, and eccentricity from the segmented image region of interest [7]. M. Saric et.al. () applied deep learning techniques of VGG and ResNet for lung cancer detection. The dataset for the architecture is the whole slide histopathology images. The output obtained from the algorithm was related using the receiver operating characteristic (ROC) plot. The patch accuracy of the VGG16 obtained was 75.41% and 72.05% for ResNet50, respectively. The author also provide justification for low accuracy as the huge pattern diversification in the dataset [8].

S. Sasikala et.al. (2019) examined CT scan images to detect and classify lung cancer using CNN. The programming tools used in this research was MATLAB. The training process was divided into two phases i.e. valuable volumetric feature extraction was the first phase followed by the classification in the next and final phase. The proposed architecture can make classification of cancerous and non-cancerous cells with 96% accuracy [9]. SRS Chakravarty et.al. () used gray level co-occurrence matrix (GLCM) along with chaotic crow search algorithm (CCSA) for the purpose of feature selection, which are being computed on the CT scan images. The extracted features are then supplied to the probabilistic neural network (PNN), which is responsible for classification. The final accuracy of the model on the features extracted from the CCSA was 90% [6].

Hatuwal et.al (2020) use convolutional neural network to detect lung cancer on histopathology images. Three types of carcinomas are considered in the study i.e. Adenocarcinoma, squamous cell carcinoma and benign carcinoma. The analysis and experimentation is based on LC25000 dataset. In which each class of the carcinoma has 5000 images. Data augmentation techniques like image flipping (horizontal, vertical), zooming, etc. is applied to increase the dataset to avoid overfitting of the model. Stacked layers of CNN (CovNets) is used for recognition and classification. The model obtain validation accuracy of 97.2% [10].

Zhu Y et.al. (2010) applied methods of texture feature extraction of solitary pulmonary nodules, followed by the application of the genetic algorithm to choose the most significant features and make classification using support vector machines [11]. S. QingZeng et.al. () used a CNN, DNN and stacked auto-encoder to classify CT images of benign and malignant classes of the lung cancer. After training the CNN achieved the accuracy of 84.15% [12]. Laksmanaprabu et.al. () designed and optimal DNN for the analysis of the CT images and extracted features from these images and classify them as malignant or benign. After training the accuracy of the model was 92.2% [13].

S. Garg et.al. (2021) utilizes the pre-trained CNN models to identify lung and colon cancer using LC25000 dataset. The dataset is also augmented using different techniques and fed to eight different pre-trained models, which are VGG16, NASNetMobile, InceptionV3, InceptionResNetV2, ResNet50, Xception, MobileNet and DenseNet169. Multiple evaluation parameters of all these models are monitored and examined i.e. precision, recall, f1score, accuracy, and auroc score. The range of results of all these pre-trained models are from 96% to 100%, on the particular dataset [14].

2.2.3 Malignancy Detection using transfer learning

Lung cancer causes high mortality among men and women. Early diagnosis will help the doctors in the treatment and the chances of survival of an individual increases. This paper proposed a systematic model for the lethal diagnosis of lung cancer [15]. For training and validation big dataset of about 25000 histopathology images is employed. These images are divided into five classes. For training eighty percent of the images in respective class is used and for testing purpose twenty percent of the images in respective class is used. For the classification Alexnet the pre-trained convolution neural network was tuned by adjusting four of its layers before training it on the dataset. To large image classification task, transfer learning is performed with deep learning models. To improve the overall accuracy and keep the model computationally efficient, instead of implementing image enhancement techniques on the entire data set, the image quality of the low performance class was improved using a contrast enhancement technique, which is relatively simple and effective. The class selective image processing has considerably increased the efficiency and overall performance of the experimental findings. This methodology has improved the overall accuracy from 88% to 98.2%.

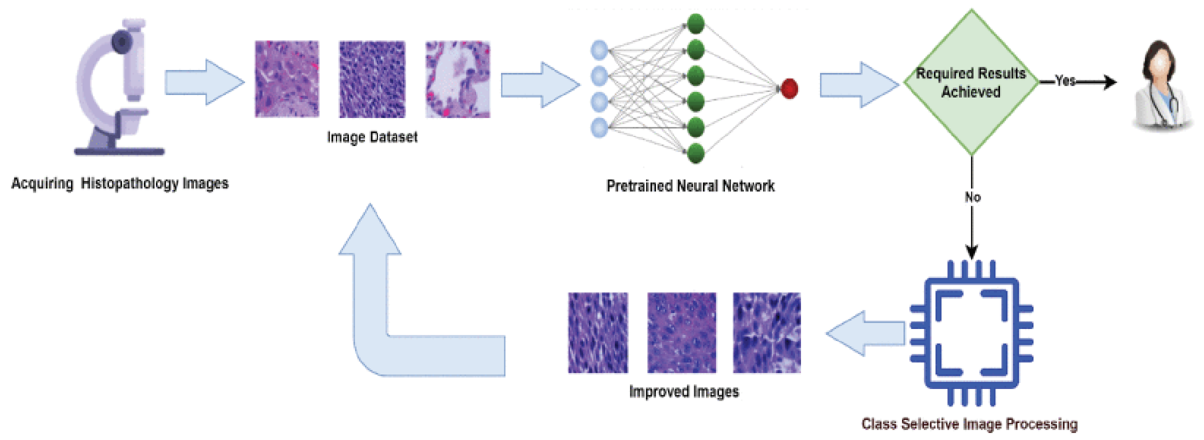

Acquiring Histopathology Images
Image Dataset
Pretrained Neural Network
Required Results Achieved
Yes
No
Class Selective Image Processing
Improved Images

Figure 2.2.3.1.: Proposed model of Malignancy Detection [15]

2.2.4. CNN-based Method for Lung Cancer Detection on WSIs

Early detection of lung cancer is crucial for the patient's treatment and survival. Pathologists analyze the tissue, but this work is very time consuming, and error is also susceptible. This paper proposed automated detection of lung cancer which helps the pathologists by relatively increases the speed of detection. It uses two CNN architectures VGG and ResNet on image patch stage for the classification [16].

2.2.5. Grand Challenge Solutions for Cancer Detection

In histopathological slides correct segmentation is very crucial to patient's care. It will tell the pathologist that whether the patient is recovering or not. This paper proposed ACDC in whole slide lung histopathology [17]. The focus of this paper is pixel wise detection which in scientifically called segmentation. Segmentation improves the overall performance of the system. It uses annotated data set, it includes 150 images for training purpose and 50 images for test purpose.

For evaluation purposes, following metrics are used

- Precision
- Accuracy
- Sensitivity
- Specificity
- Dc coefficient

Multi-model method is better than single-model method because DC value of multi-model is greater which is 0.7966 and for single-model it is 0.7544.

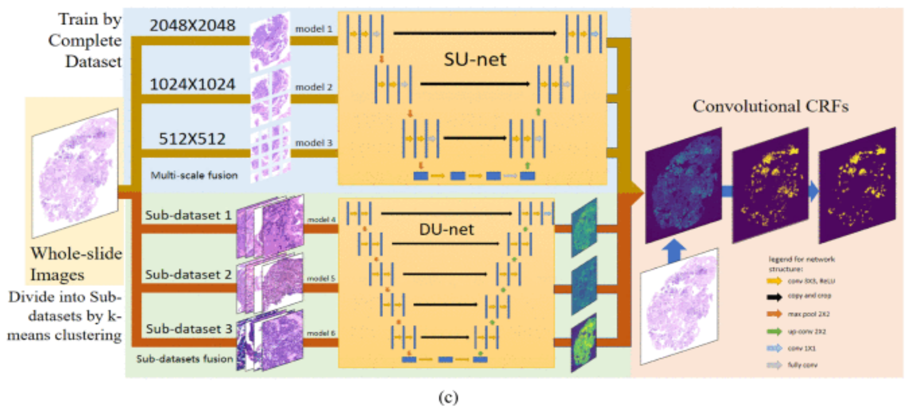


Figure 2.2.5.1.: Comparative Architecture of SU-Net + CRF [17]

2.2.6. CNN for cancer diagnosing histopathological images

Lung cancer is the leading cause of death around the globe. Histopathological diagnosis eases the classification of cancer cells. It proposed computer aided diagnosis system, which diagnosis squamous cell carcinoma and adenocarcinoma using convolution neural network (CNN). It uses LC25000 data set which includes five different classes, lung benign, colon adenocarcinomas, lung adenocarcinomas, lung squamous cell carcinomas and colon benign. Augmentation was used for the expansion of dataset into 25000 images [18]. This is done by flips and rotation. We will use this technique in our project for improving accuracy. The most advanced techniques of machine learning is deep learning as it does not require feature extraction from field expert rather it's model acquire knowledge itself. This model has showed improved accuracy than those of conventional machine learning models. CNN allows high-resolution textured images without converting them into low resolution textured images.

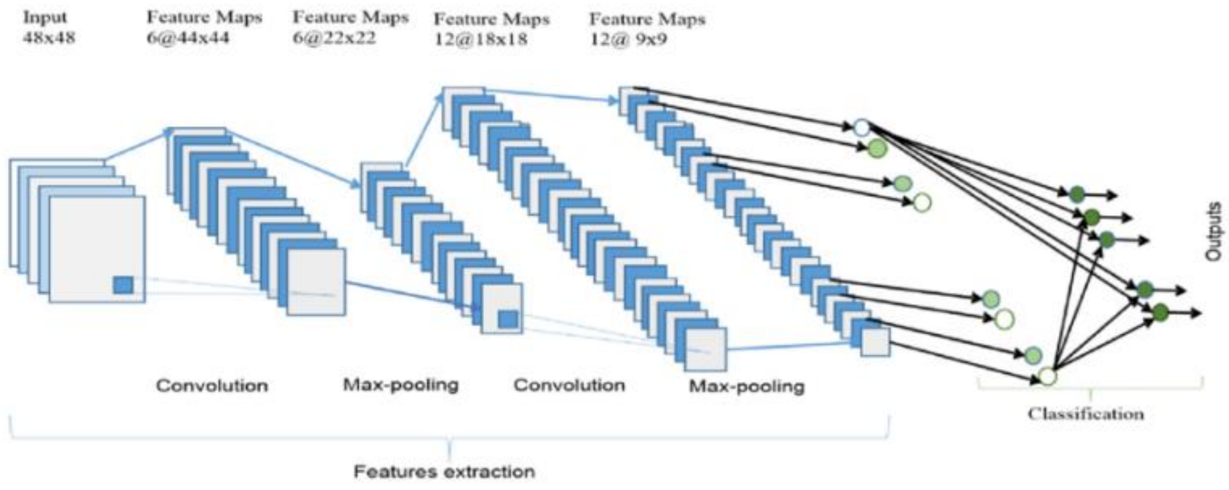
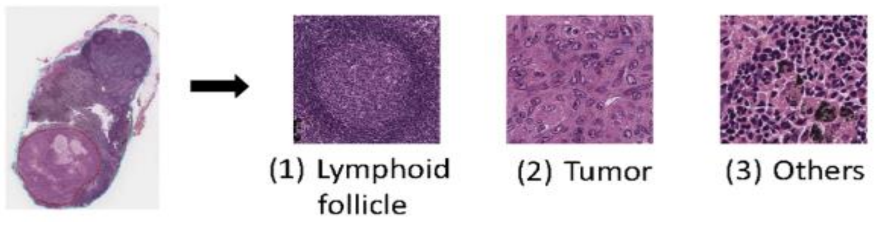


Figure 2.2.6.1.: ResNet Based CNN Architecture [18]

2.2.7. Two-Step Deep Learning Approach for Cancer Detection

For the interpretation of patient's situation and treatment, the detection of lymph node metastases on histologic slides has been used [19]. It uses an application of deep learning. This model deals with the false-positive predictions by developed two-step deep learning algorithm. There were total 349 full-slide lung cancer lymph node images, out of which 233 slides were used for algorithm training, 106 slides for evaluation and 10 slides for validation. Firstly, deep learning algorithm abolished missorted non-cancerous parts. Secondly deep learning classifier was developed for the detection of cancer cells. This approach has resulted in the reduction of error by 36.2 %. Sensitivity and specificity were also increased by using foci size thresholds of 0.6 mm and 0.7 mm.



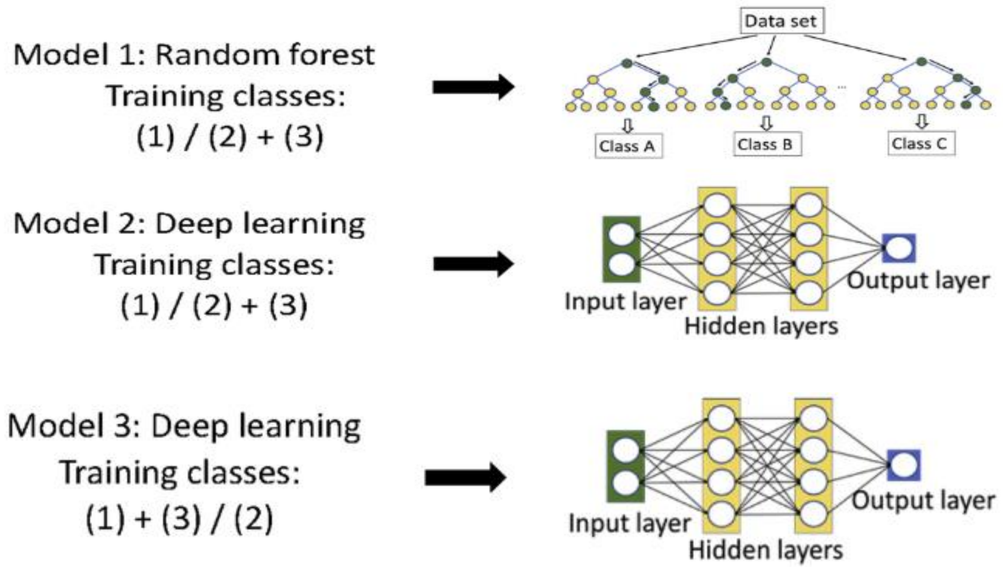


Figure 2.2.7.1.: Comparative Classification Architecture [19]

2.2.8. Artificial Intelligence in Lung Cancer Pathology Image Analysis

Early detection of lung cancer is very crucial to patients' health. Pathologists are struggling to find out the ways for the early diagnosis of the disease [20]. Whole slide image gives accurate and efficient analysis of the pathology image analysis. This paper has shown that how artificial intelligence such that deep learning has improved the overall efficiency of the diagnosis and pathology images analysis such as:

- tumor region identification
- prognosis prediction
- tumor microenvironment characterization
- metastasis detection

2.3. Summary

Extensive literature review has opened architectural diversity in lung cancer classification and detection as well. Transfer learning approaches are being used for lung cancer classification considering the LC25000 dataset and the U-Net architectures beside other optimization architectures are being used in multi-modal semantic segmentation for lung cancer detection in top 10 submissions of the ACDC lung cancer Grand Challenge Dataset. From the empirical knowledge obtained through literature review. Similar approaches will also be employed in lung cancer classification and detection.

CHAPTER 3
Implementation

3.0. Conceptual Background

Convolution neural networks have evolved so much in the recent decade that machine are now able to learn trivial as well as the most intricate patterns within the subject of their analysis. The complexity of these patterns can vary with the subject and the domain of the application of that particular subject. Recently matured domain of machine has offered engineers and researchers with large arsenal of tools to conquer brilliance in the domain of their research and application. The process of teaching machine to observe and figure out common patterns in different images of same domain is called training, and the complex mathematical amalgam which is being trained is called model. Model is the fundamental and most important component in machine and deep learning. Training machine to detect particular patterns and make inferences (decisions), based on these patterns are heavily dependent on mathematics and statistical concepts. It is of utmost importance to hold strong theoretical grounds of these mother concepts.

Teaching machines can be classified into three section based on difference of their working and output as well. In the first step, the model is being researched or developed based on the big picture regarding the final desired results. Once the model is researched and finalized then corresponding data is also cleaned and preprocessed for training model onto that dataset. In the next step when model is ready to be trained on cleaned data hyper-parameters are being set and the model is trained on the dataset. During the training process the model is also monitored for its performance on small chunk of data which the model is oblivious of is continuously feed so that computation and time can be saved in case the model training start to perform anomalous. In the final step, if the model have shown satisfactory training then it is evaluated on the new data on which it is predict the desired results.

Following are the fundamental elements of the model and training, which will play a pivotal role in the successful accomplishment of the results desired in our case of lung cancer classification as well as detection.

3.0.1. Skip Connections

It is imperative that image processing neural networks can become immensely deep and dense, which requires more computational power and time for better training and quality results. There are wide range of drawbacks deep and dense neural network can face like overfitting besides many non-trivial problems. So, an easy fix will be to use neural network which are shallow to bypass all these problems and train model smoothly, but shallow networks are unable to learn complex mathematical functions and thus are limited in their scope of real world problems. So the deep network are necessary evil in the real world problems. As the depth of the model increases the performance also starts to decrease and is called degradation problem.

In case of the shallow network, the model learns quickly due to less number of layers and the weights adjusts faster as compare to the deeper network. In case of the deeper network many problems arise which can impede the model performance such vanishing gradient, exploding gradient or overfitting. In case of ResNet, 20 layer network has less error than 56 layer network, and the author admits that error in 56 layer network can be bypassed by using BatchNormalization and proper weights initialization techniques can show satisfactory norms as well.

Considering this problem, a technique in which deeper network crafted with abilities of shallow network is used, this technique is called skip connection. In skip connection few layers of the network do not perform training and the data already learned weights are directly passed onto next layers, in this way the depth of the architecture do not deteriorate the performance of the model.

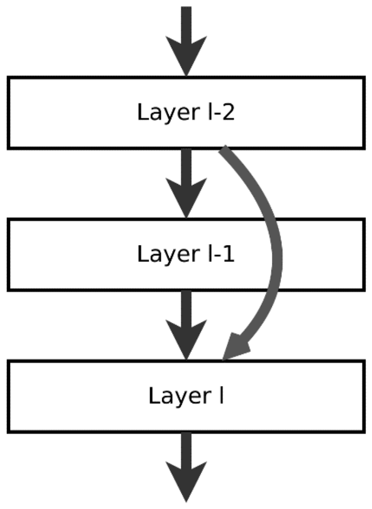


Figure 3.0.1.1.: Shallow network Vs Deep Network with skip connection

In the above figure the dense network also learns the same way as the shallow network but in the next layers the model layers will not learn from this and the weight will be transfer to the next layers. In other words the last three layers will only be transferring weights obtained from the previous layers onto the next layers and thus are called identity mapping or identity block, meaning it preserves the weights. In this way deeper network will also have error equal to that of the shallow network.

Skip connections are used for different purposes in different architectures, in case of resnet it is used to bypass degradation problem but in case of DenseNet, it is used for feature reusability. Another important feature of the skip connection is the smooth loss function which ensures fast training.

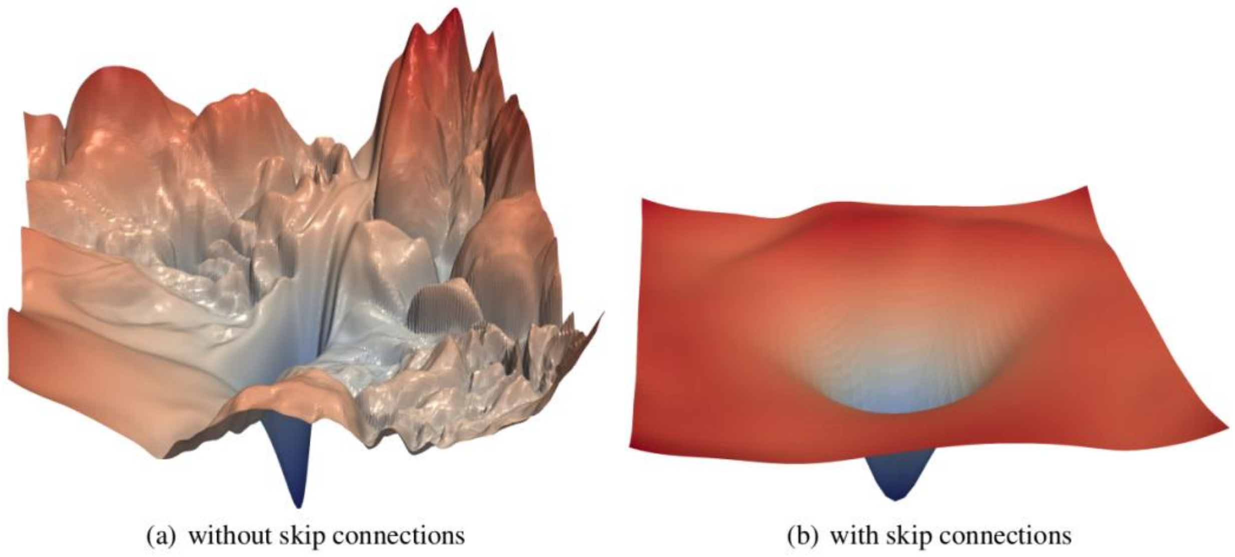


Figure 3.0.1.2.: Skip connection in model learning [21]

It is evident from the above figure that skip connections can improve the efficiency and reduce the loss of the model by introducing the identity block, or performing the identity mapping. The left side convex function more efficiently and faster to train model on and in more close to ideality as well, which is derived the real deep network based on the left.

3.0.2. 2D Convolution

Convolution is the primary step in deep learning for image processing. It would not be an exaggeration to say that all image processing based architectures employ convolution in one form or other. The purpose of convolution is to skim details of different pixels into one pixel. In this way the most important details of the images come out after convolution. Convolution is dependent on the input as well. If the input is a single dimensional vector then the convolution will be one dimensional as well and is called 1D convolution, similarly, if the input expand to two dimensional or three dimensional vectors/matrices then the output will also be 2D or 3D convolution.

As in the case of image the color image is a 3D matrix and thus requires 3D convolution, but it is imperative to know that 3D convolution results in 3D cube which contains the most abundant and important information in the original 3D cube of the image. If we want the output to be in 2D matrix then we should employ 2D convolution on individual channel or 2D matrix of the image. In this way the output will be 3D cube but the information obtain will also be the abundant of the information of individual channels of the original image.

In case of 2D convolution operation it is denoted by * (asterisk), mathematically it is sum of products of the smaller 2D matrix with the original image. The smaller matrix is called kernel and it differs from application to application. Convolution can be employed for wide range of functions, and these functions are accomplished through making changes in kernel only and performing the original mathematical of sum of products of the images with the kernel. The output of the convolution operation is called feature map the original image, as it contains distinguishing features of the original image, which are immensely important for training of the model and making inferences.

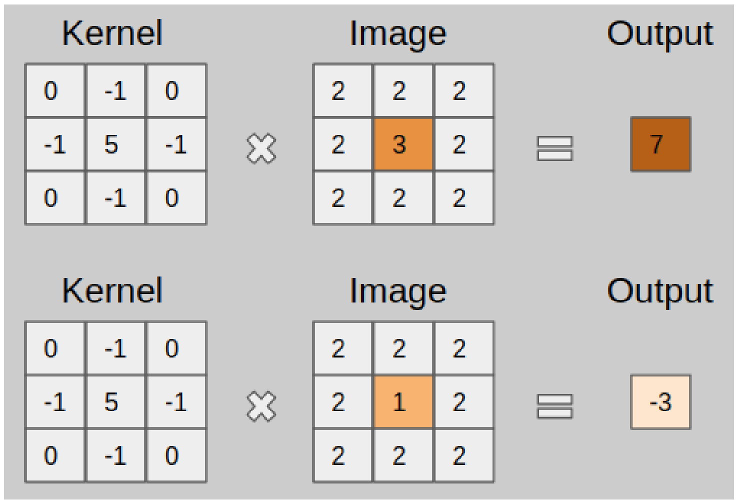
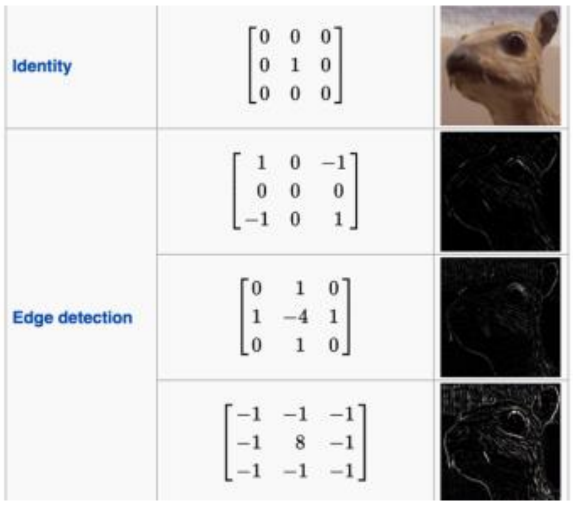


Figure 3.0.2.1.: 2D Convolution

In the above figure, it is illustrated that the image, which is 5x5 2D matrix in this case is convolved with smaller 2D matrix called kernel of 3x3 and the resultant feature map is also of size 3x3. It is important to know that output of convolution is smaller in size than the original image and the size is also dependent on the size of the kernel. In short words, the convolution will reduce the size of the image to some multiple of the size of the kernel, provided that the steps being took in each product is one. The steps taken in original image by the kernel to obtain consecutive values of the feature map is called **stride**. And the region which is being occupied by the kernel in individual step in the original image is called **receptive field**. In the above diagram bottom right region is the receptive field of that particular operation. The length and width of the feature map is governed by the length and width of the original image, kernel size and stride rate.



Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	

Figure 3.0.2.2.: Effect of changing kernel on convolution [22]

3.0.3. Transpose Convolution

Unlike 2D convolution, transpose convolution is its inverse. Convolution is a technique for down sampling, whereas up-sampling is accomplished through transpose convolution. It is imperative to understand up-sampling before jumping into transpose convolution. Up-sampling is obtained by using to construct the original image from the available feature map of the convolution. There are several techniques in up-sampling, some of the most common techniques are also outlined below.

Nearest Neighbor is the most-simplest approach of up-sampling. As the name of the approach suggest the value of the pixel is copied to its k-adjacent neighbors. This approach called k-nearest neighbors and the value of k is governed by the expected output of the up-sampling.

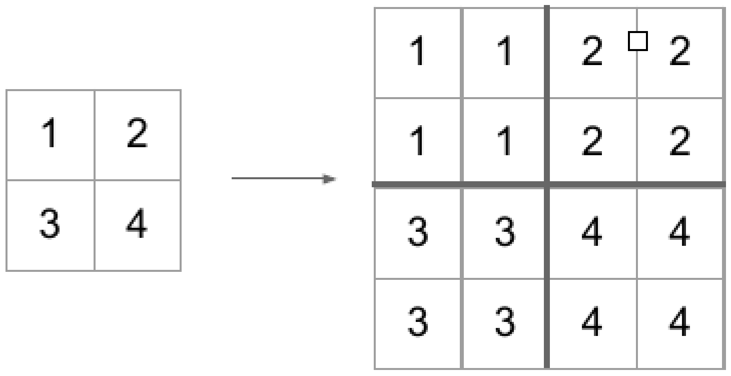


Figure 3.0.3.1.: Nearest Neighbor Up-Sampling

Bilinear interpolation is the second most common approach and widely used technique for up-sampling. In this method the weighted averages are being used to find the overall value of the pixels between the adjacent pixels of the feature map.

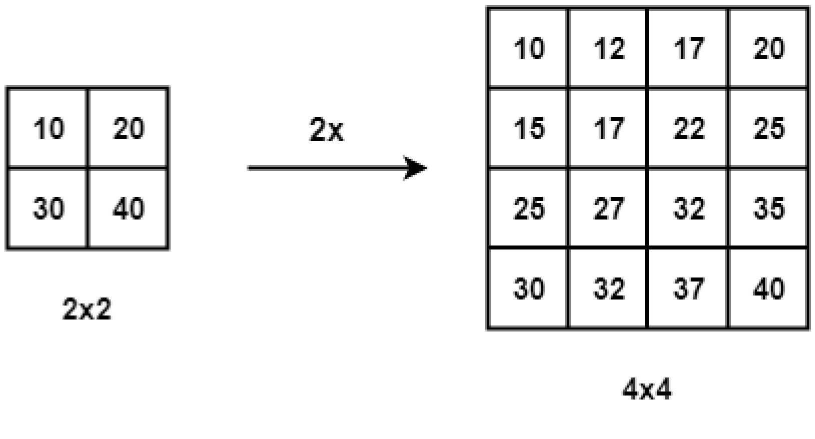


Figure 3.0.3.2.: Bi-Linear Interpolation Up-Sampling

Just like convolution, transpose convolution also has kernel, but we have feature map instead of image in transpose convolution. The output of the transpose convolution is the image itself. Due to the opposite nature of the transpose convolution it is also called inverse convolution or fractional strided convolution.

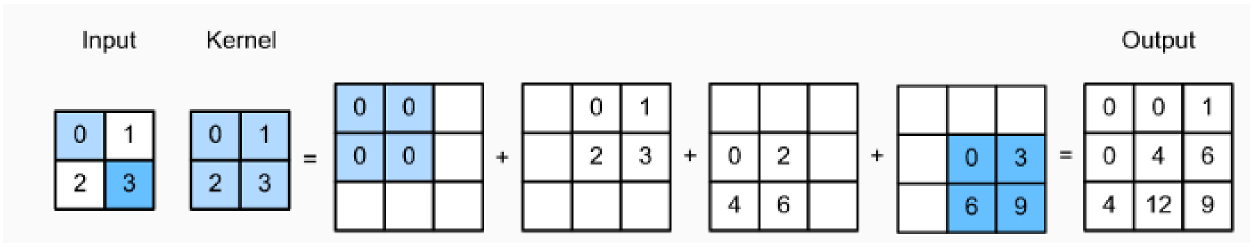


Figure 3.0.3.3.: Transpose Convolution [23]

The above figure demonstrate the process of fractional strided convolution or transpose convolution. In the second step, it the individual value of the kernel of is multiple by the same individual of the feature map or input image, in this case. In the last step the value of all individual output matrices are added to get the output matrix. Which is the original image whose convolution will give the feature map.

3.0.4. Activation Functions

Activation function is the most computed function in the complete pipeline of deep learning. In simple terms it is used to assist machine learn complex patterns and relationships within data. The output of the activation function depends on the input data but the choice of the activation function is completely dependent on the application and the user as well, thus it is a hyper-parameter. Activation function decide which neurons to fire in the next layer based on the quantitative decision from the previous layer. It is also used to distinguish layers between the neural networks. The output of the one layer of the neuron is fed to the activation function, which then decide for the input of the next layer.

The most prominent feature in the activation function is to add non-linearity into the deep network. Practical and real world problems are non-linear in their nature and thus can’t be represented as set of linear functions. From the linearity property, set of sequential linear functions can be represented as a one linear function. In the real world problem, non-linearity is incorporated and all the phenomenon can’t be modelled mathematically as linear equation. Activation function is the only block in deep learning responsible for introducing non-linearity in the network simulating one stop solution to the real world problem at hand.

There are few properties of the activation function, which are important to regard while choosing activation function hyper-parameter. In machine and deep learning, the model is trained using gradient descend process, in which the ultimate target of the process is to reduce the loss function and to find the global optimum of this convex function. The first and the primary feature of the activation function is that its output must be centered at zero, or symmetrical about y-axis. As mentioned earlier that activation function is the highly computed function in deep learning, so it must computationally cheap to avoid drain on the resources. The function must be able to differentiate between the layers of the neural network, thus producing output which is different from the input.

Another important feature of the activation function is that it must avoid vanishing gradient problem. Which is very prominent in the domain of machine and deep learning. If activation function faces vanishing gradient problem, then the learning of the model slows down and can be stopped as well. Suppose if activation function start to have value between 0 and 1, then in each step of back propagation the values multiplies and become smaller and smaller which will reduce the learning rate. As the value become very small the gradients disappear or in other words vanishes.

3.0.4.1. Sigmoid

Sigmoid is among the first activation functions for training the machine learning model. Sigmoid has some serious vanishing gradient problem in deep neural networks. In deep neural network weights of the new layers are calculated using the old weights and gradient of all the activation functions.

It is a computationally expensive activation function, and thus is not used heavily in deep networks. Shallow network can work fine with sigmoid. The basic function sigmoid performs is that if the probability of the received signal is more than 50% than the neurons of the output layers will be fired otherwise they will not fire.

In the above figure the derivative (gradient) of the sigmoid is Gaussian curve with maximum value of 0.25 similarly the maximum values of the gradients of multiple sigmoid functions can give value near to zero. In this case the gradient vanishes and the model will stop learning anymore, and it will cost computation as well as time loss.

The generic form of sigmoid is softmax, whose value also range from 0 to 1, and just like sigmoid it is also prone to vanishing gradient problem and is used in multiclass classification problems in the end of deep network or can also be used in between the layers of a shallow network.

3.0.4.2. ReLU

Most commonly used activation is the Rectified Linear Unit known by the acronym ReLU. In convolutional neural networks it has computational advantage, beside one side effect called dying relu problem. ReLU activation gives zero value for all negative inputs, causing nodes to die off, and completely hindering learning to zero. This problem can be bypassed using slightly different version of ReLU called leaky or parametric ReLU. It can also be restricted to particular maximum values using different thresholds for maximum. Setting the upper threshold to fix value avoids the ReLU explosion to infinity. Leaky relu is computationally expensive than simple relu.

3.0.5. Batch Normalization

Normalization is the process of changing value of the element in the data to typical value of the data such it do not disturbs the overall range of the data. Another meaning of the normalization can be understood from the fact that all variables in the dataset are normalized to a fix range scale. Suppose the case of car driving dataset in which we have two variables as car no of miles cars driven for past five years and the age of the driver. In the first variable the range can drastically differ from 1000 miles to 100,000 miles but age will be restricted from 18 to 80 years. If we do not normalized both these variable to a fix range than training of the model can take significantly large time as well as resources to train and it will be more error prone due to huge difference in the range of the both these variable in this case it is imperative to normalize these variables in a range between 0 and 1.

In the case of batch normalization, also known by acronym batch norm, if the weight of particular become immense large in comparison to the weights of other neurons then it will cause un-stability in the network. So the batch norm is applied on individual layer, whose weights can explode and particular weight dominate a neuron. In this case the batch normalization will ensure that weights are evenly distributed and none of the neuron is dominating in its weight.

Following steps are being in applying batch normalization.

$$Z = \frac{x - m}{s}$$

The above equation will evaluate the z-score of the batch which requires mean and standard deviation of the batch under normalization.

$$Z = (Z * g) + b$$

Two arbitrary variables i.e. g and b is multiplied to the z to obtain a new score, which is ensures even distribution of the output weights of the neuron. Arbitrary variables are also trainable and thus can be optimized as well for better results and computational advantages.

3.0.6. Pooling, Dense and Flatten Layers

Pooling layers are used for reducing dimensions. It is most commonly used in image processing techniques and convolutional neural networks. There are different types of pooling methods in image processing libraries of python. The famous among them are max, min and average pooling. Just like convolution, pooling method also a kernel but it is invisible and do not have any value, and output is the direct value of the input. In the case of max or min value, the maximum or the minimum value is being taken from the kernel chunk from the original image. But in case of average pool the average of the all values is taken and delivered to the output.

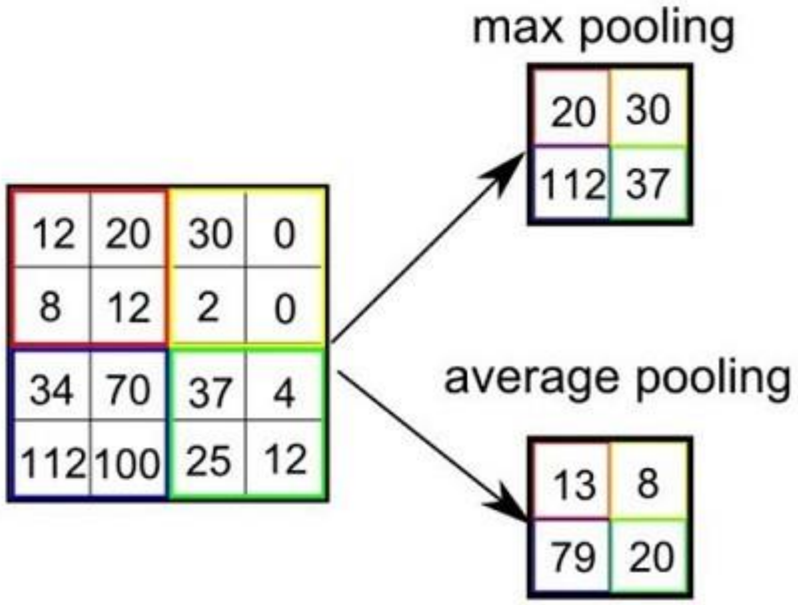


Figure 3.0.6.1.: Pooling Types

Dense layer is another type of layer in neural network, in dense layer each neuron of the input layer is connected with the each neuron of the output layer. Dense layer is most commonly used in artificial neural network. Dense layer is required in architecture, when input of all neurons is equally important in deciding the output of the layer. In classification task after performs all convolution, dense layer is used which will decide on the basis of inputs of all the neurons of the previous layers and give probability to each of the class, corresponding to number of neurons in the dense layer.

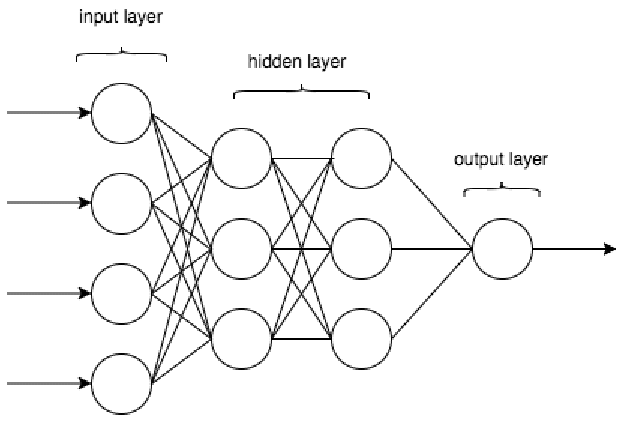


Figure 3.0.6.2.: Dense Layer generic structure [24]

Flatten layers is also used to reduce dimensionality of the input. In case of 2D image flatten layer will output 1D vector, there are different techniques which are employed in flattening 2D image or matrix into 1D vector.

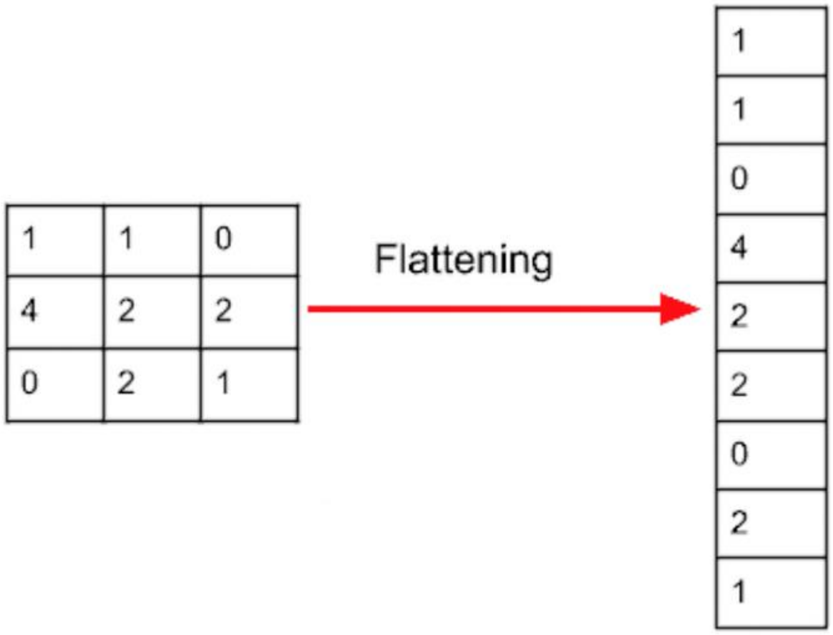


Figure 3.0.6.3.: Flattening 2D matrix into 1D vector

3.1. Lung Cancer Classification and Detection

Complete Characterization of the Lung cancer requires wide range of steps and numerous approaches to obtain correct results and draw precise and correct conclusions. Broadly, these steps can be divided into two categories, i.e. detection and classification. The former category deals with finding out the tissues which have cancer and the latter category determines the kind of cancer present in previously detected tissue. We have applied different models to obtain results in both of these categories. It is intuitive to detect cancer in the first step and follow classification afterwards. But the reversal of the techniques can also be applied without introducing significant error in the final obtain results. In other words, pathologist can bear small precision errors in the final obtains results unless it does not impart significant difference in threshold, responsible for governing the type and urgency of the treatment. Although, it is generally observed that classification of lung cancer is relatively less complicated than detection of the lung cancer. Keeping this perspective into regard we also have worked out classification in the first step then followed by detection in the final step. Furthermore, dataset also plays central role in the machine and deep learning results. In our case, we have used different dataset for classification and detection, which is also an important in going with classification in the first step.

3.1.1. Lung Cancer Classification

Microscopic analysis of the lung tissues, tumors more precisely, have classified the lung cancer in two major types: i.e. Small Cell Lung Cancers (SCLC) and Non-Small Cell Lung Cancers (NSCLC). The cure and treatment of both these cancers is different due to their different mode of growth and spread. As generally believed that lung cancer is caused due to smoking. Technically, smoking cause small cell lung cancer, which is not the most common type of lung cancer. It only comprised of 10-15% of all lung cancers, although it is aggressive and its detection can only be confirmed after its extensive growth. Whereas, NSCLC is the most common type of lung cancer and account for about 85% of the total cases.

As NSCLC is the most common type of lung cancer, it is further divided into three subclasses i.e. adenocarcinoma, squamous cell carcinoma and large cell carcinoma. Among these three types, first two are the most common. Whereas large cell carcinoma is rarely and is often undifferentiated from the other two types till extensive growth of the malignant tissues of this type. In simple words, the NSCLC initially can be classified into two types if it is in early stage, or classified into three types at the later stage. Adenocarcinoma is the major type of the lung cancer, and it can also be predicted with chest x-ray imaging and it is most likely pneumonia. This type of lung cancer damages alveolar walls, and it is most common among women. Squamous cell is also the second most common type of lung cancer after adenocarcinoma, and it creates a cavity into the lungs if remained unchecked and let to grow beyond bounds.

In our case, we have choose adenocarcinoma and squamous cell carcinoma for the purpose of the purpose classification. We also provide healthy tissue images, called benign, which are non-cancerous. In other words, the classification will be trained to predict on three types i.e. adenocarcinoma, squamous cell carcinoma and benign.

3.1.1.1. Dataset and Meta-Data

In cancer classification, we are using LC25000 dataset, which is a five class dataset and contains lung and colon images, and 5000 images per class, and total of 25000 images. As our problem is concentrated into lung cancer so we discarded colon image section of the LC25000 dataset, which has total 10,000 images. In the end we have 15000 lung tissue images belonging to three classes i.e. Adenocarcinoma, squamous cell carcinoma and benign. Originally, the dataset has 750 lung tissue images, 250 images belonging to each class. The resolution of each image is 1024x768, which is then cropped into 768x768 pixels. These original 750 images are augmented into 15000 images using augmentor library in python. Augmentor is a standalone library and it is most commonly used in the numerous real world image processing methods and approaches, and provides more control on the augmentation. Four augmentation techniques are being applied i.e. left and right rotation upto 25degrees and horizontal and vertical flips.

Table 3.1.1.1.: LC25000 Dataset Meta-Data.

Dataset	LC25000		
Original Images	750		
Classes	3		
Class Label	▪ Adenocarcinoma ▪ Squamous cell carcinoma ▪ Benign		
Per-class Images	250		
Original Dimensions	1024 x 768		
Cropped Dimensions	768 x 768		
Augmentation Ratio	1 : 20		
Augmentation Module	Augmentor Python Library		
Augmentation Type	Rotation	Right	Upto 25 degrees
		Left	Upto 25 degrees
	Flips	Horizontal	
		Vertical	
Total Images	15000		
Per-class Images	5000		
Dataset Size	2GBs		

The dataset is prepared and released by James A. Haley Veteran's Hospital and is also publicly available for the research purposes.

3.1.1.2. Model Architecture

From the knowledge of the available dataset and the final objective i.e. classification of the input image is dependent on complex mathematical functions, which are most commonly non-linear in their implementation and approach. Understanding and implementation of these functions using programming tools comes under the domain of deep learning. The network design for classification of the input image is the amalgamation of these smaller linear and non-linear functions. The ultimate benefit that can be derived from implementation of these deep neural network is the simulation of the real world complex processes. But deep neural network comes with wide range of problems as well. Most common among is the vanishing gradient problem, where model starts to learn slowly for the coming layers or explode at the exponential rate. These problems are solved using Residual Networks.

3.1.1.2.1. Residual Network

The vanishing gradient problem appear as the depth of the network increases in sequential fashion. This problem can be overcome by using the skip connections, which will act as the reinforcement of the original signal after passing through every few blocks of computation. In skip connections we simply transfer/copy the values of the signal and provide it to the output of the values after the desired computational blocks. The residual network architecture consist of two major blocks which are replaced with different filters replicated sequentially. Comprehensive architectural understanding of both these blocks are given below.

Identity Block

It is the standard block used in Residual Network architecture. As the name of the block suggest it preserves the identity of the input reaching at the input of the block to the information leaving out of the block. In technical terms, the dimensions of the input activation is exactly same with dimensions of the output activation. In order to completely understand the identity block smaller blocks are also introduced which simulates the identity block using the working of already known blocks. The working of the identity block can be simulated using the 2d convolution, batch normalization and ReLU activation. In simulating the identity block functionality we also require skip connection which will transfer the functionality of the input data to the output of the block.

The above schematic demonstrate the working of the identity block. The connection to the add layer is adding two things i.e. the data from the input layer and the output obtain the batch normalization for the second time. In order the limit the output into particular range the ReLU activation is being used. It is important to know that the dimension of the input and the dimensions of the output is exactly same and thus the functionality of this entire block is called as the identity block. The identity block can also be expanded to three layers as well. In that case we have to replicate the first layer twice, which is then followed by second layer. The layers of the identity block is dependent on the application of the residual network, and it is often added when the features of the input are diverse and has large variance, thus increasing the depth of the identity block assist in finding the features of the input.

Convolution Block

As the name suggest convolution block is responsible for performing convolution. Unlike identity block the dimensions of the input and output of the convolution block are not same, which can be intuitively understood by the schematic diagram of the convolution block. Unlike identity block convolution is a three layer block, and is generally deeper than the identity block.

In the above schematic, it is can be easily observed that there are three layers, and the skip connection is also performing convolution rather than directly copying the input to the output. Which is the major reason for dimension mismatch at the input and the output layer. It is important to understand that the dimensions of both operand must be same in order to perform addition. In the above schematic the addition represents the concatenation of layers. Number of methods can be implemented to achieve the desired shape of the output layer.

50-layer Residual Network (ResNet-50)

Different arrangements of convolution and identity blocks results in different architectures of the residual networks, which are efficient in different applications. Dense and deeper networks have high accuracy and precision in terms of features extraction, but they are also computationally expensive, thus are not suitable for small low feature applications. On the other hand, shallow networks of the resnet are computationally cheap but they are not a good fit for large feature datasets. Thus the decision of the architecture is highly dependent on the input data and the application of the results.

In our case the input data has less features but have similar composition of the cells, which make them difficult to distinguish if we use shallow network. But on the other hand if we use too deep network then it will be computationally expensive and can give results which might be overfitting. In this case we have choose the median architecture, which is both computationally cheap and viable to extract all important features to make significant distinction between different types of carcinomas.

We chose the middle architecture which is median between different architectures i.e. 50 layer architecture of residual network called as resnet50. Following schematic clearly demonstrates the resnet50 architecture.

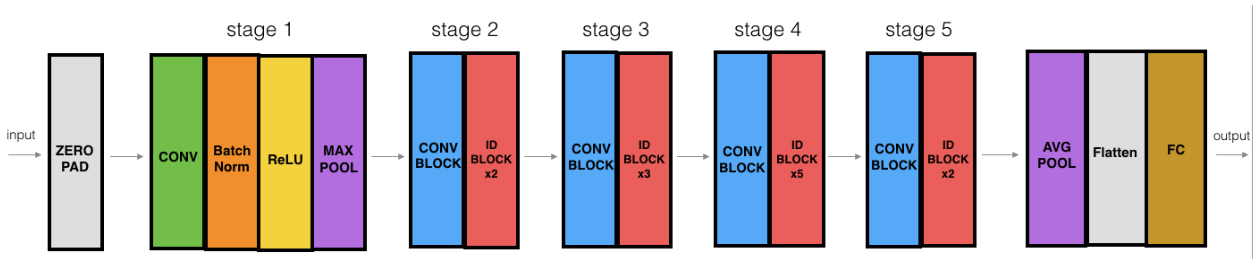


Figure 3.1.1.2.1.3.: Residual Network Architecture [25]

The above architecture has 50 layers in total. Commonly repeating layers are replaced by bigger blocks, such as conv block and ID block. The multiplier presented with different blocks represents the number of times the block is repeated. Combination of blocks are joined to form a stage. Stages are represented within the input and the output block. In other words, hidden layers are represented in terms of a stage. After the stage five the network of layers is used, which is responsible for making classification on the features received from the hidden layers. The classification network consist of average pooling layers, Flatten layer and Fully Connected layers. The average pooling layer is used to average the values obtained from the last layer of the hidden layer using 2x2 kernel. The matrix obtained after averaging is converted into single vector using flatten layer. The vector is then make into probability function using the fully connected network. The fully connected layers has neurons. And the number of neurons in the fully connected layer is equal to the number of classes. In our case we have three classes to predict so the fully connected layer also called dense layer has three neurons and the probability of each of these neurons governs the confidence of that particular class.

We have three neurons in the dense layer of the fully connected network and the final prediction is based on the highest probability of the neuron corresponding to each class.

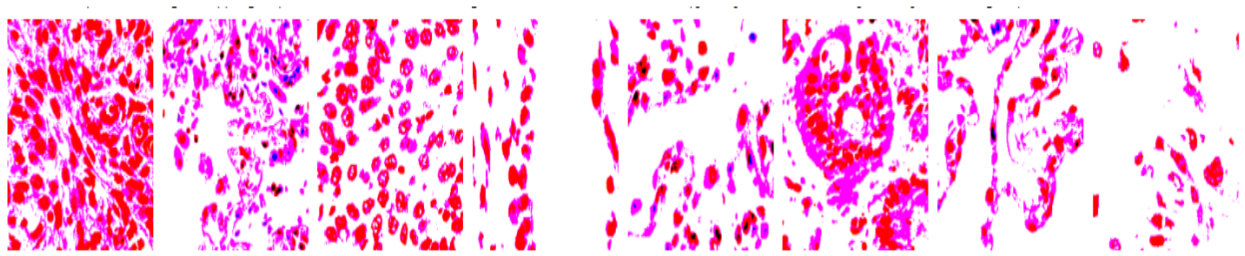
3.1.1.3. Data Preprocessing

In deep learning, data preprocessing is the most important step. Poor data preprocessing results in bad model training and thus poor predictions and results and vice versa. Originally, we have 15000 images and three classes. 5000 images corresponding to each class. We have divided that data in training and validation in the ration of 80 to 20. Which means that 80% of the images are corresponding to training images and 20% images are corresponding to validation images. Following table illustrates the division of the dataset in the corresponding folders.

Table 3.1.1.3.1.: ResNet Data-Split

Dataset	**Training (80%)**			**Validation (16%)**			**Test (4%)**		
	ACA	SCC	Ben	ACA	SCC	Ben	ACA	SCC	Ben
LC25000	4000	4000	4000	800	800	800	200	200	200
Total	12000			2400			600		
	15000								

After splitting the data into training and validation folders we have used the residual network provided preprocessing to input data, which removes red channel from the input images and then trains the network. Other than removing the red channel resnet preprocessing does not make any other preprocessing. After preprocessing the input images the final images shown as follow.



```
[[0. 0. 1.]
 [0. 1. 0.]
 [0. 0. 1.]
 [1. 0. 0.]
 [0. 1. 0.]
 [1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

Figure 3.1.1.3.1.: ResNet Preprocessed Input Batch

After preprocessing the input images the above image shows the batch of 8 images. The top is the images obtain after applying preprocessing and the 2D matrix contains the labels corresponding to each image. The index of each array represents the class of the image. For example index 0 represents the adenocarcinoma and the index 1 and 2 corresponds to squamous cell carcinoma and benign. The value 1 occurring in particular index is shows that the class of that particular image. In the above case the first image corresponds to benign and the second image is squamous carcinoma. The model is trained on following hyper parameters.

Table 3.1.13.1.: ResNet50 - Hyper-Parameters

Model	ResNet50
Optimizer	Adam
Learning Rate	0.0001
Loss	Categorical Crossentropy
Metrics	Accuracy
Epochs	5
Batch Size	128
Model Store Name	ResNet50.h5
verbose	2

3.1.1.4. Mobile Application Deployment

3.1.1.4.1. Software Development Kit

It is possible completely to write our own code from scratch but is a tedious and time-consuming process for developers to do this every time they came up with a new idea. Like, imagine you have to grow plants and harvest your own wheat and you want a delicious slice of raisin every time. The software development kit is like a virtual toolbox and its files are called as libraries which are essentially pieces of code that help a program with whatever platform it is on such as the libraries in the windows SDK which allow programs to make calls to the operating system to get it to perform certain functions easily without having to tediously code them manually. Other common SDK tools include visual editors so that a developer can design and layout graphical elements like buttons and text boxes easily with sample codes so that a developer can become familiar with the platform they are coding on and debuggers that help developers find errors and omissions in the code but SDK's are not just limited to making the programs on windows or smartphone apps. An individual SDK is often heavily customized or their platform such as the Cosmo SDK from the Robotics company Anki which can be used to harness the robots camera for facial recognition and spatial awareness, animate the robot to make it say or do certain things plan out movement path to make it easy for the robot to avoid obstacles all with just a few lines of code instead of having to create from scratch and even with other platforms we are not limited to just using one SDK. There are a lot of different ios and Android SDKs that specialize in different types of apps. There are chat-focused SDKs that make it easy to create notifications or speech balloons image editing SDKs that allow making certain photo effects and filters available without having to code them yourself and video chat SDKs that make it easy for the app to talk to phone's camera.

3.1.1.4.2. Flutter SDK

Flutter is a software development environment and a UI toolkit from Google to build natively compiled apps for mobile, web, and desktop although it's mostly used for mobile and that's what we will be focusing on. The most exciting thing about flutter is that we can build very performant iOS and Android apps native apps using only one code base rather than the traditional route where we would have either Java or a Kotlin base for an Android App and then we would have a swift code base for iOS app with flutter not only do we have a single code but it's easier to create applications. Flutter is cross-platform so we can use it on Mac, Windows, and Linux. Flutter is extremely fast as it gives the native field. There are other options for mobile development using kind of web technologies like react native, script ionic, and some others but flutter is generally faster because it doesn't have that extra JavaScript bridge to the OEM user interface. Flutter actually uses the native ARM Architecture binary which makes flutter faster than all of its competitors.

3.1.1.4.3. Dart Programming Language

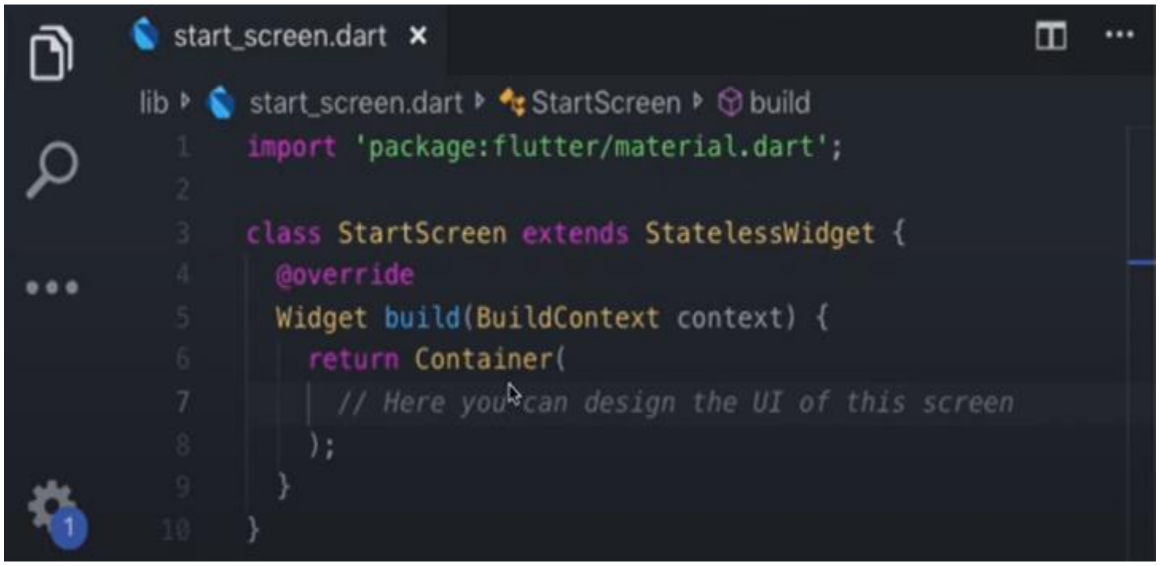
Flutter uses Dart language which is an object-oriented language that uses classes and its very fast on all platforms. Dart language resembles much to JavaScript and it even has some elements of Java.

3.1.1.4.4. Widget

Everything in flutter is essentially a widget and flutter widgets are pixel for pixel and material design comes out of the box with flutter. Some of the common widgets are scaffold which is like high-level widget that can have lower-level ones inside of them like a container, an image, an icon even text and app bar things. There is a whole catalog of widgets, like widgets for animation and motion input accessibility, etc. Widgets can be stateful and stateless. There is a lot of similarity of widgets with reacts or any front-end JavaScript framework. Instead of using components we use widgets and each widget has a build method that can be overridden just like in react, we have render. It’s a method in which a component that will render it into the UI, so there’s a lot of similarities in here.

3.1.1.4.5. Stateless Widgets

Stateless widgets are immutable means there state cannot change during the runtime of the app. Here is an example in which we have a class called start screen and extends stateless widgets and we are overriding the build method and at the end we are just returning an empty container widget.



```dart
import 'package:flutter/material.dart';

class StartScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Container(
      // Here you can design the UI of this screen
    );
  }
}
```

Figure 3.1.1.4.5.1.: Stateless Widgets

3.1.1.4.6. Stateful Widgets

In Stateful Widgets, we have a mutable state that can be redrawn on-screen multiple times. In this example, we have a Start Screen class that extends stateful widgets and in the next line instead of overriding a build method we are overriding create state and returning an instance of start screen stateand in the end, we are outputting UI which is just an empty conatiner. For stateful widgets we are gonna have two classes, one of which overrides the create state and the other one to build UI.

3.1.1.4.7. Requirements

For building iOS app on Mac, we require the latest Xcode and that will give us iOS simulator that will be used to view our application. On windows, we need Android Studio.

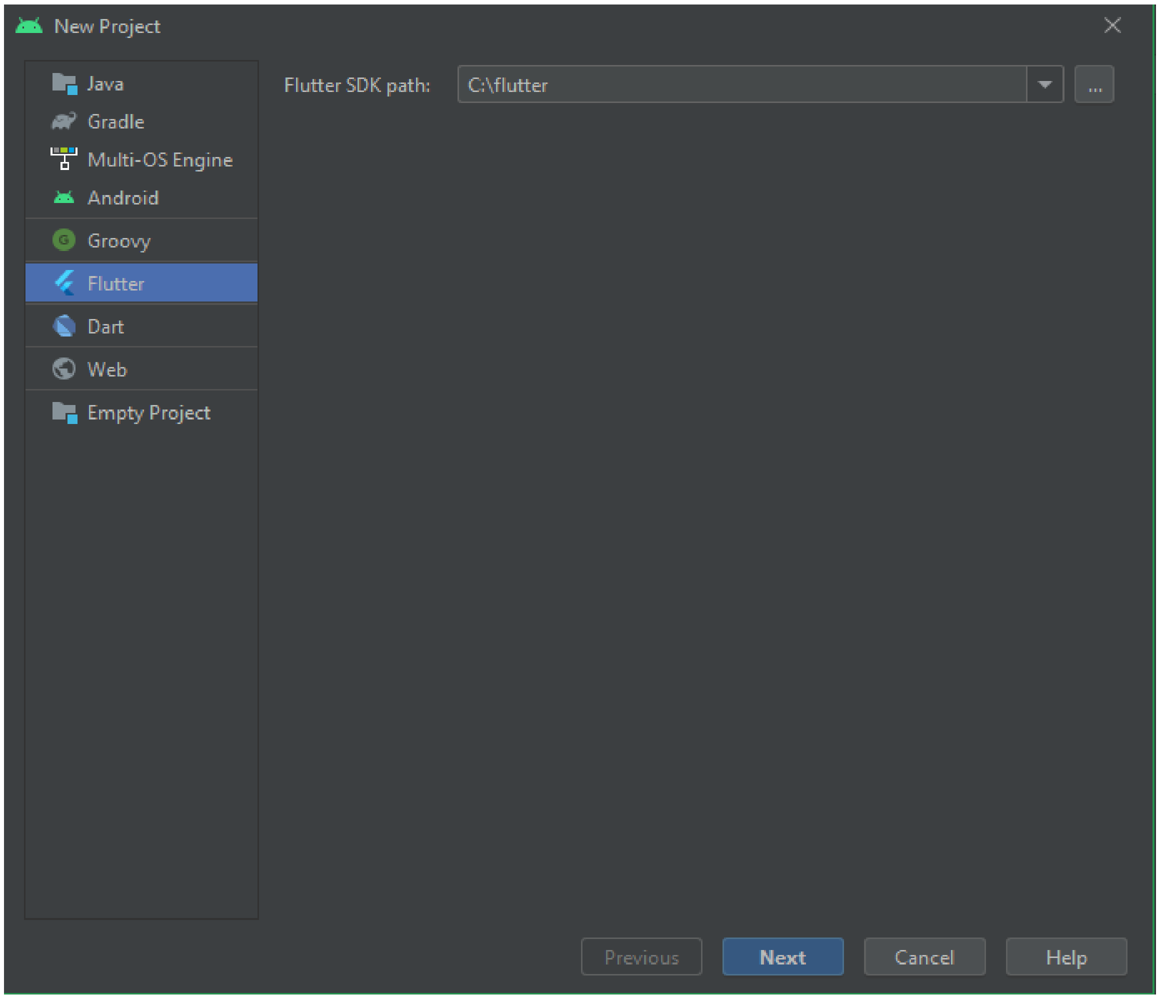
If we are using Mac and building app for Android then we need Android Studio, the SDK, the emulator which is the AVD (Android Virtual Device). We also need Flutter Plugin for Andoid Studio and if we are using VS Code, we will need flutter extension which has tons of intellisense features.

3.1.1.4.8. Installation

Flutter. dev is the official website for downloading flutter. Choose the operating system between Windows, macOS, and Linux. A minimum of 4GB RAM is required for efficient workflow.

3.1.1.4.9. Setting up Android Studio with flutter

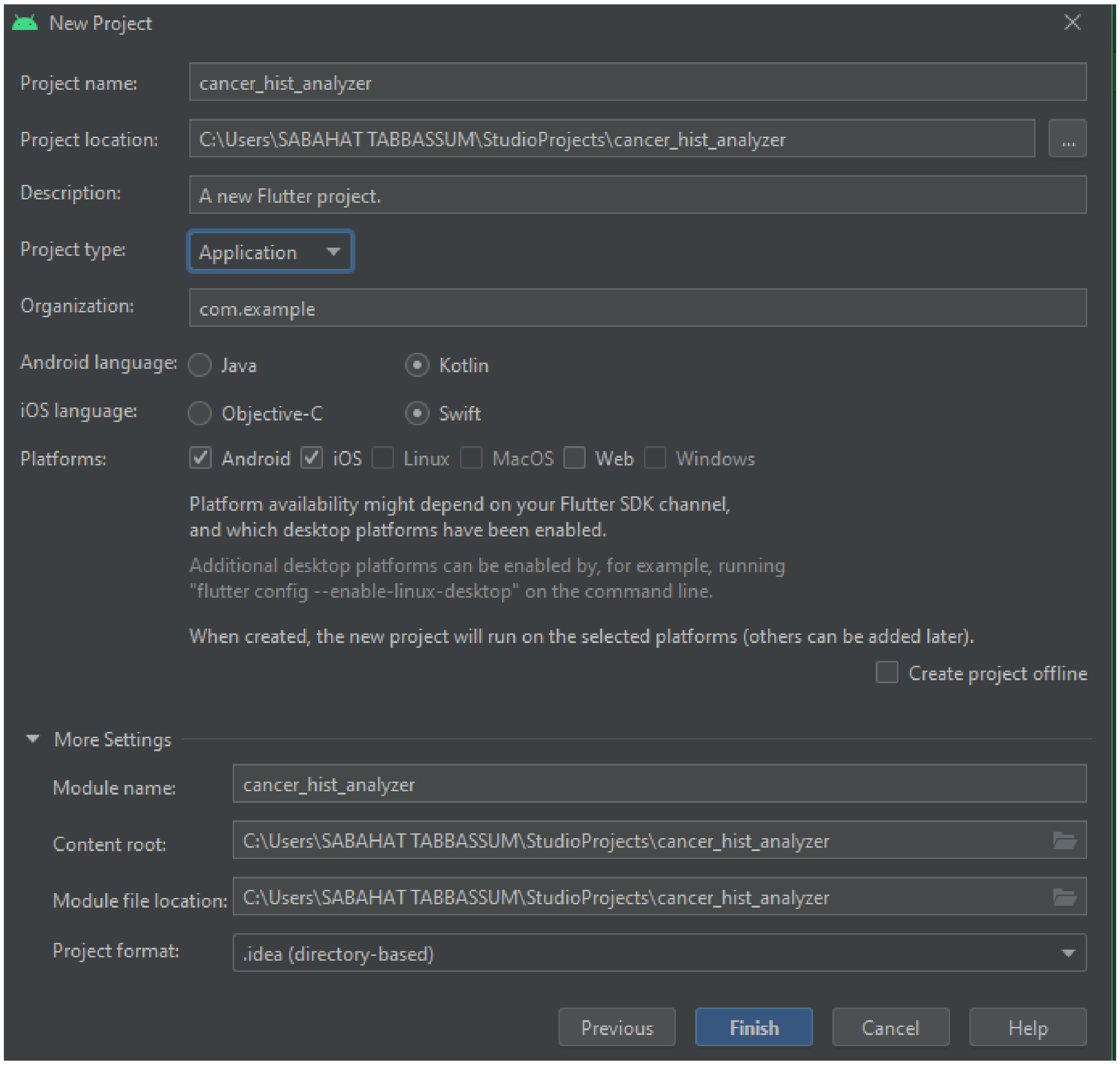
Start the Android Studio App, open the plugins of Android Studio and searhc for Flutter plugins from marketplace. Install the Dart Plugin also.



Flutter SDK path: C:\flutter

Figure 3.1.1.4.9.1.: Setting up Android Studio with flutter

Then restart the app and click on New flutter project, after that we have to provide Flutter SDK Path and then create the project.



Project name: cancer_hist_analyzer

Project location: C:\Users\SABAHAT TABBASSUM\StudioProjects\cancer_hist_analyzer

Description: A new Flutter project.

Project type: Application

Organization: com.example

Android language: Java [✓ Kotlin]

iOS language: Objective-C [✓ Swift]

Platforms: [✓ Android] [✓ iOS] Linux MacOS Web Windows

Platform availability might depend on your Flutter SDK channel,
and which desktop platforms have been enabled.

Additional desktop platforms can be enabled by, for example, running
"flutter config --enable-linux-desktop" on the command line.

When created, the new project will run on the selected platforms (others can be added later).

Create project offline

More Settings

Module name: cancer_hist_analyzer

Content root: C:\Users\SABAHAT TABBASSUM\StudioProjects\cancer_hist_analyzer

Module file location: C:\Users\SABAHAT TABBASSUM\StudioProjects\cancer_hist_analyzer

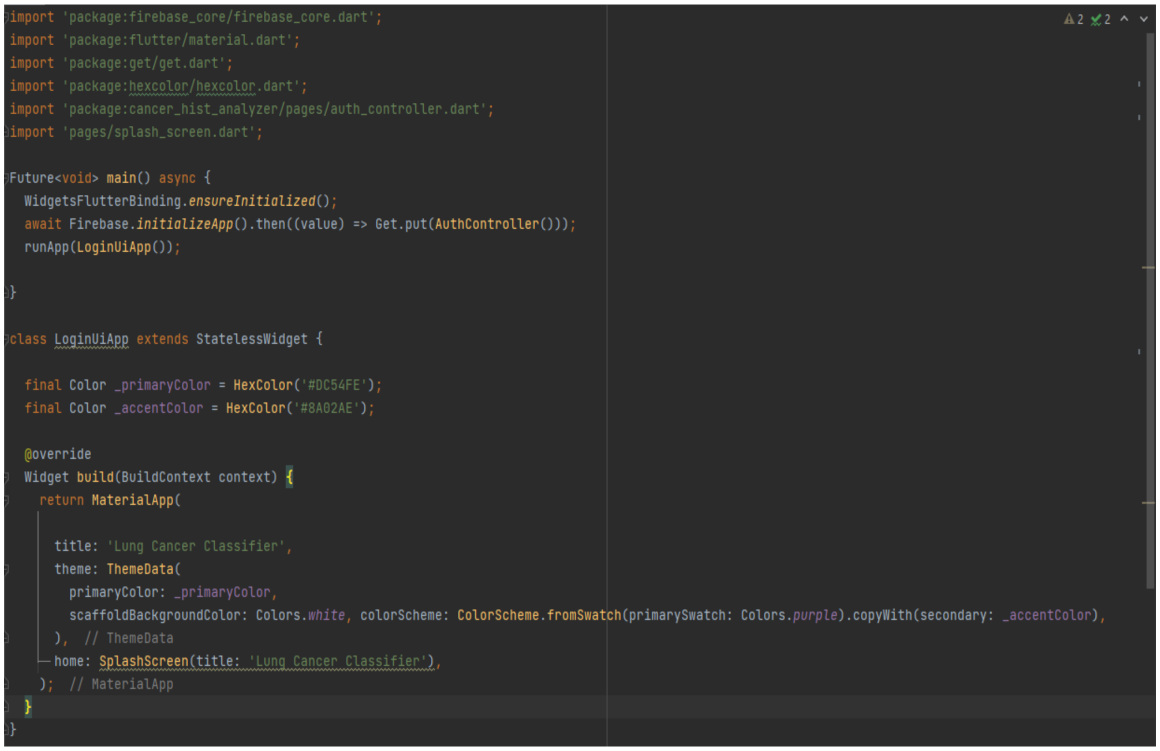
Project format: .idea (directory-based)

Figure 3.1.1.4.9.2.: Setting up Android Studio with flutter

3.1.1.4.10. main.dart

In lib folder there is a file name main.dart. In this file, the code will be compiled and bundled with the Android or iOS code so it will be bundled into one application which is then shipped to the native application device and the device in this app. Then the main.dart file or one specific function in there will be compiling unbundling this project and executing the first functions that are displayed to the user. This is called the main. dart file and also called main. This file is very important as it will apply the bundle and execute the code for iOS and Android.

Whenever the app starts, the flutter looks for the main method which is called *main*. First of all, we import the necessary basic packages needed for the code-like material. dart package. After that, we initialized the firebase by creating an instance of AuthController. The Firebase Authentication is needed for login and sign-up purposes. After that create a class named LoginUiApp having Stateless Widget.



```dart
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:hexcolor/hexcolor.dart';
import 'package:cancer_hist_analyzer/pages/auth_controller.dart';
import 'pages/splash_screen.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp().then((value) => Get.put(AuthController()));
  runApp(LoginUiApp());

}

class LoginUiApp extends StatelessWidget {

  final Color _primaryColor = HexColor('#DC54FE');
  final Color _accentColor = HexColor('#8A02AE');

  @override
  Widget build(BuildContext context) {
    return MaterialApp(

      title: 'Lung Cancer Classifier',
      theme: ThemeData(
        primaryColor: _primaryColor,
        scaffoldBackgroundColor: Colors.white, colorScheme: ColorScheme.fromSwatch(primarySwatch: Colors.purple).copyWith(secondary: _accentColor),
      ), // ThemeData
      home: SplashScreen(title: 'Lung Cancer Classifier'),
    ); // MaterialApp
  }
}
```

Figure 3.1.1.4.10.1.: main.dart

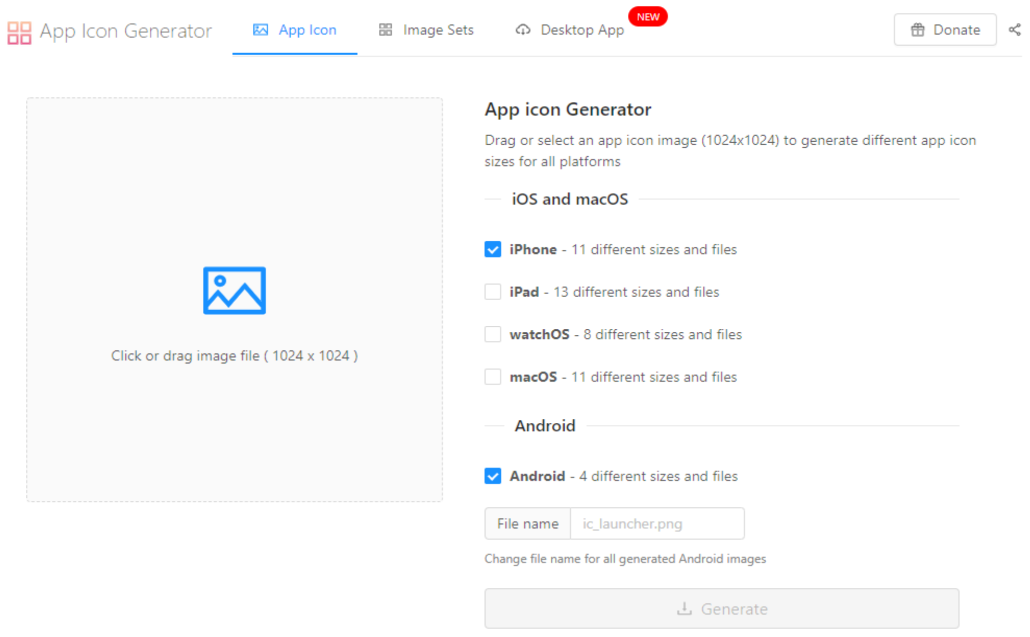
After that we created different pages like

- login_page. dart
- HomePage.dart
- splash_screen. dart
- PredictionPage. dart

- SecondRoute.dart
- ThirdRoute.dart
- Auth_controller.dart

3.1.1.4.11. Logo

We designed the app logo on Canva. By default the flutter logo is added in flutter apps which is displayed as the app icon. But we can customize it. After designing the logo we made it transparent by online means. There is a website named App Icon Generator.



App icon Generator

Drag or select an app icon image (1024x1024) to generate different app icon sizes for all platforms

iOS and macOS

- [✓] **iPhone** - 11 different sizes and files
- [ ] **iPad** - 13 different sizes and files
- [ ] **watchOS** - 8 different sizes and files
- [ ] **macOS** - 11 different sizes and files

Android

- [✓] **Android** - 4 different sizes and files

File name: ic_launcher.png

Change file name for all generated Android images

Figure 3.1.1.4.11.1.: Logo

There are different sizes and files for the different operating system. As our app is a mobile app so we selected iPhone and Android and uploaded the app logo. As a result files are generated.

For Android

Click

Android > app > src > main > res

Right click on res and open with *show in explorer*. Delete all the mipmap files in this folder and replace them with mipmap files generated by App Icon Generator.

For iOS

Click iOS and open Assets.xcassets with file explorer. Delete this entire folder and replace it with the folder generated by App Icons Generator.

The app icon is

Figure 3.1.1.4.11.2.: Logo

3.1.1.4.12. SpalshScreen.dart

Splash Screen is needed because Android App takes some time to start especially when it is loading for the first time on a device. A Splash Screen may display some startup progress or indicate a branding. Check Appendix.

3.1.1.4.13. login_page.dart

Login Page allows users to log in to the application by entering their email address and password. If don’t have an account already, first create the account. Login Page directs the user to the next page named as Prediction Page. Check Appendix.

3.1.1.4.14. PredictionPage. dart

Login Page refers us to the page we named as PredictionPage. It has three tabs

- Home
- Gallery
- Camera

All of these tabs are control by different dart pages. Home tab is controlled by dart file named as HomePage.dart. Gallery tab is controlled by SecondRoute.dart and Camera Tab is controlled by ThirdRoute.dart. This page also has a drawer which has options of splash screen, login page, Registration Page and Logout. Check Appendix.

3.1.1.4.15. HomePage .dart

This is the HomePage of the application. A basic overview of the app is outputted on its screen and it tells the user to upload histopathological images for the detection. Check Appendix.

3.1.1.4.16. SecondRoute .dart

The gallery tab is controlled by dart file named as Second Route. It gets the image from gallery as input. Then upload it on server by clicking the button. It send the image to the web link which is entered in the code. The deep model analyze the image and classify its type and output result is displayed by clicking on predict button.

3.1.1.4.17. ThirdRoute .dart

The camera tab is controlled by dart file named as Third Route. It is similar to the SecondRoute except that it inputs the image from camera instead of gallery. Check Appendix.

3.1.2. Lung Cancer Detection

Lung cancer detection is the two-step process of classification and localization. Cancer detection is destined to classify a tissue as tumor or benign, then also finding the exact location of the tumor tissue in the input image. Cancerous tissues are taken through biopsy and observed under microscope, and through manual test cancer is being identified. Digital image of microscopic specimen is also made using different types of scanners. The final scanned digital image is called as histopathology image. The radiologist input is required to mark the histopathology image with the region of tumor, and this process is called annotation. The annotation of the image is then used to create the mask of the histopathology image. As one obtain both elements i.e. histopathology image and the mask then deep learning can be applied to lung cancer detection. The scope of detection of lung cancer is dependent on classifying into cancerous and non-cancerous region. The sub-type detection of the lung cancer is not in the scope of this work.

Size of the histopathology is immense large and is dumped into less compressive image format like tif. Due to these protocols the overall size of the dataset increases drastically into GBs and similarly high specification computation machine is required to perform fruitful computation. In our case we have employed U-Net for lung cancer detection. The ultimate reason beyond the choice of architecture is its ability of semantic segmentation and relative simplicity in contrast to other semantic segmentation architectures. The dataset used for lung cancer detection is ACDC Grand Challenge dataset.

3.1.2.1. Dataset and Meta-Data

For lung cancer detection, we are using ACDC Grand Challenge Lung Cancer Dataset. The dataset consist of histopathology images, various code snippets and packages, assisting in different ways to understand the dataset. The dataset contains 150 histopathology images of variable size and dimensions with the same compression format i.e. .tif. The overall size of the dataset is about 250GBs, and the smallest byte size of an image in the dataset is 800MBs and the largest byte size is 5.4GBs. The dimensions of these images also differ greatly and typical dimension of any image is larger than 100,000x100,000. All these specification of the data marks it a unique piece of information, which requires different preprocessing techniques as well as architectural innovation to obtain desired results.

Beside 150 histopathology images .xml annotations are also provided, which basically have the coordinates of tumor region within the histopathology image. The annotations are provided by expert pathologists. As mentioned earlier about the dimensions and size of the histopathology image, it is evident that normal windows image viewer cannot open and give visualization of such large scale image. Thus a special package is also provided by dataset providing committee called ASAP, Automated Slide Analysis Platform. ASAP is a light weight and fast histopathology image visualization package. In ASAP histopathology image can open and viewed as well as the .xml annotations can also be loaded onto the same image. In this way the cancerous tissues can be visualized by the bounding boxes. Furthermore, custom annotation or bounding can also be made using the same tool, which GUI based user friendly and easy to use software, and no-coders can also use this software for histopathology image analysis and annotation. The customs annotations will also have .xml files which can be easily shared between the systems.

Dataset also provide code snippets two code snippets. The first code snippet is used to generate mask of the histopathology image using .xml annotations. And the second code snippet provides small patches of large WSI, Whole Slide Image.

Table 3.1.2.1.1.: Detection Dataset and Meta-Data

Dataset	ACDC Grand Challenge Lung Cancer
Image Type	Histopathology Whole Slide Image
Image Format	TIFF
No. of Images	150
Smallest Image Size	800MBs
Largest Image Size	5.4GBs
Total Dataset Size	250GBs
Annotation format	XML
No. of Annotations	150
Cancerous Patches	1,080,000
Patch Dimensions	512 x 512 x 3
Patch Format	JPG

In the initial step of data preprocessing the histopathology images are being break into small patches for manageable analysis and preprocessing. 2D binary mask is being generated using the code snippet provided by the dataset and the images are the mask is also break into small patches. Due to large dimensions of the original histopathology image the small patches of dimensions 256x256 become abundant and nearly 1.08Million cancerous patches are being made out of these 150 original WSIs. The patches are also stored for later use and easy visualization in .jpg format. Finally trained model also provide output results in the form of mask of a patch of the input image rather than giving the overall mask of the entire WSI.

3.1.2.2. Model Architecture

Due to twostep process of detection i.e. classification and localization in the image, deep and dense architectures are being explored and the simplicity is also regarded. In the extant literature, complex architectures are being employed, which achieve phenomenal results as well as quality performance on training and validation metrics. But thorough understanding of these architectures is both complex and beyond the scope of this study. Considering the simplicity and the objective of semantic segmentation, U-Net find out to be the perfect candidate. In the literature, U-Net is being used for histopathology image analysis and different applications of semantic segmentation, in addition with other technique for boosting results.

3.1.2.2.1. U-Network

U-Net is a convolutional neural network architecture especially designed for semantic segmentation and detection in biomedical images. The network is dedicated to train on lesser training images and work on larger images for testing. In this limited sense, the architecture can have weakly unsupervised method of learning and is more desirable in wide range of instances for medical images where dataset is often scarce and required results should be highly precise. Although the architecture of U-Net is deep as well as dense but on modern GPU it takes less than a second to give segmentation results for an image of dimensions 512x512. The architecture stems from the fundamental concept of fully connected convolutional networks, where every layer is connected with the other layer and thus is more produce feature channels. The biggest modification in U-Net is the larger number of feature channels and the high resolution mapping of the feature or context information.

Contracting Path or Encoder

U-Net architecture consist of two parts i.e. encoder side and decoder side. The encoder side is also called contracting path. The contracting path contains three fundamental layers, which keep on replicating themselves until the feature vector gets into the shape of the 1D vector. Among these three layers two of them are convolutional layer and the one is max-pooling layer. Using only these layers in contracting path will make the U-Net architecture overly simple and can have more rough results, which have higher error probability and architecture may give anomalous results as well. This contracting path can also fall prey to normal problems like overfitting. So wide range of optimizations are being applied to avoid these shortcomings. Additional layers are being added like dropout and Batch normalization. Beside all these optimizations padding is also applied in some necessary cases.

In the above diagram the contracting path with three steps is shown. In each step three fundamental operations are being performed i.e. two times convolution and one max pooling. The arrows in the horizontal direction denotes convolution and the down-ward arrows show max-pooling. The contracting path keep on doing the same operations multiple times until the feature vector reaches into least dimensions or flatten into 1D array.

As the feature vector reaches near one-dimension. Then pooling can also be bypassed and only convolution is performed. In this way, pure convolutional network is at the joining link between two parts of the U-Net. The joining link is called backbone of the U-Net. In different variants of the U-Net is backbone is often replaced by some other robust architecture like Residual Network. It is also evident that the backbone is responsible for making classification and shaping feature vector according which is then feed into decoder part, whose job is to find the location of the feature into the original image.

Expansive path or Decoder

The other side of the U-Net consist of expansive path or decoder. Decoder is responsible to locate the context of the feature in the original image. In other words, it is performing localization of the classification obtained from the encoder. Primarily, decoder consist of four fundamental layers. Among these layers two are convolutional layers, one is the inverse convolutional layer also called transpose convolutional layer, and the last layer is concatenation layer. As the nearly flatten vector is received from the encoder then in the very first step it is transposed convolved followed by concatenation with the information received from the encoder through skip connection, which is then convolved two times in a row. In this entire process the image is being up-sampled only by transpose convolution, and none of the layers in the decoder reduces dimension of the incoming image or feature which expands the dimensions and thus in the end, the final mask obtained is exactly of same size to that of the input size of the image.

In the above block diagram, the upward arrows are showing transpose convolution, and the arrows between the same color blocks are for convolution only, whereas the arrows between different color blocks are responsible for concatenation or is the skip connection coming from encoder side

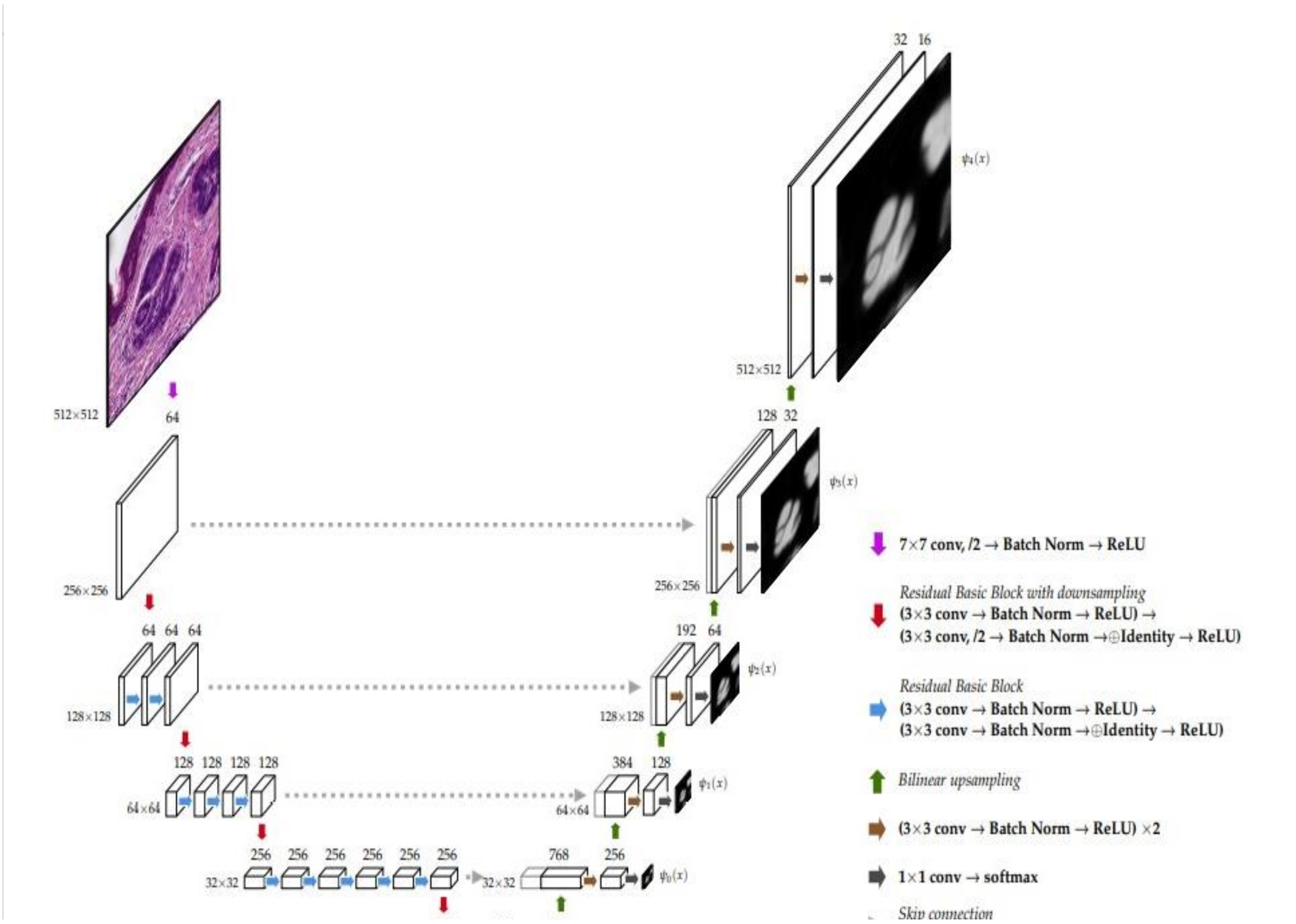


Figure 3.1.2.2.1.4.: U-Net architecture for Lung Cancer Detection [26]

The above figure is the complete 3D diagram of the U-Net which is being used for lung cancer detection. In the above architecture, Residual Network is also used as the backbone of the network, which give better quality results in classification and thus is more significant in overall tumor detection. In the above diagram, addition 1x1 convolution is performed followed by softmax activation only for the purpose of visualization of the output mask at each step of the decoder. In the above diagram it is apparent that context information is also being up-sampling and expanding along with the mask.

3.1.2.3. Data Preprocessing

In the very first step of training any machine learning model is the pre-processing of the input data. In real time machine learning project, it is the most fundamental and crucial step, and has the most influence on the final results. Thus we have spent quality time in data preprocessing. Data preprocessing comprised of following fundamental steps, i.e. Data split, Tessellation, and Augmentation. In the last step, final preprocessed images are also shown using image Data generator.

3.1.2.3.1. Data Split

In the very first step before doing any rigid preprocessing it is imperative to split data into training validation and testing batches for better results. As we are dealing with WSIs thus we need to split them into training and validation folder on the basis of WSI as well. The dataset has 150 WSIs, 80% of these images are used for training and remaining 20% images are used for validation. Due to large training time requirement the model is being trained on patches rather than WSIs. Random patches are selected for training and validation in the exact same ratio. The training and validation patches are moved out the folder and the remaining patches provide the dataset for the testing. It is kept in mind that all patches are cancerous.

Following table shows the number of dataset split as well as the system specification for the training the U-Net.

Table 3.1.2.3.1. Detection Data Split and Machine Specification

Dataset	Training (80%)	Validation (20%)
ACDC-Lung Cancer	160,000	40,000
Total	200,000	
Machine Specification		
RAM	64GBs	
Hard Disk	6TBs HDD – 1TBs SSD	
GPU	RTX A6000 Quadro (24GBs + 33GBs)	
Cores	40	

3.1.2.3.2. Tessellation

In biomedical image processing, the process of cutting large image into small images is called tessellation. As mentioned that enormous dimensions of the WSI are both difficult to process and is computationally expensive as well. Thus WSIs are cropped into small patches of fix dimensions for training the model. Another plausible reason for the need of tessellation is that the dimensions of the input image also governs the depth of the architecture. So small dimensional image will also make architecture less deep.

There are numerous means of making small patches of the large image, but most easy among them is patchify. It is python based library dedicated for making patches of the large matrices, and has sufficient functions to give sense of control and custom size of patches as well. It is important to know that all the adjacent patches from the same WSI are mutually exclusive and none of the pixel overlaps in any two of the patches. In order to give a generic perspective about the size of the WSI, it is important to know that one image of about 1.4GBs can produce more than 0.28 Million patches of size 256 x 256.

Tessellation of the mask is also performed, so that each cancerous patch also has the corresponding mask patch. The mask patch will act as the ground truth for the cancerous patch. As the mask is binary so black region represents non-cancerous region and white region represent cancerous tissues.

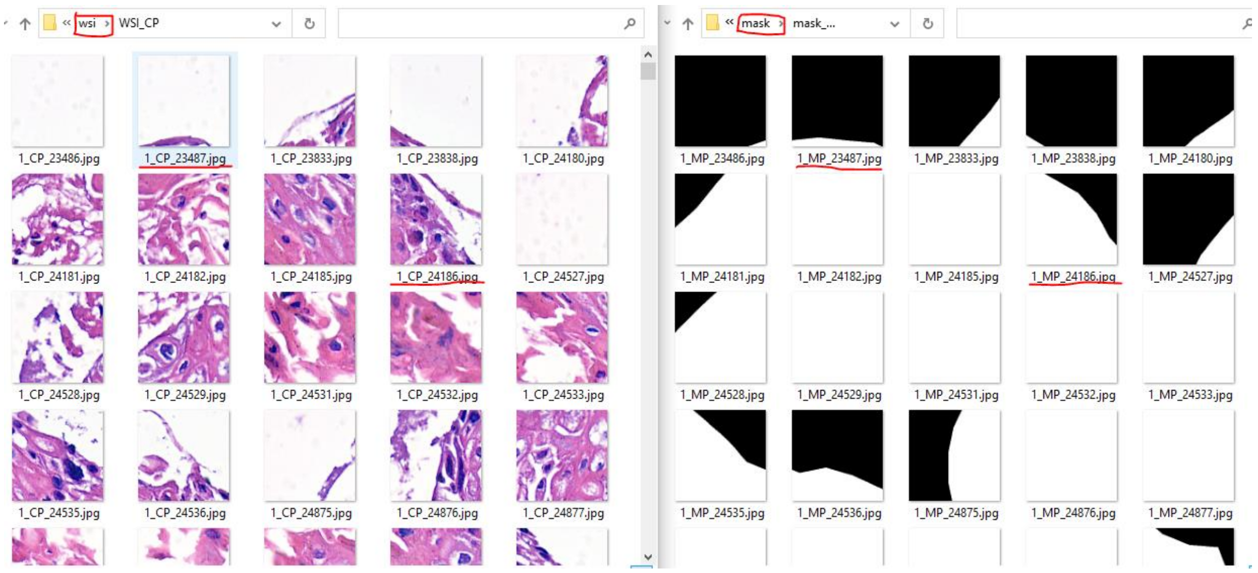


Figure 3.1.2.3.2.: WSI and Mask Tessellation

3.1.2.3.3. Augmentation

Augmentation is the process of increasing dataset by performing operations, ranging from simple to complex, on the already existing images and appending the changed images to the dataset as well. Due to tessellation, the architecture has great deal of data and can be trained even without augmentation as well. The ultimate reason for the augmentation is that abundant data do not ensures maximum range of the diversity of the dataset. Augmentation enhance diversity of the dataset and also contribute to the generality of the model in terms of learning. Augmentation always have positive effects and is also a good source to avoid overfitting of model in dataset. It is recommended in low data training but for the purpose generality it is a good resolve.

In our case we have use augmentation in ratio of 1 to 7, meaning, we have applied seven different operations on one image and then obtain seven new images which are then append to the dataset.

In simple words this augmentation will increase the dataset by seven times. Following table summarizes the augmentation applied to the dataset.

Table 3.1.2.3.3.1.: Detection Dataset Augmentation

Augmentation Ratio	1 : 8	
Augmentation Operation	Sample wise centering Sample wise standard Normalization	
	Shear range	0.02
	Flip	Horizontal Vertical
	Brightness range	0.4 to 1.5
	Zoom range	0.3

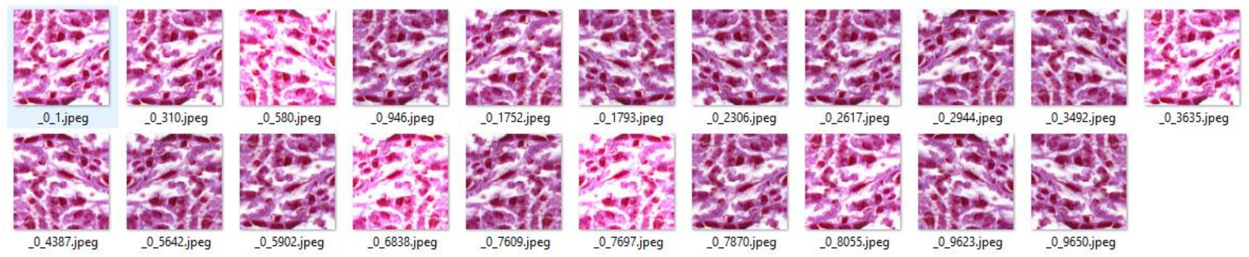


Figure 3.1.2.3.3.: Output images of the dataset obtained after augmentation for one image

3.1.2.4. Hyper Parameters and Model Training

Hyper-parameters are the peripheral parameters required by every machine or deep learning model. In ML or DL parameters are the variables which can be derived from the data itself or they have some relationship with it, either direct or indirect. Whereas, hyper-parameters are the variables which do not have relationship with the data or cannot be derived from the dataset, it is completely the objective choice of the user to decide on its own, but just parameters hyper-parameters are also optimize to fit the output to the desired results. Most prominent hyper-parameters are epochs, batch size, optimizer, learning rate etc. Simple example of parameter is steps per epoch, which is calculated using the formula and is dependent on the batch size as well. In this way, the parameter of steps per epoch has indirect relationship with dataset to some extent.

Following table summarizes the hyper-parameters used in training of U-Net.

Table 3.1.2.4.1.: Detection Hyper Parameters

Model	U-Net
Optimizer	Adam
Learning Rate	0.001
Loss	Binary Crossentropy
Metrics	▪ Accuracy ▪ Recall ▪ Precision ▪ MeanIOU
Epochs	30
Batch Size	32
Model Store Name	Unet_B2_v1.h5
verbose	2

3.1.2.5. Web Application Deployment

In this section the integration process of the model is being discussed. The backend is being developed by using the Flask (Python) while front end is being developed by using HTML and CSS. This section explains how simply we can integrate the Model into Web application. For integration we have used Visual Studio Code.

3.1.2.5.1. Requirements for web application

We need following things need to be done before Web application.

- H5 model: Trained Model must be saved in h5 form. H5 format saved model weights and its configuration in single file. We saved Model in H5 from. Both models i.e. UNET and Resnet are in h5 format.
- All Packages must be installed on the system. Installed packages with pip install packages. So we needed Flask, Keras, tensorflow, werkzeug, `matplotlib, shutil, numpy and other packages.
- Creating folder named as web application. This folder have two sub folders named as templates and static folder.

3.1.2.5.2. Backend of Application

Backend of application is being developed on FLASK. Flask is Python framework that makes the web development easier. We have to make app.py file in folder named as Web application. The h5 are being saved in same directory as that of app.py file. The Model is loaded by using load_model function. When we load model by using the load_model function, it can be used for prediction by using model.predict function. But we have to keep in mind that model.predict can only have certain type of input image whose size is equal to the image size on which model is being trained. Output of Model.predicit is prediction among the classes present in the Data Set.

Flask application has different routes and each route can have output. We render HTML template as output of the Flask and output. This is how backend is connected to the front end of the web application. We made two routes i.e. classify and detect. Image is passed by using Multipart request 'POST' method from front end to the back end of the web application. The Request return a response, the response in both method is different.

- **Classify:** When image is posted to classify route, Image is being loaded by load_img and is converted into array and reshaped to size of (224,224,3) . This image is then passed to the resnet50.predict, which gives the output as probability of each of the three classes i.e. Adenocarcinoma, Benign, and Squamous Cell Carcinoma. The Highest probability will be returned as the class of the tissue.
- **Detect:** When Detect is route. The loaded IMAGE is in reshaped to (256,256,3) size. The image is then normalized by dividing by 255. Then normalized version is then passed to the unet.predict function. The output of the function is probability mask (2D). We have used threshold of 0.75 i.e. if the probability of having cancer is more then 0.75 then give it then make this pixel value as dark otherwise the pixel value must be light value. This threshold

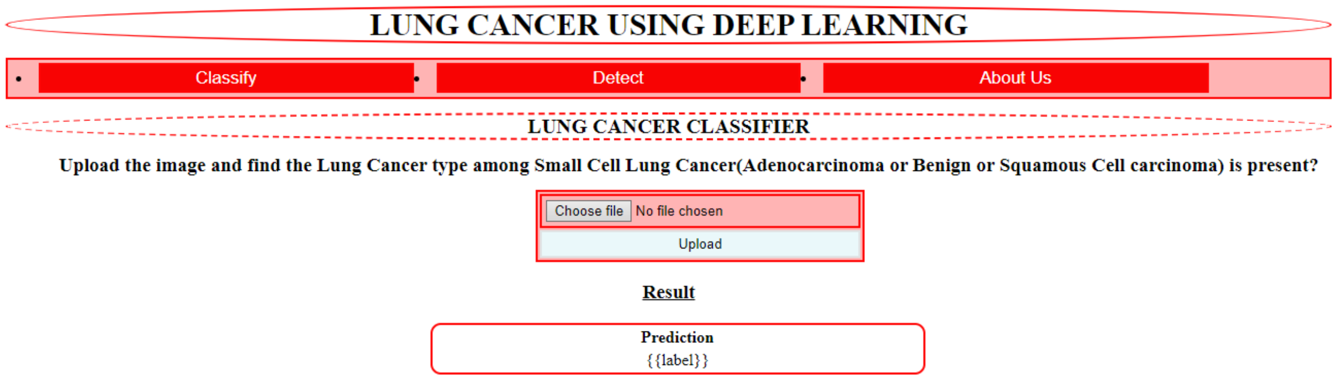
value can be varied as different images give better mask at different threshold in refrence to ground truth masks. This mask is saved by using matplotlib.save function to the web application folder. Then we copied the mask shutil.copy function to the static folder of the web application folder. Flask can only send images to front end if is in the static folder. So the Mask and Image must be saved in the static folder to be viewed on the front end of the Html page.

3.1.2.5.3. Front End of application

The front end of application has two pages named as the Classify and Predict. Each HTML page is for specific route of the FLASK application. The pages html are being saved in the templates folder of the web application. If we want to get the image back end to the front end, it must be present in the static folder of tour application. HTML is being used to interact with the FLASK and CSS is being used for templating the web page into better representation.

3.1.2.5.3.1. Classification page

Classify page will post image by multipart request method to the back end. The form is present, By clicking on it, user can upload image and send it the backend of the flask application. Below the form, there is prediction table, as response of the classify call, the prediction is printed in the prediction table.


LUNG CANCER USING DEEP LEARNING

LUNG CANCER CLASSIFIER

Upload the image and find the Lung Cancer type among Small Cell Lung Cancer(Adenocarcinoma or Benign or Squamous Cell carcinoma) is present?

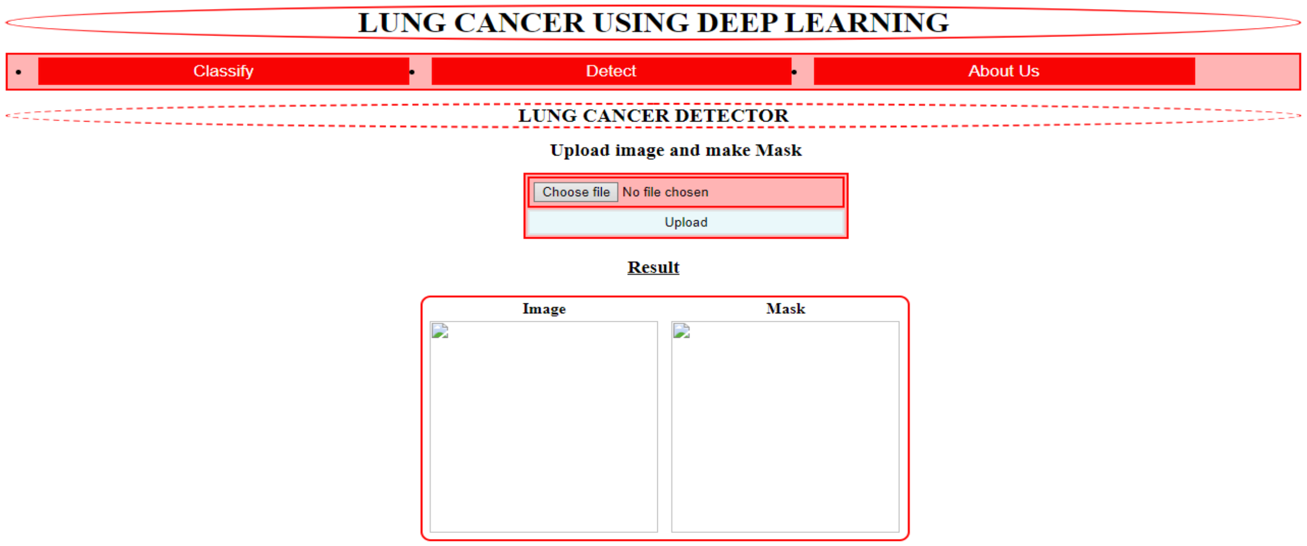
Result

Prediction
{{label}}

Figure 3.1.2.5.3.1.1.: Classification page

3.1.2.5.3.2. Detection page

For Detection image is being posted by using multipart request post method and its mask is saved is the static folder of the Flask application. By multipart get request method, image is get to the front and image would be visible in the Result table. The darker part of the mask will be the cancerous region and lighter part of the image will be represent the non-cancerous part of the image. The page can be shown below as under:


LUNG CANCER USING DEEP LEARNING

LUNG CANCER DETECTOR

Upload image and make Mask

Result

Image
Mask

Figure 3.1.2.5.3.2.1.: Detection page

CHAPTER 4

Results

4.0. Classification

Residual Network known by the acronym ResNet with 50 layers model is used for classification purpose, which is trained on LC25000 dataset containing 15000 images of lung cancer in three classes. The model training is observed on one metric i.e. accuracy. The loss function is also monitored during the training.

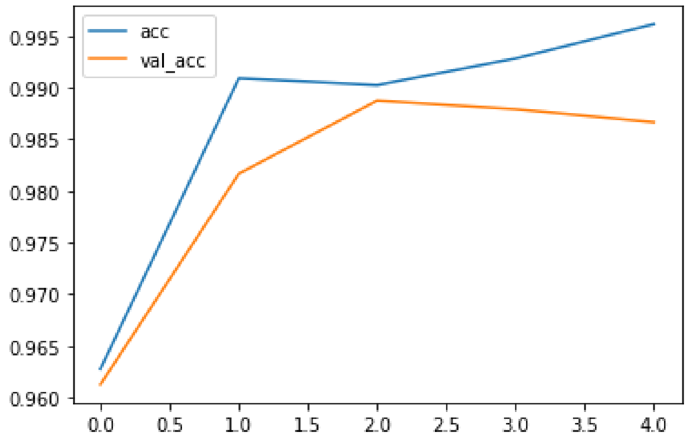


Figure 4.0.1.: Classification Accuracy

Due to transfer learning, the model is able to achieve phenomenal accuracy of 99% on the training dataset and also 98% accuracy on the validation dataset. The validation curve is below training, which shows good learning of the model on the new dataset as well.

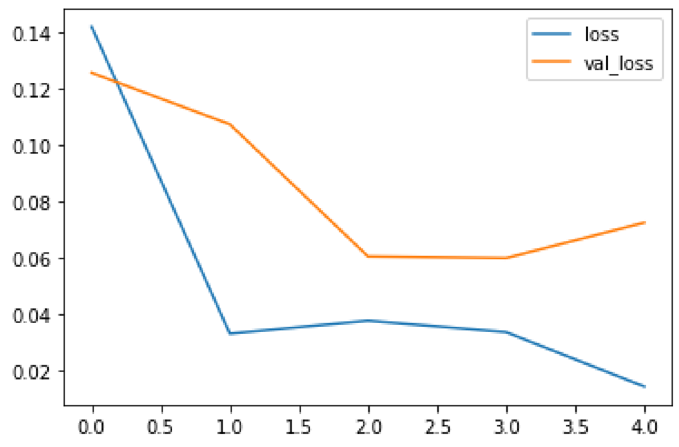


Figure 4.0.2.: Classification Loss

The loss function curve is also plotted in the above diagram which shows that that loss is reducing with each epoch and the minimum the last epoch has the minimum loss on the training curve whereas the loss is constant on the second and third epoch for the validation dataset. Validation dataset has higher loss than training loss thus model also has higher training accuracy than validation accuracy. And it can also be confirmed from the accuracy plots as well.

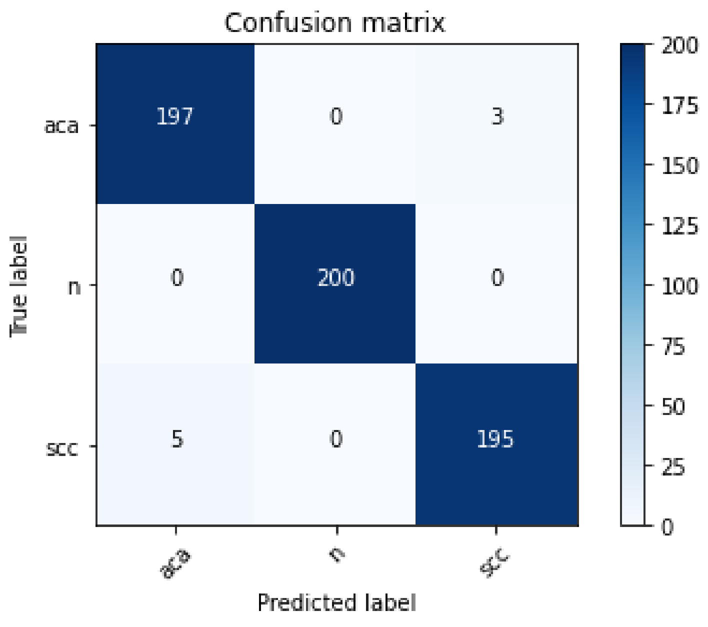
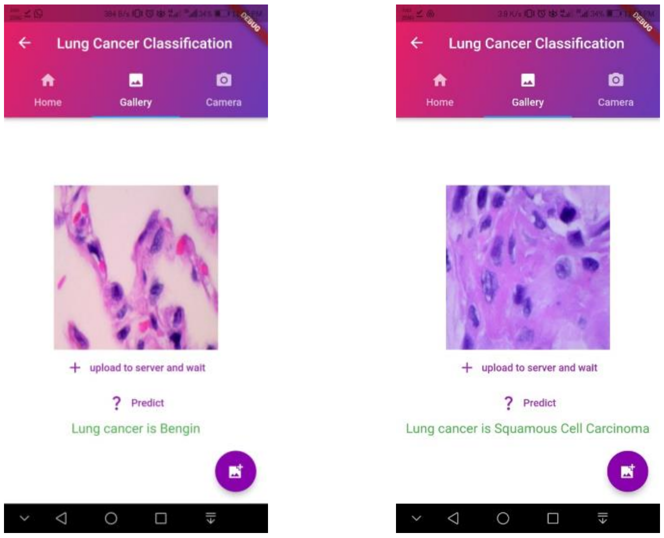


Figure 4.0.3.: Classification Confusion Matrix

As the resnet model is dedicated for three class classification so the results of test dataset are plotted in the form of confusion matrix. There are total 600 test images and 200 images belong to each class and the confusion matrix show that model has predicted correct class for 98% of the dataset. Mobile application classification results are shown below.

4.0.1. Mobile App Classification Results

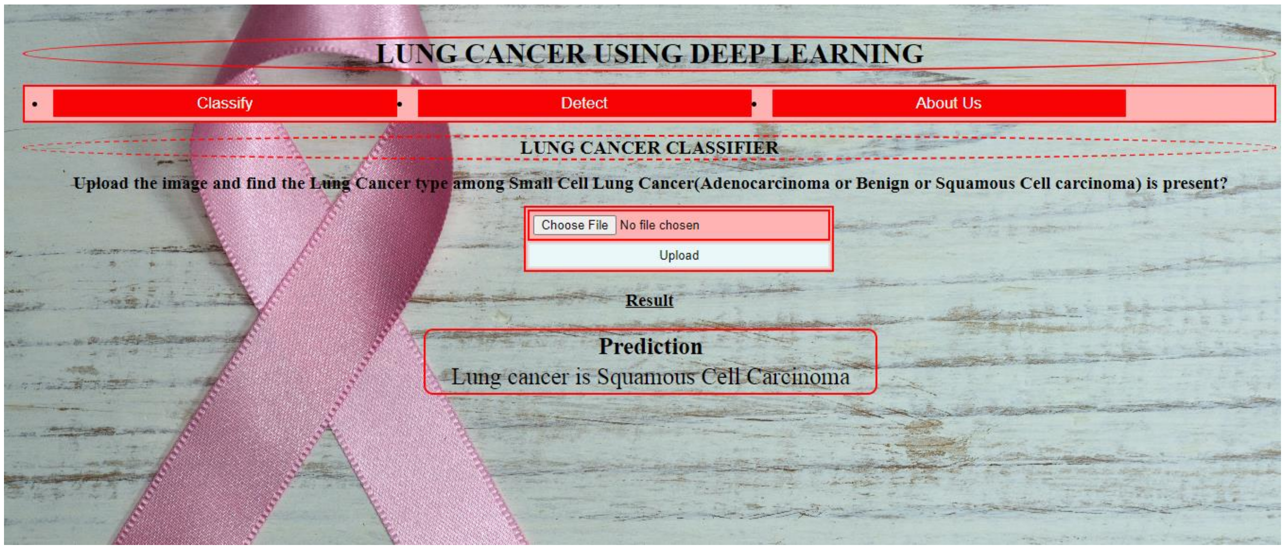

Lung Cancer Classification
Home
Gallery
Camera
upload to server and wait
Predict
Lung Cancer Classification
Home
Gallery
Camera
upload to server and wait
Predict

Lung cancer is Bengin

Lung cancer is Squamous Cell Carcinoma

Figure 4.0.1.1.: Mobile App Classification Results.

4.0.2. Web App Classification Results



Upload the image and find the Lung Cancer type among Small Cell Lung Cancer(Adenocarcinoma or Benign or Squamous Cell carcinoma) is present?

Lung cancer is Squamous Cell Carcinoma

Figure 4.0.2.: Web App Classification Results

4.1. Detection

In case of detection, U-Net model is analyzed on four metric during training, which took approximately a week for the training on the dataset. The results of the four metrics are tabulated below in the table. The model is able to achieve high accuracy of 93% on the training dataset and the 92.5% accuracy on the validation dataset. The model gives good performance on other metrics like precision and Recall. But relatively poor performance on Mean IOU score which is just 33% for training and 31% for validation dataset.

Table 4.1.1.: Detection Performance Metric's Results

Metric	**Training**	**Validation**
	Performance	**Performance**
Accuracy	93	92.50
Precision	93.40	92.26
Recall	100	100
MeanIOU	33.27	31.06

The training curve is plotted for the accuracy only and there is relatively close accuracy of the model on training as well as validation dataset. Due to relatively close accuracies of the model on traing and validation dataset it is evident that model is have precision, which is also evident from the tabulated results in the above table.

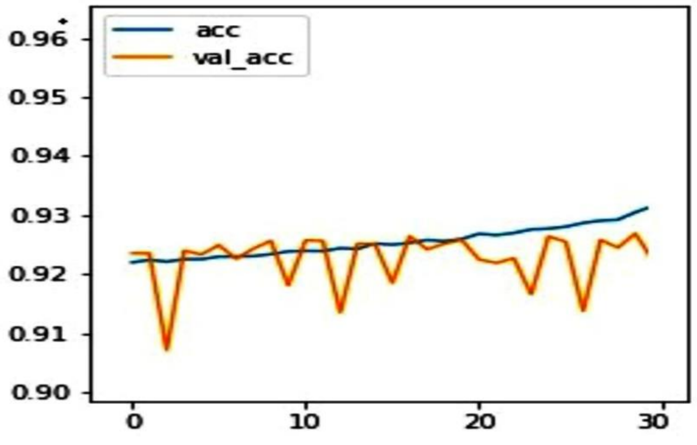


Figure 4.1.1.: Detection Accuracy

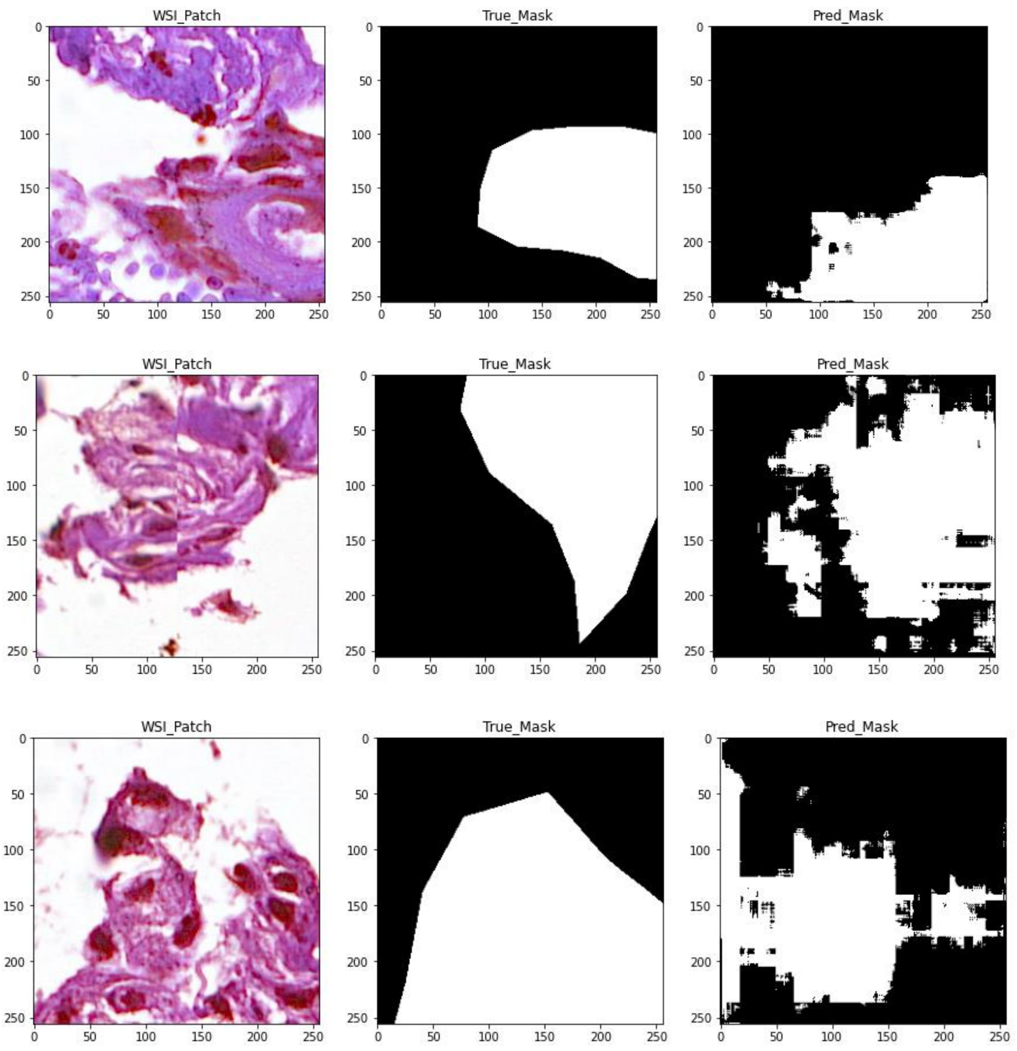
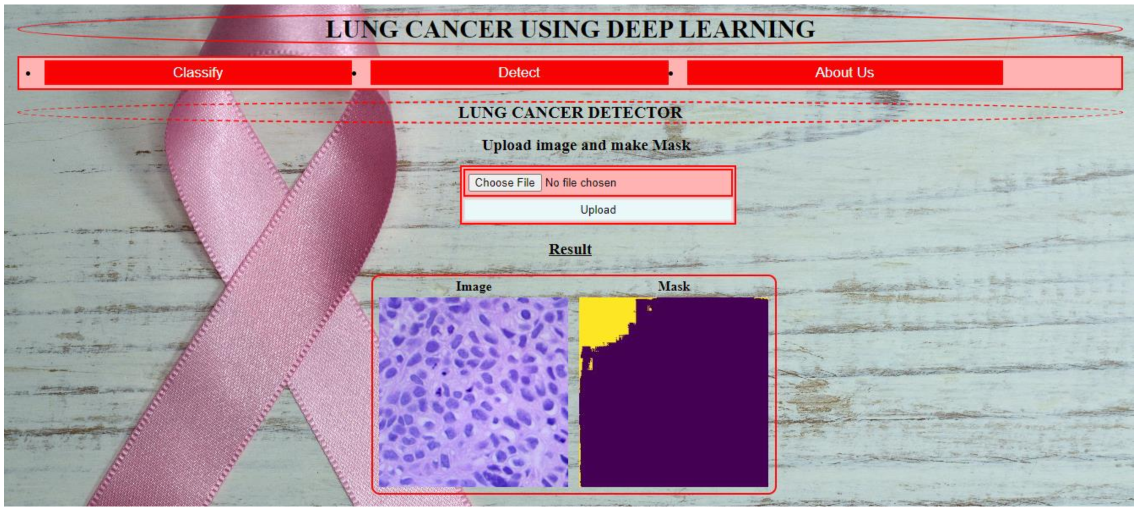
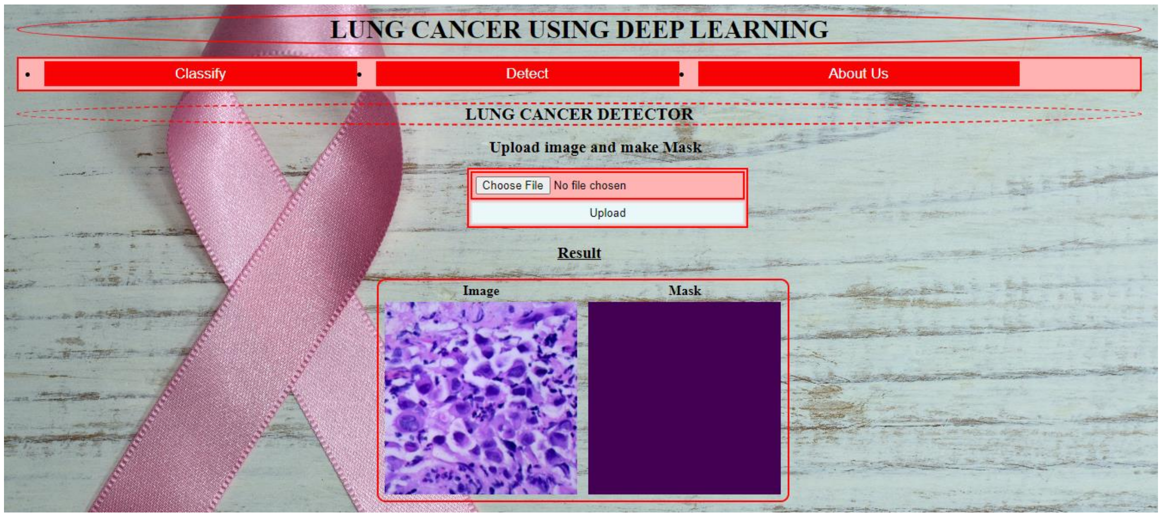


Figure 4.1.2.: Detection Ground Truth and Prediction

4.1.1. Web App Detection Results





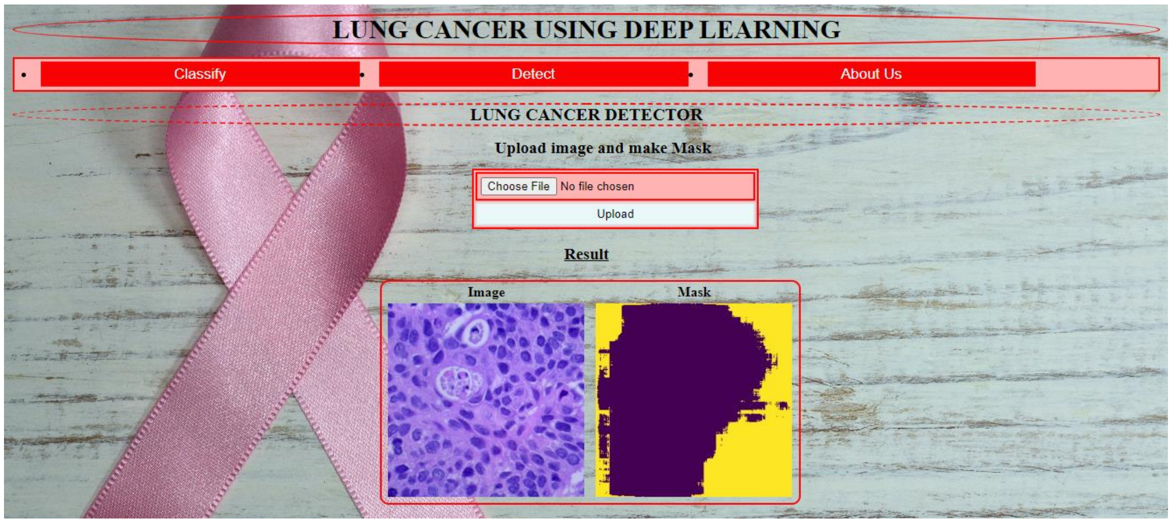


Figure 4.1.1.1.: Web Application Detection Results

CHAPTER 5
Conclusion

5.0. Classification

Classification model i.e. ResNet50 is trained for the feature vectors only and imagenet weights are being used. In simulation based application, without concern of real subjects, the results are significantly reliable and can be directly applied without taking significant error into account. As the model accuracy is near perfect thus model's prediction on individual test case will have negligible probability of error. But in the real time, when cancer patients are concerned it is imperative to cross validate the results using external or manual testing mechanism already in practice. Blind reliability over the model can have severe consequences and in the worst case can lead to life loss of the patient. The classification model is deployed on mobile application and smaller chunk of the histopathology image can also be used by the pathologist to check the type of the cancer. In other words, pathologist in real time situation can rely at maximum 10% on the application in terms of decision making.

5.1. Detection

It is apparent that predicted mask is not closely related to the true mask and has many fluctuations within the pixels. Detection results obtained are not significantly reliable and have larger error probability. There are primarily following reasons beyond the less similarity between true mask and predicted mask.

- The annotations provided for model training is subjective understanding of the disease and has immense error due to human factor involvement, as the pathologist cannot mark each pixel as cancerous and non-cancerous rather it takes into account large chunk of tissue to mark the region as cancerous.

- Low Mean IOU score also satisfies that the overlap between true mask and predicted mask has lower percentage as well, which is also evident from the true and predicted mask shown above as well.
- The cancerous region in the predicted mask have high probability of being cancerous and has rigid threshold.

CHAPTER 6
Future Work

The existing work can be extended in the following means.

- Improvement of Detection architectural model to obtain higher Mean IOU score.
- Improvement of mobile application to use robust digital tools for long term applicability of the app.
- Improvement in web application deployment to use CRM (Customer Relationship Management) platforms for the cloud based deployment like Salesforce, and Oracle Cloud Infrastructure (OCI).
- Improvement of peripheral software to provide upward as well as downward compatibility for the integration of pathologist’s toolbox integration.
- Connectivity with the centralized database can also be introduced into the application, which will initiate data accusation and in the long term will allow performance tracking.

6.0. Sustainable Development Goals, Goal 3: Good Health and Well being

According to Envision 2030 of United Nation’s sustainable Development Goals. It is desired to reduce maternal mortality to less than 70 per 100,000 live births. According to Global Con data, the lung cancer is more common among women than men so promoting mechanism and tools, which assist in automated lung cancer detection will surely assist in enhancing the mortality of the women.

References

[1] S. Hyuna, F. M. Jacques, R. ME, S. M. L, L. M. Mathieu, S. M. Isabelle and P. J. D. P. B. B. M. P. MSc, "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," *A cancer Journal for Clinicans,* vol. 71, no. 1, pp. 209-249, 2021.

[2] R. Cancer, "What is cancer," Cancer Research UK, [Online]. Available: https://www.cancerresearchuk.org/about-cancer/what-is-cancer. [Accessed 20 July 2022].

[3] P. K, M. R. M and P. Sumathi, "Analysis of statistical texture features for automatic lung cancer detection in PET/CT images," *International Conference on Robotics, Automation, Control and Embedded Systems (RACE),* pp. 91-96, 2015.

[4] S. Moritz, A. Daniela and J. M. Ferraro, "Automated detection of lung cancer at ultralow dose PET/CT by deep neural networks – Initial results," *Lung Cancer,* vol. 126, pp. 170-173, 2018.

[5] A. Worawate, T. Arjaree, M. Sanparith, M. Sanparith, W. Theerawit and W. Theerawit, "Automatic Lung Cancer Prediction from Chest X-ray Images Using the Deep Learning Approach," *11th Biomedical Engineering International Conference (BMEiCON),* 2018.

[6] T. Atsushi, T. Tetsuya, K. Yuka and F. Hiroshi, "Automated Classification of Lung Cancer Types from Cytological Images Using Deep Convolutional Neural Networks," *Hindawi,* 2017.

[7] M, Hutang, L, Zeidia. Malignancy Detection in Lung and Colon Histopathology Images Using Transfer Learning With Class Selective Image Processing, *International Conference on Robotics and Automation*, pp. 210-222, 2019.

[8] S, Sornesian, J, Wilton, et.al. CNN-based Method for Lung Cancer Detection in Whole Slide Histopathology Images, *International Journal of Medical Research, pp. 310-327, 2018.*

[9] J, Milicka, J,Brooks, et.al, Deep Learning Methods for Lung Cancer Segmentation in Whole-Slide Histopathology Images—The ACDC@LungHP Challenge 2019 *IEEE Transaction of Computer vision and Pattern Analysis*, pp. 121-142

[10] J,seidis, J.S. Munjasher, Convolution Neural Networks for diagnosing colon and lung cancer histopathological images (arxiv.org) [2009.03878]

[11] P. Al-jasar,K. Al-jaser, et.al, Detection of Lung Cancer Lymph Node Metastases from Whole-Slide Histopathologic Images Using a Two-Step Deep Learning Approach - *Journal* of science and medics, pp. 138-148, 2021

[12] M. Jamil, K. Jamal, et.al Artificial Intelligence in Lung Cancer Pathology Image Analysis *mdpi*, pp. 20-32, 2019.

13] A. T. S. M. a. T. W. Ausawalaithong, "Automatic Lung Cancer Prediction from Chest X-ray Images Using the Deep Learning Approach.," in *11th Biomedical Engineering International Conference (BMEiCON),*, Chiang Mai, 2018.

[14] T. T. K. Y. a. F. H. T. Atsushi, "Automated Classification of Lung Cancer Types from Cytological Images Using Deep Convolutional Neural Networks," in *BioMed Research International.* , 2017.

[15] H. D. Y. M. A. K. a. S. J. W. Rahane, "Lung Cancer Detection Using Image Processing and Machine Learning HealthCare," in *International Conference on Current Trends towards Converging Technologies (ICCTCT),*, Coimbatore, 2018.

[16] M. R. M. S. a. M. S. M. Šarić, "CNN-based Method for Lung Cancer Detection in Whole Slide Histopathology Images.," in *4th International Conference on Smart and Sustainable Technologies (SpliTech), Split.*, Croatia, 2019.

[17] M. B. B. R. S. S. Sasikala, "Lung Cancer Detection and Classification Using Deep CNN.," 2019.

[18] S. C. a. H. Rajaguru, "Lung Cancer Detection using Probabilistic Neural Network with modified Crow-Search Algorithm.," *Asian Pacific Journal of Cancer Prevention,* vol. 20, no. 7, pp. 2159-2166, 2019.

[19] B. Hatuwal and H. Thapa, "Lung Cancer Detection Using convolutional Neural Network on Histopathological Images," *International Journal of Computer Trends and Technology,* vol. 68, no. 10, pp. 21-24, 2020.

[20] Y. T. Y. H. Y. W. M. Z. G. &. Z. J. Zhu, "Feature selection and performance evaluation of support vector machine (SVM)-based classifier for differentiating benign and malignant pulmonary nodules by computed tomography.," *Journal of digital imaging,* vol. 23, no. 1, pp. 51-65, 2010.

[21] Q. Z. L. L. X. &. D. X. Song, "Using deep learning for classification of lung nodules on computed tomography images.," *Journal of healthcare engineering,* 2017.

[22] S. K. M. S. N. S. K. N. &. R. G. Lakshmanaprabu, "Optimal deep learning model for classification of lung cancer on CT images.," *Future Generation Computer Systems,* vol. 92, pp. 374-382, 2019.

[23] S. G. S. Garg, "Prediction of lung and colon cancer through analysis of histopathological images by utilizing Pre-trained CNN models with visualization of class activation and saliency maps," *arXiv,* pp. 1-12, 2021.

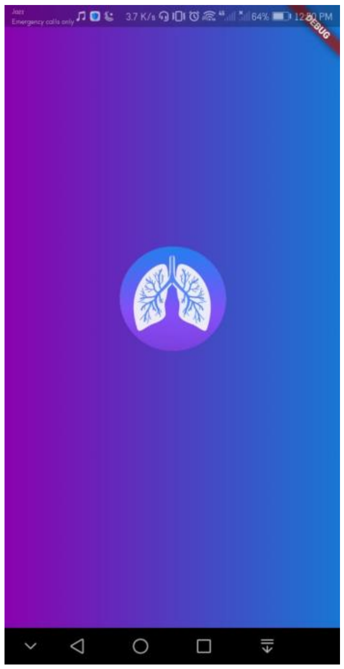
[24] M. Cocks, "Ethical Considerations in Pathology," *AMA Journal of Ethics,* 2016.

[25] J.P. chris, "Lung Cancer Detection using segformer detection mechanism.," *Journal of Cancer Research,* vol. 20, no. 7, pp. 215-228, 2019.

[26] Al-Saeeda,S, Al-Saeedi, "Lung Cancer Detection using mixed neural network with probabilistic techniques.," *Hindawi,* vol. 6, no. 12, pp. 322-328, 2021.

Appendix

Mobile Application Deployment Images are given with section caption.



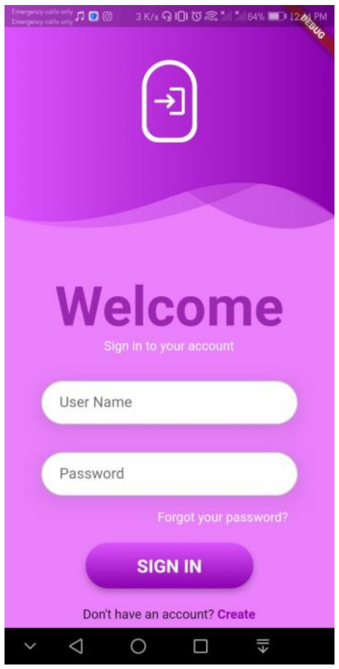
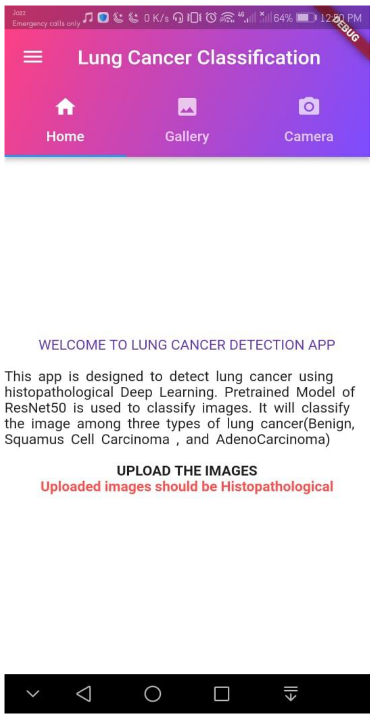


Figure A.3.1.1.4.12. SpalshScreen.dart | Figure A.3.1.1.4.13. login_page .dart



WELCOME TO LUNG CANCER DETECTION APP

This app is designed to detect lung cancer using histopathological Deep Learning. Pretrained Model of ResNet50 is used to classify images. It will classify the image among three types of lung cancer(Benign, Squamus Cell Carcinoma , and AdenoCarcinoma)

UPLOAD THE IMAGES

Uploaded images should be Histopathological

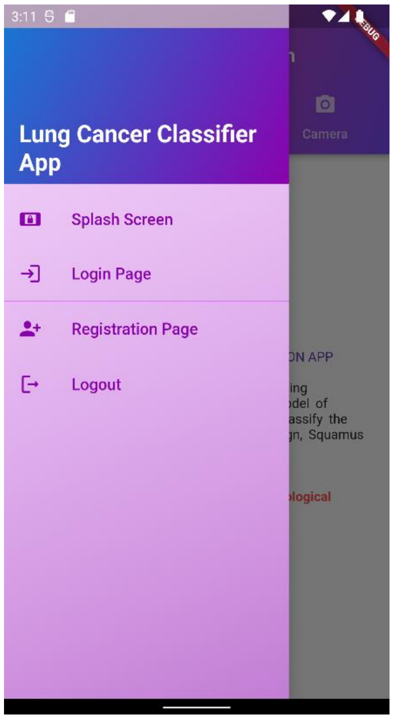
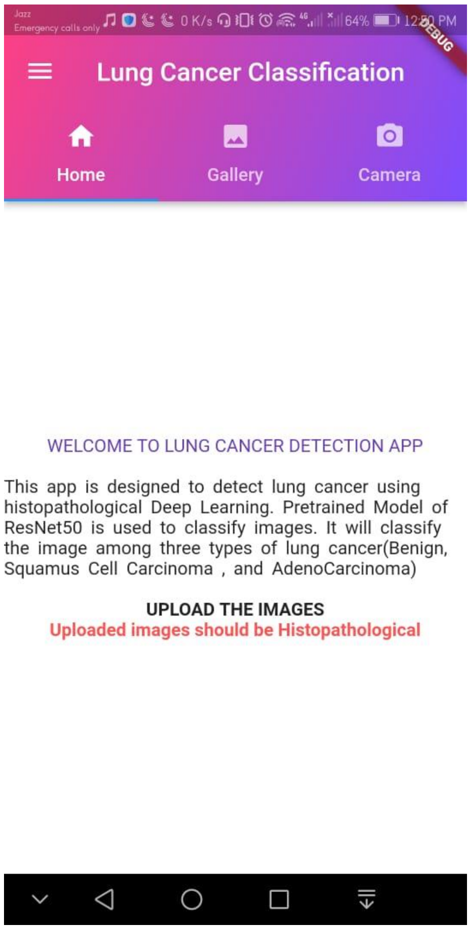


Figure A.3.1.1.4.14. (a) PredictionPage .dart (b) Drawer

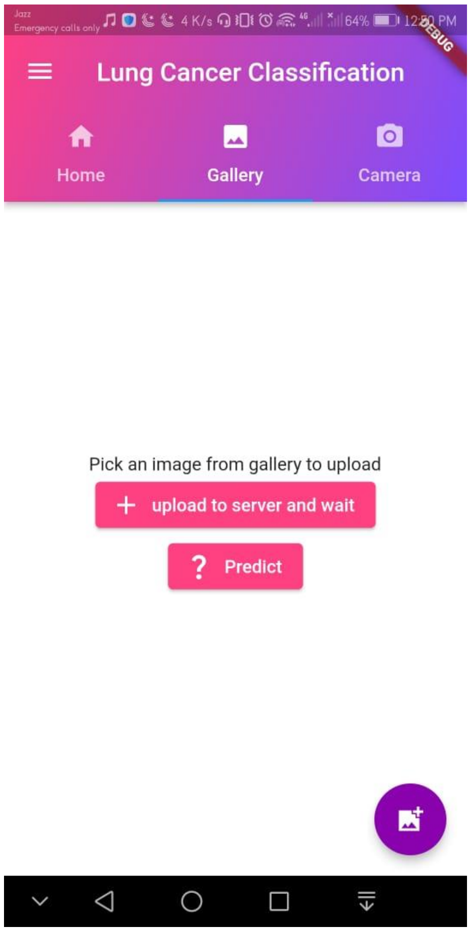


WELCOME TO LUNG CANCER DETECTION APP

This app is designed to detect lung cancer using histopathological Deep Learning. Pretrained Model of ResNet50 is used to classify images. It will classify the image among three types of lung cancer(Benign, Squamus Cell Carcinoma , and AdenoCarcinoma)

UPLOAD THE IMAGES

Uploaded images should be Histopathological



Lung Cancer Classification
Home
Gallery
Camera
Pick an image from gallery to upload
upload to server and wait
Predict

Figure A.3.1.1.4.15. HomePage .dart | Figure A.3.1.1.4.17. ThirdRoute .dart