

SUMMER RESEARCH FELLOWSHIP

2022

FINAL PROJECT REPORT

LUNG CANCER CLASSIFICATION USING DEEP LEARNING TECHNIQUES

Submission Date: September 8, 2022

Submitted by:

Gautam Prakash

National Institute Of Technology, Warangal

Supervisor

Dr. Ram Rup Sarkar

Senior Principal scientist,

Chemical Engineering and Process Development

Council of Scientific and Industrial Research,

National Chemical Laboratory(NCL) Pune

ABSTRACT:

As evidenced by real-world data, Lung cancer is the most prevalent due to excessive smoking, pollution, etc in today's world. Deep learning algorithms can help us with early identification of this terrible disease by significantly reducing time taken and automating the process. In this project, real world data from The Cancer Genome Atlas (TCGA) was used to identify Lung cancer using various Deep Learning techniques, specifically ResNets, to automate and speed up an essential process in the diagnosis of the disease. The data used is in the form of Whole Slide Images (WSI) which need to be preprocessed before they can be used for prediction.

ACKNOWLEDGEMENT:

The completion, success, and final outcome of this project work required a lot of guidance and assistance from many peoples and I am extremely fortunate to have got this all along with the completion of project work. Whatever is done in this project is only because of their guidance and assistance and I would not forget to thank them all. I would like to express my appreciation to Dr. Ram Rup Sarkar, Senior Principal Scientist Professor, CEPD, CSIR-NCL, Pune, for giving me an opportunity to do this project work and providing me with valuable support, advice, and guidance during this project work. I are extremely grateful to him for providing such nice support and guidance.

I would like to thank Mrs. Mudita Shukla, Research scholar-NCL, who has been kind enough for technical advice and help in there for everything. Finally, I thank the almighty God, without blessings of whose, nothing would be possible.

SECTION – 1: INTRODUCTION

Lung Cancer is the most often diagnosed of all cancers, accounting for about 5.9% of all cancers in India. One of the major causes of this is prevalent smoking throughout the world. According to the Journal of Thoracic Oncology, 80% of Lung Cancer patients are habitual smokers [1]. India being the second larger consumer of tobacco in the world, has a projected ratio of one in sixty-eight males developing lung cancer during their lifetime.

Like any other cancer, Lung cancer is treatable of found at early stages. However, due to lack of symptoms, it is often diagnosed later when there is high probability for metastases. Therefore, it is essential that the cancer is treated as soon as possible after symptoms begin to show. Traditionally, a cancer is classified by a physician using a whole slide image of the histopathological slide. However, with increasing cases there is a greater chance for misclassification. This can sabotage the entire process and put the patient's life at risk. This necessitates computer aided diagnostic techniques which if overseen by a physician can significantly reduce the probability of a misclassification. Here, presented pipeline will be helpful for physician to distinguish between normal and cancerous tissue.

Lung Cancer is broadly classified into two types: Small cell lung carcinoma (SCLC) and Non-small cell lung carcinoma (NSCLC). NSCLC is further divided into three subtypes, namely: Adenocarcinoma, Squamous-

cell carcinoma and Large-cell carcinoma. Adenocarcinoma is the most common among all of these affection about 40% of all lung cancer patients [2], it is also the most common type of cancer among non-smokers. It is usually found near the periphery of the lungs in a CT scan. Squamous-cell cancer affects about 30% of the patients and normally occurs close to large airways [3]. SCLC on the other hand is composed of dense neurosecretory granules because of which it gives endocrine symptoms and hence is easier to diagnose.

Though it can be extended to identify subtypes, the model developed only classifies the tumor slides into 2 classes, cancerous or non-cancerous [3]. The images used are those of histopathology slides of biopsy with a pyramidal structure and high resolution. Biopsy includes tissues stained with Hematoxylin and Eosin stains after which they are examined under the microscope.

1.1 PROBLEM STATEMENT

The objective of this project is to develop a pipeline to distinguish normal (benign) slides from cancerous slides using whole slides images of the same. This is done using the concept of deep learning and neural networks, specifically the Convolutional Neural Network (CNN). Since the data is irregular various preprocessing steps and regularization techniques need to be employed to effectively and efficiently classify the WSI's.

SECTION – 2: PROPOSED METHODOLOGY

The goal of the project is to develop a deep learning model for automated lung cancer detection using various Image Classification techniques. Different deep learning architectures were tried for this purpose. These architectures used the labelled images and applied various transformations on them to extract and learn different features and visual patterns. These features and patterns were then used to distinguish cancerous slides from non-cancerous slides.

The following figure (Fig 2.1) shows the various steps involved in the prediction process. It starts with importing the WSI's and applying various preprocessing steps on them including tile-formation, ideal tile-selection, normalization, deconvolution, nuclear segmentation etc. The final products of this pipeline (Tiles) are then split into training and validation sets to track the models progress. This is followed by choosing an ideal network architecture to classify these tiles, hyperparameter tuning and applying regularization techniques to maximize the accuracy of classification. After training this network, performance metrics evaluation are done to assess the reliability of the model.

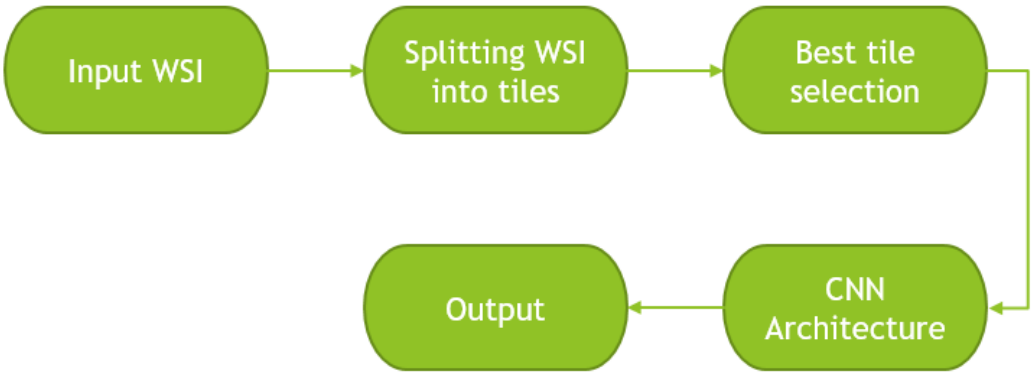


Fig 2.1 Block Diagram of Proposed Method

2.1 DATASET:

The data for this project was obtained in the form of whole-slide images (WSI) from The Cancer Genome Atlas (TCGA). Whole slide images are specialized images of microscopic slides which have a pyramidal structure composed of various resolutions of the same image reaching as high of a resolution as 15X or 17X. As a result of this composite high resolution nature, they are very large in size[4]. The sizes of the 2 slides used to train the model was about 1 GB (16000 X 21690 pixels) for the normal slide and 0.5 GB (14000 X 11987 pixels) for the tumor slide (Fig. 2.2).

A whole slide image is created by scanning a histopathological slide at high magnifications and combining the various constituent images to form a large image. Techniques include combining scanned square tiles into a whole-slide image and combining scanned strips into a resulting whole-slide

image. The images used to train and validate the model are given below (Fig 2.2).

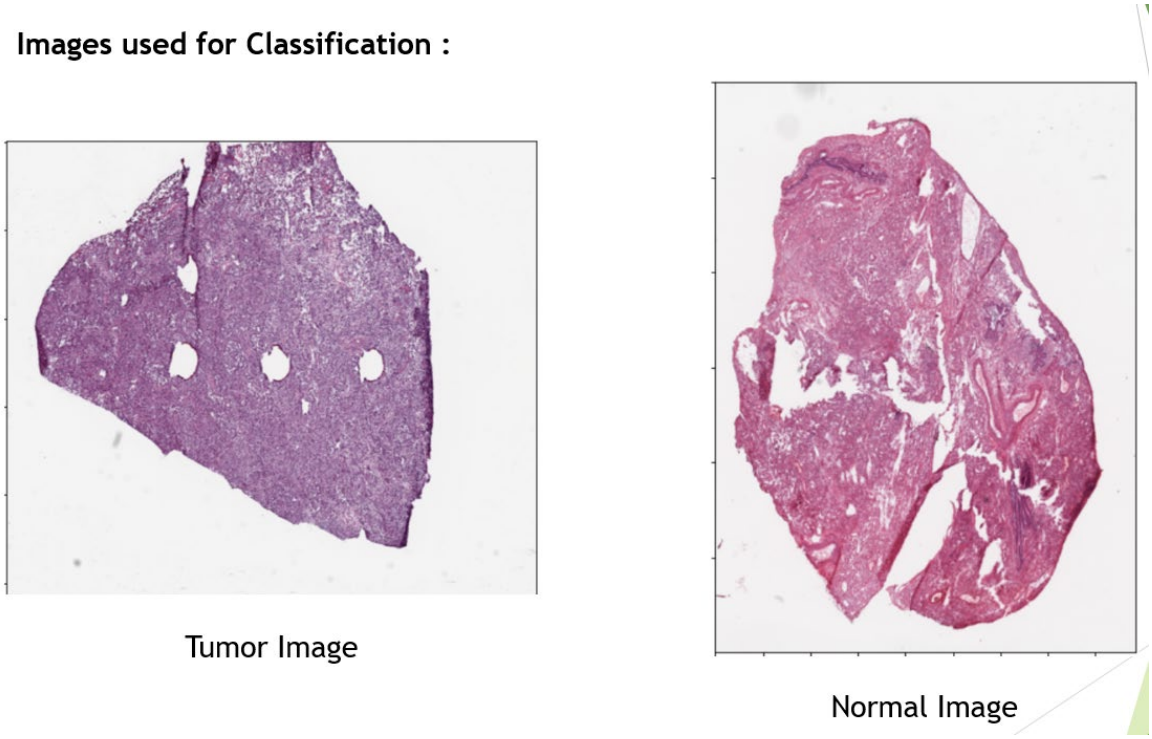


Fig 2.2 Images used to train the model

2.2 PYTHON LIBRARIES USED:

The computer language Python is being used to complete this project entirely. So, the list of libraries needed to finish this experiment is provided below.

- OpenSlide
- Numpy
- Pandas
- SciPy

- os
- PyTorch
- HistomicsTK
- OpenCV
- Matplotlib.Pyplot
- Scikit

2.3 IMAGE PREPROCESSING:

Before the image data could be processed in a deep learning framework, it had to be reduced to tiles of uniform size that could be fed to the network. To perform this operation DeepZoomGenerator class of Openslide was used. However, many of the tiles were plain white or contained very minimal tissues area. Hence, they had to be filtered based on their average pixel value and standard deviation so that majority of the area was of tissue. These tiles were further filtered based on the nuclear count threshold which ensured quality of tiles.

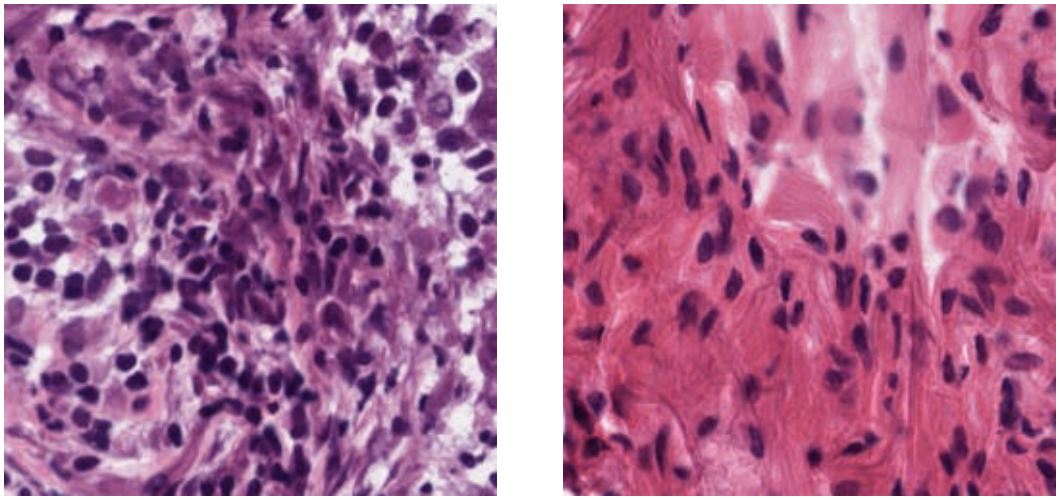
Since the defining characteristic of cells are nuclei, it was necessary to segment the tiles so that the nuclei were clearly visible. This was done in a step wise process described below.

2.3.1 CONVERTING WSI INTO TILES:

The enormous size of histopathology images makes working with them very difficult. Some histopathology slides could be as big as

100000x100000 in size which not only makes handling the image hard but also exponentially increases the computational complexity of the model developed for it. Due to this huge size, the feature extraction from histopathology slides has to be both memory and time efficient and the machine learning algorithm developed for it needs to be able to extract as much information as possible from it.

For this reason, Images were divided into tiles (Fig. 2.3) of reasonable size (256x256 or 512x512) which can be processed easily and doesn't hugely affect the computational complexity of the model. The DeepZoomGenerator module of OpenSlide Library (python) was used for this process.



(1) Tumour tile (2) Normal tile

Figure 2.3 Example of tiles Generated

2.3.2: SEGMENTATION

In our whole slide image, the nuclei are the defining characteristics of the cell which can be easily observed. Moreover, the nuclei can also be used

to distinguish between cancerous cells and non-cancerous cells since they differ in their shape, size, concentration etc. Hence marking the nuclei is important.

For this purpose, the HistomicsTK[5] library of python was used. The tiles were subjected to a series of processes (Fig. 2.4) as follows and the final output was the segmented tiles which could be fed into the CNN.

NORMALIZATION:

Staining variation and un-uniform concentration of stains is a common problem in whole slide images. This can lead to few tiles being heavily stained while others being lightly stained or stain concentration can vary within a slide as well. To correct this the image has to be normalized using the color profile of a properly stained image (target image). Reinhard normalization function has been used which normalized all the tiles to the color profile of a properly stained tile.

COLOR-DECONVOLUTION:

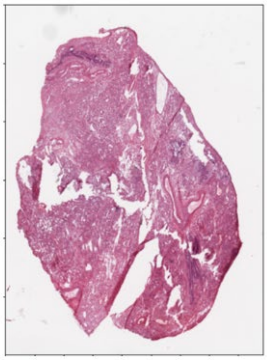
Normalized images that were obtained from the previous step have 2 constituent stains: Hematoxylin and Eosin. In order to observe the nuclei clearly these 2 stains need to be separated, this process is called color deconvolution. The Hematoxylin stained image shows clearly all the nuclei while the eosin stained image shows the structure of cytoplasm.

NUCLEAR SEGMENTATION:

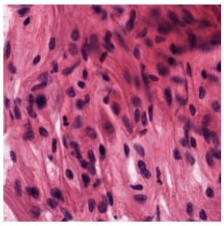
To mark the nuclei of the tiles with a color map, the segmentation was used from HistomicsTK. The Hematoxylin stained tiles were used for this purpose since hematoxylin stains mostly just the nuclei, hence the nuclei are more conspicuous.

Changes Observed after each step:

Imported .svs file:



Tiles Generated:



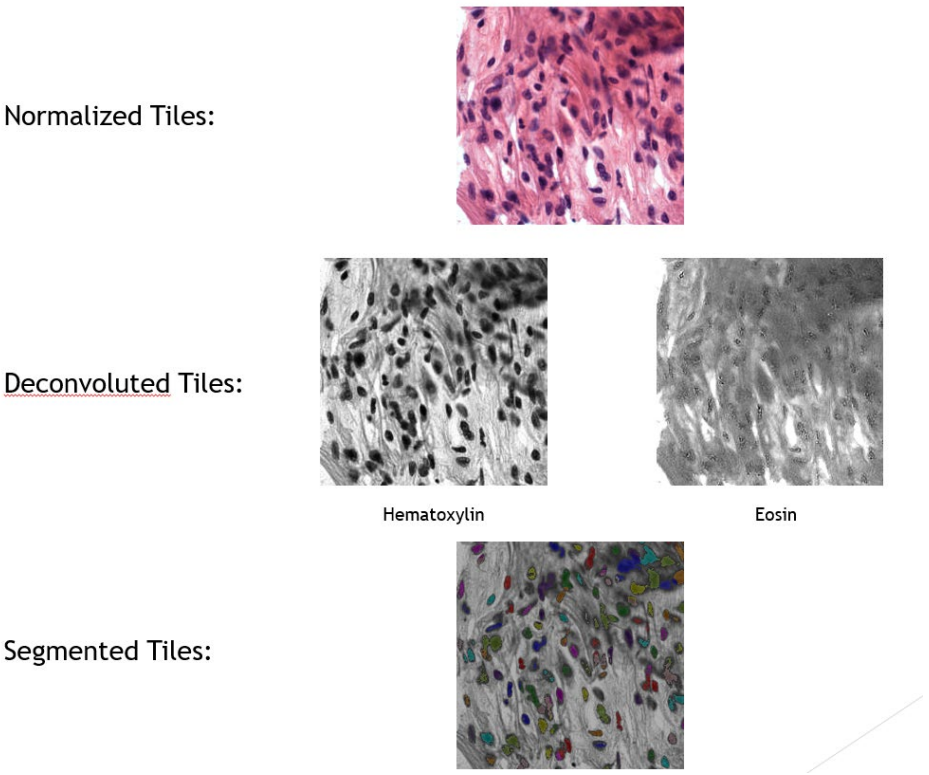


Fig 2.4 Changes observed after each step on a normal tile

2.3.3 STORING GOOD TILES FOR TRAINING:

The white tiles were initially removed using a threshold value of pixel mean value and standard deviation[6]. However even after this there were many tiles with majority of white area and very less tissue. These tiles were then eliminated using the len(objprops()) function from the HistomicsTK library which returned the number of nuclei in each tile. The minimum threshold for a tile to be valid was kept as 60 to obtain a decent proportion of images to train the model. The following table (Table 2.1) gives the number of tiles after each of the steps was performed.

Input	Process	Output
.svs file of Whole slide Image (Size: 20k X 10k pixels(normal) 11k X 10k pixels (Tumor))	Iteration over tiles of size 256x256 and selecting non-empty tiles	Colored tiles of size 256x256 (.png format) (Normal tiles: 2807 Tumor tiles: 1207)
Tiles from previous Step	Color Normalization using target image to correct staining variations	Same number of color normalized tiles of same size
Normalized tiles from previous step	Color deconvolution to separate different stains (Hematoxylin and Eosin)	2 sets of images in gray scale one of each stain
Hematoxylin stained images from previous step	Segmenting and counting the nuclei to select best images to feed in neural network	Segmented images selected based on a given threshold nuclear count (Normal tiles: 1216 Tumor tiles: 1034)

Table 2.1 Statistics of preprocessing steps

2.4 SPLITTING THE DATASET:

The 2250 tiles generated from the 2 WSI's had to be split into 2 sets in order to train the model and then validate the model. For this purpose, the randomsplit() function from pyTorch was used.

- TRAINING SET:

The training set would contain approximately 80 percent of the data available (both cancerous and non-cancerous). It is used to train the model, i.e the model parameters would be adjusted after each epoch in order to reduce the loss function value calculated on this set of data.

- VALIDATION SET:

The validation set contains data that is used to fine tune the hyperparameters of the model. This data helps generalize the model to

previously unseen data. If the training loss and validation loss vary by a large amount, it indicates high variance of the model trained and corresponds to a case of overfitting.

2.5 TRAINING AND CLASSIFICATION

Before the advent of Deep Learning, tasks like image classification were close to impossible. This is because machine learning models couldn't get neighbor information from a picture, they could only learn pixel level information.

Deep learning has helped achieve human level perfection in deep learning tasks using a technique called the convolutional neural network (CNN) [7].CNN is a type of deep learning model that learns representation from an image. This model can learn from low to high-level features without human involvement. The model learns not only information on a pixel level, the model also learns the neighbor information from an image by a mechanism called convolution. Convolution will aggregate neighborhood information by multiplying the collection of pixels in a region and sum them into a value. Those features will be used to classify the image into a class.

ResNet9 (Fig. 2.8) architecture was used to classify the cancer images from normal images[8]. ResNet9 is a convolutional neural network which layers of residual network which function as described by the image below.

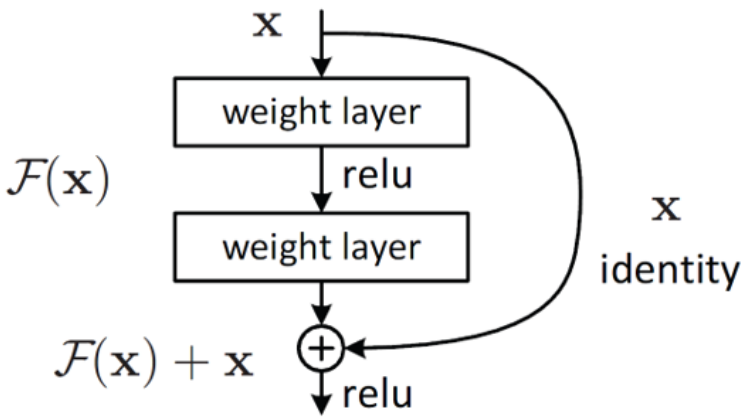


Fig 2.8 Structure of Residual Network

Deep learning is an area of Computer Science where math is used extensively. In the following section we understand the various mathematical tools used for classification along with their roles and advantages.

2.5.1 LAYER ACTIVATION FUNCTION:

Activation functions are used in hidden layers of neural networks. The primary purpose of these activation functions is to provide non-linearity without which neural networks cannot model non-linear relationships.

ReLU (Fig. 2.9) is the most commonly used activation functions in CNN due to the low computational load. It's equation is represented as

f(x) = max(0, x)

• Advantages of ReLU Function:

– The function is very fast to compute it doesn't calculate exponent.

– Converge very fast.

– Solve gradient saturation problem if input is positive.

• Disadvantages of ReLU Function:

– It is not a smooth gradient.

– ReLU function is not zero-centric.

– The major problem is that it suffers from dead neuron.

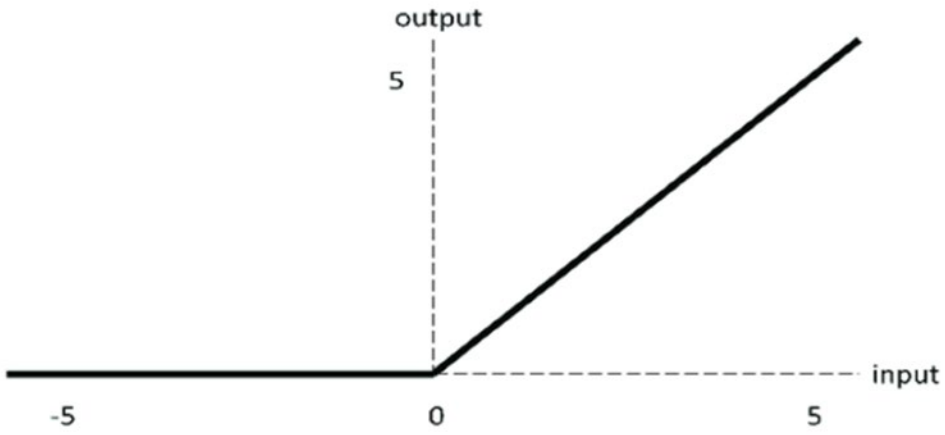


Fig 2.9 ReLU Activation Function

2.5.2 OPTIMIZER FUNCTION:

Optimizer is the algorithm or method used to reduce the loss function of the model by altering weights and biases of the model.

Adaptive moment estimator (ADAM) is an optimization technique or learning algorithm that is widely used. Adam represents the latest trends in deep learning optimization. Adam is a learning strategy that has been designed specifically for training deep neural networks. More memory efficiency and less computational power are two advantages of Adam.

2.5.3 LOSS FUNCTION:

Loss function is used to measure how inaccurate the model is. Higher the loss function value, more percentage of tiles are misclassified. The value of the loss function is further used to calculate gradients and alter weights and biases in order to make the model better after each step.

Binary Cross-entropy is the most common loss function used for binary classification tasks. It compares the true value and predicted value in order to generate the loss value.

2.5.4 REGULARIZATION:

For CNN models, over-fitting represents the central issue associated with obtaining well-behaved generalization[10]. The model is entitled over-fitted in cases where the model executes especially well on training data and does not succeed on test data.

To prevent this, Dropout technique has been used (Fig. 2.10). During each training epoch, neurons are randomly dropped. In doing this, the feature selection power is distributed equally across the whole group of neurons, as well as forcing the model to learn different independent features. During the training process, the dropped neuron will not be a part of back-propagation or forward-propagation. By contrast, the full-scale network is utilized to perform predictions during the testing process.

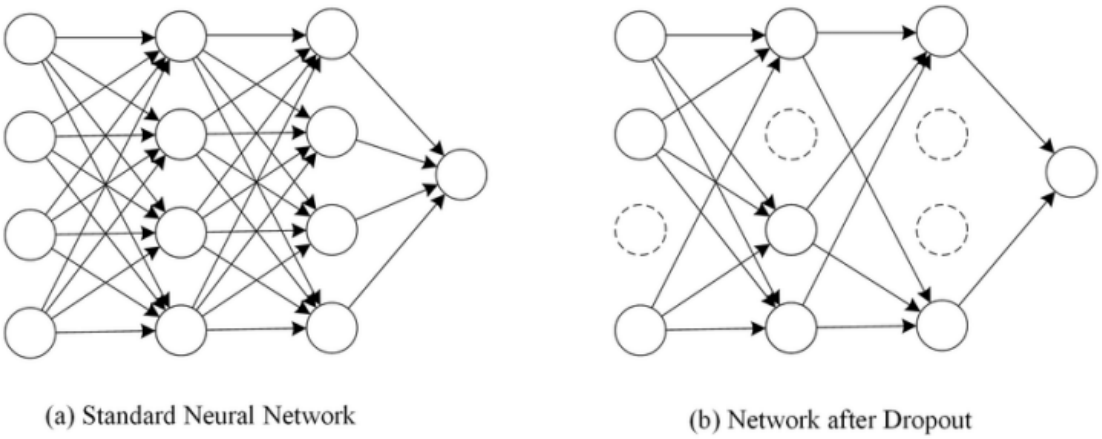


Fig 2.10 Example of Dropout

2.6 PREDICTION:

During the training process, the model generates a classifier that is dependent on the last output layer of the CNN model[11]. This classifier is converted into index value and arithmetic value by using Numpy library, and with the help of these index values and arithmetic values, the prediction of unknown images is done with the help of CNN model.

2.7 CNN ARCHITECTURE:

The CNN architecture used for classification was the ResNet9 architecture (Table 2.2). It contains a series of convolutional, pooling and residual layers as shown below. The final layer includes a linear layer that takes in the output of all previous layers in order to make a prediction. The following table (Table 2.2) shows the architecture of the network along with the input and output dimensions of vectors in each layer.

Function	Input	Output
Convolutional Layer	3x256x256	64x256x256
Convolutional Layer	64x256x256	128x256x256
Max-Pooling Layer	128x256x256	128x64x64
Residual Layer	128x64x64	128x64x64
Convolutional Layer	128x64x64	256x64x64
Max-Pooling Layer	256x64x64	256x16x16
Convolutional Layer	256x16x16	512x16x16
Max-Pooling Layer	512x16x16	512x4x4
Residual Layer	512x4x4	512x4x4
Max-Pooling Layer	512x4x4	512x1x1
Linear Layer	512	2

Table 2.2 CNN Architecture

SECTION - 3: RESULTS

On passing the training data through the model and checking the accuracy of validation data. An accuracy of 95% (Fig. 3.1 & 3.2) was achieved of validation data. The graphs of the validation accuracy and loss across the number of epochs are as shown below.

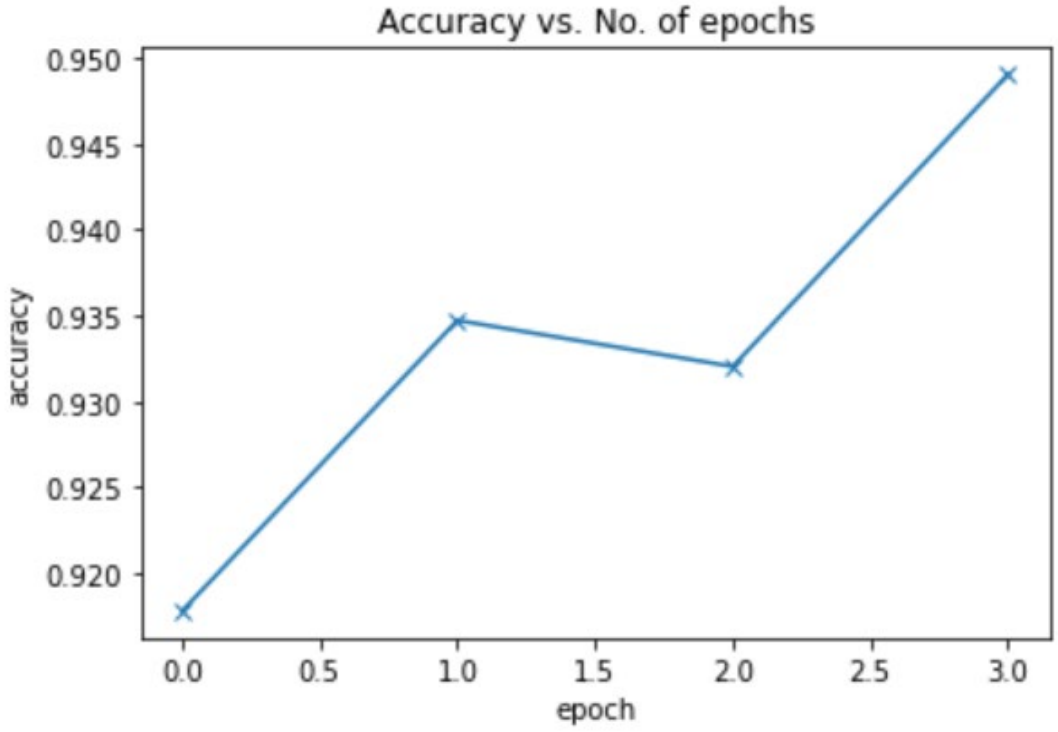


Fig 3.1 Loss vs Epoch Graph

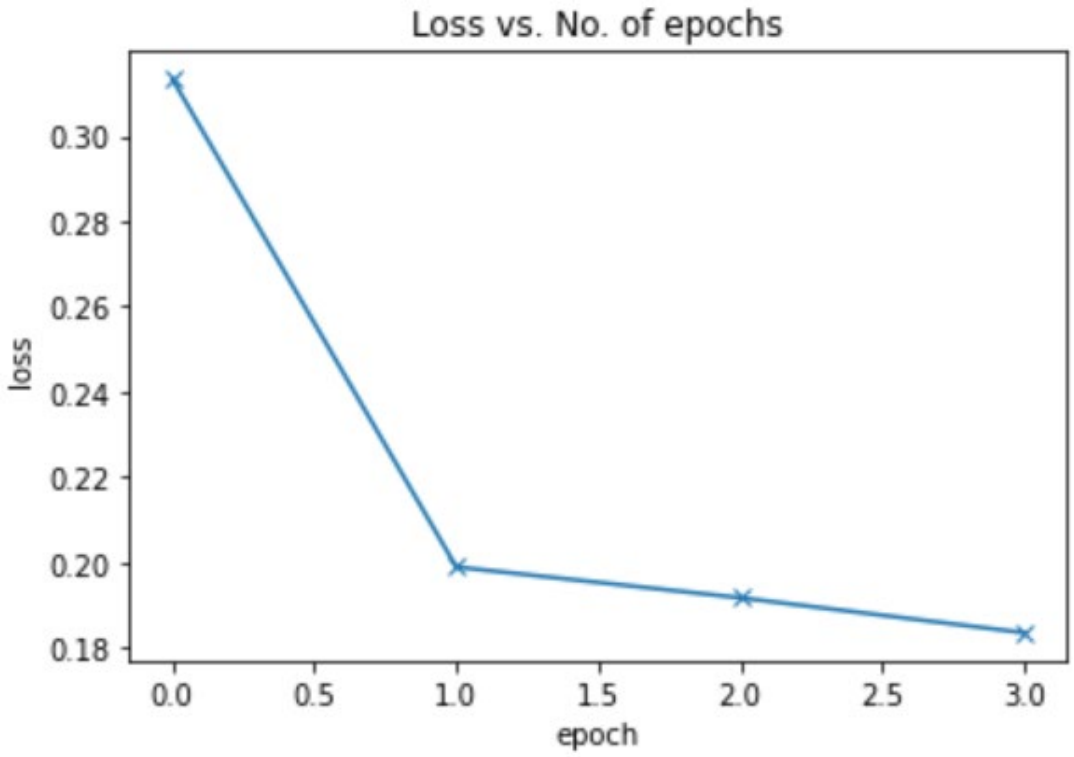


Fig. 3.2 Loss vs Epoch Graph

If the model was run for more epochs, it starts overfitting the model in accordance with the training data which gives rise to poor validation accuracy. In order to combat this more training data is required and better regularization strategies need to be applied.

3.1 BLIND TESTING

Blind testing is the process of passing previously unseen data through the model. In the case of whole slide images, this is done by breaking the image into tiles and then passing all tiles through the classification model. Following this the percentage of slides that have been classified into the correct class is calculated and is reported as the accuracy.

The following blind image (Fig. 4.3) was passed through the model after training it for 4 epochs.

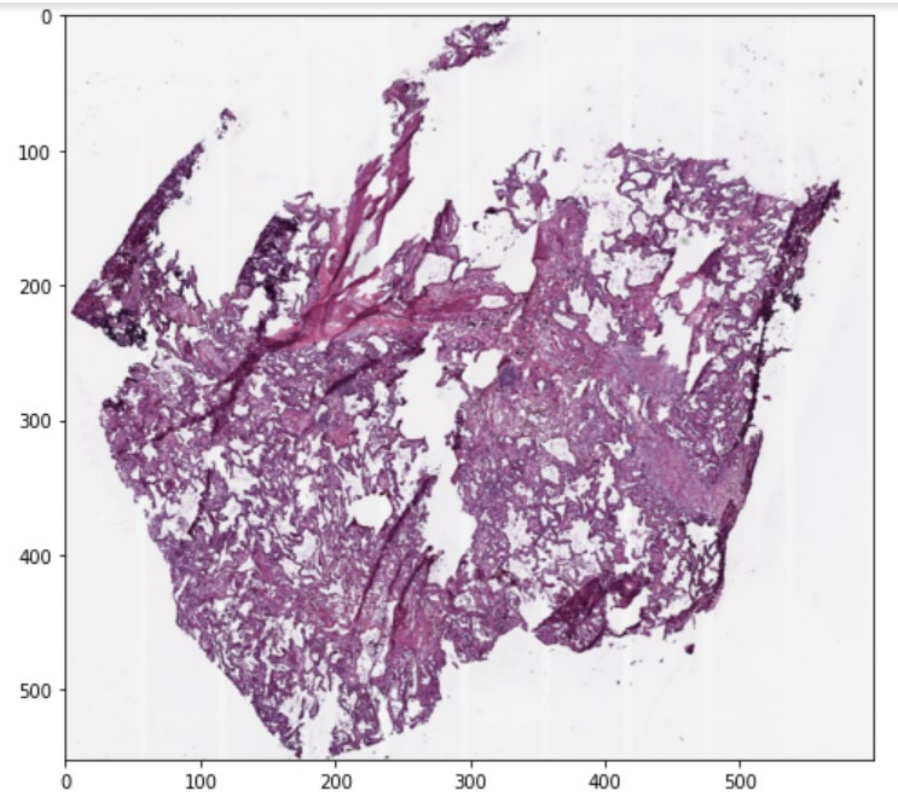


Fig. 4.3 WSI used for Blind Testing

On passing this image through the model, 37.7% of the tissue area was predicted to be cancerous.

SECTION - 4:
FUTURE SCOPE

Lung cancer detection by using digital/digitized histopathology images is a milestone in the field of medical pathology. It has also opened a door to new opportunities for research as there are many undiscovered areas that can be revealed by techniques and tools of machine learning and deep learning.

In this project, CNN model was used for the extraction of features from brain tumor histopathology image datasets. Though histopathology slides are of vast size, we only need to extract the good tissue part in order to train the model. Experiments demonstrate that this framework achieves an accuracy of 97% for classification.

Some of the things that can be further explored include

- Classifying lung cancer into its subtypes, if found cancerous.
- Using gene information too, in order to combine their results and making a more accurate prediction.

REFERENCES

1. WHO: World Health Statistics 2019: Monitoring Health for the SDGs. Geneva, Switzerland, World Health Organization, 2018
2. Parkin DM : The evolution of the population-based cancer registry . Nat Rev Cancer 6 : 603 - 612 , 2006
3. Swaminathan R , Selvakumaran R , Esmy PO , et al : Cancer pattern and survival in a rural district in South India . Cancer Epidemiol 33 : 325 - 331 , 2009
4. de Camargo B , de Oliveira Santos M , Rebelo MS , et al : Cancer incidence among children and adolescents in Brazil: First report of 14 population-based cancer registries . Int J Cancer 126 : 715 - 720 , 2010
5. Pongnikorn D , Daoprasert K , Waisri N , et al : Cancer incidence in northern Thailand: Results from six population-based cancer registries 1993-2012 . Int J Cancer 142 : 1767 - 1775 , 2018
6. Bray F., Ferlay J., Soerjomataram I., Siegel R.L., Torre L.A., Jemal A., Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA Cancer J Clin. 2018; 68: 394-424
7. Singh N., Aggarwal A.N., Gupta D., Behera D., Jindal S.K., Unchanging clinico-epidemiological profile of lung cancer in North India over three decades. Cancer Epidemiol. 2010; 34: 101-104

8. https://www.mayoclinic.org/diseases-conditions/lung-cancer/symptoms-causes/syc-20374620
9. Horn L, Lovly CM (2018). "Chapter 74: Neoplasms of the lung". In Jameson JL, Fauci AS, Kasper DL, Hauser SL, Longo DL, Loscalzo J (eds.). *Harrison's Principles of Internal Medicine* (20th ed.). McGraw-Hill. ISBN 978-1259644030.
10. Lu C, Onn A, Vaporciyan AA, et al. (2017). "Chapter 84: Cancer of the Lung". *Holland-Frei Cancer Medicine* (9th ed.). Wiley Blackwell. ISBN 9781119000846.
11. https://www.cs.toronto.edu/~kriz/cifar.html
12. https://digitalslidearchive.github.io/HistomicsTK/