

# **PENTA CANCER DETECTION SYSTEM USING DEEP LEARNING**

## **A PROJECT REPORT**

*Submitted by*

**VIBHA TI                            2017103608**

**YUVASHREE N                        2017103615**

**SHREENIDHI S                        2017103620**

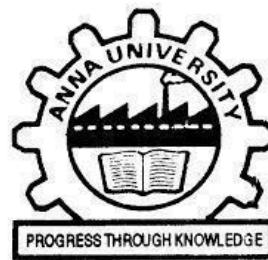
*in partial fulfillment of the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**COLLEGE OF ENGINEERING, GUINDY**

**ANNA UNIVERSITY : CHENNAI 600 025**

**MAY 2021**

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report "**PENTA CANCER DETECTION SYSTEM USING DEEP LEARNING**" is the bonafide work of "**VIBHA TI (2017103608), YUVASHREE (2017103615) and SHREENIDHI S (2017103620)**" who carried out the project work under my supervision, for the fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. Certified further that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on these or any other candidates.



**Place:** Chennai

**Dr.A.P.Shanthi**

**Date:**

Professor

Department of Computer Science and Engineering

Anna University, Chennai – 25

**COUNTERSIGNED**



**SIGNATURE**

**Dr. Valli S,**  
Head of the department,

Department of Computer Science and Engineering,  
Anna University Chennai – 60002

## **ABSTRACT**

Early detection of cancer is the top priority for saving the lives of many. Typically, visual examination and manual techniques are used for these types of cancer diagnosis. This manual interpretation of medical images demands high time consumption and is highly prone to mistakes. Thus, in this project, we apply deep learning algorithms to classify between the different types of cancer and detect its presence without the need for a number of consultations from different doctors. This leads to the earlier prediction of the presence of the disease and allows us to take prior actions immediately to avoid further consequences in an effective and cheap manner avoiding human error rate. In this project five different types of cancer such as lung cancer, brain tumor, cervical cancer, thyroid cancer, and skin cancer are determined. A web application will be developed as a hospital application where an input CT/MRI/histopathological image will be given to predict which type of cancer.

## திட்டப்பணிச் சுருக்கம்

புற்றுநோயை முன்கூட்டியே கண்டறிவது பலரின் உயிரைக் காப்பாற்றுவதற்கான முன்னுரிமை. பொதுவாக, இந்த வகை புற்றுநோய் கண்டறிதலுக்கு காட்சி பரிசோதனை மற்றும் கையேடு நுட்பங்கள் பயன்படுத்தப்படுகின்றன. மருத்துவப் படங்களின் இந்த கையேடு விளக்கம் அதிக நேர நுகர்வு கோருகிறது மற்றும் தவறுகளுக்கு அதிக வாய்ப்புள்ளது. எனவே, இந்த திட்டத்தில், பல்வேறு வகையான புற்றுநோய்களுக்கு இடையில் வகைப்படுத்தவும், பல்வேறு மருத்துவர்களிடமிருந்து பல ஆலோசனைகள் தேவையில்லாமல் அதன் இருப்பைக் கண்டறியவும் ஆழந்த கற்றல் வழிமுறைகளைப் பயன்படுத்துகிறோம். இது நோய் இருப்பதைப் பற்றிய முந்தைய கணிப்புக்கு வழிவகுக்கிறது மற்றும் மனித பிழை விகிதத்தைத் தவிர்ப்பதன் மூலம் மேலும் விளைவுகளை பயனுள்ள மற்றும் மலிவான முறையில் தவிர்க்க உடனடியாக முன் நடவடிக்கைகளை எடுக்க அனுமதிக்கிறது. இந்த திட்டத்தில் நுரையீரல் புற்றுநோய், மூளை கட்டி, கர்ப்பப்பை வாய்ப் புற்றுநோய், தைராய்டு புற்றுநோய் மற்றும் தோல் புற்றுநோய் போன்ற ஐந்து வகையான புற்றுநோய்கள் தீர்மானிக்கப்படுகின்றன. ஒரு வலை பயன்பாடு மருத்துவமனை பயன்பாடாக உருவாக்கப்படும், அங்கு எந்த வகையான புற்றுநோயைக் கணிக்க சி.டி./ எம்.ஆர்.ஐ. / ஹரிஸ்டோபோதாலஜிக்கல் படம் வழங்கப்படும்.

## **ACKNOWLEDGEMENT**

First and foremost, we would like to express our deep sense of gratitude to our guide **Dr A.P. SHANTHI**, Professor, Department of Computer Science and Engineering, Anna University Chennai, for her excellent guidance, counsel, continuous support and patience. She has helped us to come up with this topic and guided us in the development of this project.

We also would like to thank her for her kind support and for providing necessary facilities to carry out the work. We are thankful to the project committee members **Dr. MAHALAKSHMI G.S.**, Assistant professor, **Dr. GEETHA PALANISAMY**, Associate professor for their valuable guidance and support.

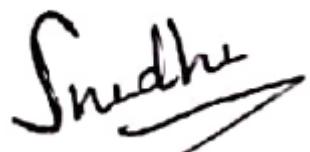
We also like to extend our gratitude to **Dr. VALLI S**, Head of the Department, Department of Computer Science and Engineering, Anna University Chennai, for supporting us with the technical resources required for our project.



**Vibha TI**



**Yuvashree N**



**Shreenidhi S**

## TABLE OF CONTENTS

<b>CHAPTER</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>BONAFIDE CERTIFICATE</b>	i
	<b>ABSTRACT ENGLISH</b>	ii
	<b>ABSTRACT TAMIL</b>	iii
	<b>ACKNOWLEDGEMENT</b>	vi
	<b>LIST OF FIGURES</b>	vii
	<b>LIST OF ABBREVIATION</b>	ix
	<b>LIST OF SYMBOLS</b>	x
<b>I</b>	<b>INTRODUCTION</b>	1
	1.1 GENERAL	1
	1.2 PROBLEM STATEMENT	3
	1.3 PROPOSED SYSTEM	4
<b>II</b>	<b>REVIEW OF LITERATURE</b>	5
	2.1 INTRODUCTION	5
	2.2 LITERATURE SURVEY	5
<b>III</b>	<b>SYSTEM MODEL</b>	10
	3.1 INTRODUCTION	10
	3.2 SYSTEM OVERVIEW	10
<b>IV</b>	<b>MODULE DESIGN</b>	11
	4.1 INTRODUCTION	11
	4.2 MODULE DESCRIPTION	11
	4.3 PREPROCESSING	11
	4.4 DATA AUGMENTATION	13

4.5	CANCER DETECTION USING VGG-16	14
4.5.1	VGG-16 Architecture	14
4.5.2	Model Generation	16
4.5.3	Optimization using SGD and ADAM	17
4.6	WEB DEVELOPMENT	22
4.6.1	Front end development using REACTJS	23
<b>V</b>	<b>IMPLEMENTATION DETAILS</b>	24
5.1	INTRODUCTION	24
5.2	SYSTEM REQUIREMENTS	24
5.3	DATASET	25
5.4	MODEL	25
5.5	IMPLEMENTATION SNAPSHOTS	27
5.6	TEST CASES	28
5.6.1	Analysis of Test Cases	30
5.7	EVALUATION RESULTS	32
5.7.1	Training	32
5.7.2	Loss and Accuracy Graph	33
<b>VI</b>	<b>CONCLUSION AND FUTURE WORK</b>	35
6.1	CONCLUSION	35
6.2	FUTURE WORK	35
	<b>REFERENCES</b>	36

## **LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>NAME OF THE FIGURE</b>	<b>PAGE NO.</b>
3.1	Architecture diagram	10
4.1	Preprocessing module	12
4.2	Data augmentation module	14
4.3	VGG architecture	15
4.4	Model generation	16
4.5	Nesterov weight update	18
4.6	Web development module	22
5.1	Model Summary	26
5.2	Image before augmentation	27
5.3	Augmented image	27
5.4	Home page	28
5.5	Cancer detection	28
5.6	Brain image	28
5.7	Breast image	29
5.8	Cervical image	29
5.9	Lung image	30

5.10	Thyroid image	30
5.11	Ignore class	30
5.12	Fine-tuned accuracy	32
5.13	Initial training	32
5.14	Final training	33
5.15	Accuracy graph	33
5.16	Loss Graph	34

## **LIST OF ABBREVIATIONS**

<b>ABBREVIATION</b>	<b>FULL FORM</b>
ADAM	Adaptive Moment Estimation
CNN	Convolutional Neural Network
CT	Computed Tomography
DOM	Document Object Model
FC	Fully Connected
GPU	Graphical Processing Unit
MRI	Magnetic Resonance Imaging
NAG	Nesterov Accelerated Gradient
ReLU	Rectified Linear Unit
RMSprop	Root Mean Square Propagation
SGD	Stochastic Gradient Descent
UI	User Interface
VGG	Visual Geometry Group

## LIST OF SYMBOLS

<b>NOTATION</b>	<b>MEANING</b>
$w_k$	Weight at kth iteration
$\eta$	Learning rate
$\gamma$	Weight decay factor
$d$	Dampening coefficient
$v_k$	Momentum vector for kth iteration
$\hat{\nabla}$	Stochastic Gradient after weight decay
$\nabla$	Stochastic Gradient calculated at $w_k$
$\mu$	Nesterov Momentum coefficient
$m_k$	First momentum vector for kth iteration
$v_k^a$	Second momentum vector for kth iteration
$\beta$	Exponential decay rates
$\hat{v}_k$	Bias corrected second moment vector for kth iteration
$\epsilon$	Negative power of 10 to prevent division by 0
$\eta_a$	New learning rate

# **CHAPTER I**

## **INTRODUCTION**

### **1.1 GENERAL**

Cancer is a cluster of diseases involving abnormal cell growth with the potential to invade or unfold to different components of the body. These distinctions with benign tumors, don't unfold. Attainable signs and symptoms embrace a lump, abnormal harm, prolonged cough, unexplained weight loss, and an amendment in internal organ movements. These symptoms could indicate cancer, they'll even have different causes. Over one hundred varieties of cancers have an effect on humans.

Breast cancer is one type of cancer that may develop from the breast tissue. Signs of carcinoma might embody a lump within the breast, a amendment in breast form, dimpling of the skin, fluid coming back from the sex organ, a newly-inverted sex organ, or a red or scaly patch of skin. In those with a foreign unfold of the illness, there is also bone pain, swollen humour nodes, shortness of breath, or yellow skin.

Lung cancer, additionally referred to as respiratory organ cancer, may be a malignant respiratory organ growth characterised by uncontrolled cell growth in tissues of the respiratory organ. This growth will unfold on the far side of the respiratory organ by the method of metastasis into near tissue or alternative components of the body. Most cancers that begin within the respiratory organ, referred to as primary respiratory organ cancers, are carcinomas. The 2 main varieties are small-cell respiratory organ cancer (SCLC) and non-small-cell respiratory organ cancer

(NSCLC). The foremost common symptoms are coughing (including expulsion of blood), weight loss, shortness of breath, and chest pains.

A brain tumour happens once abnormal cells are kind among the brain. There are 2 main forms of tumors: cancerous (malignant) tumors and benign (non-cancerous) tumors. Cancerous tumors are often divided into primary tumors that begin among the brain, and secondary tumors that have developed from elsewhere, referred to as brain metastasis tumors. Every type of brain tumor might manufacture symptoms that change reckoning on a part of the brain concerned. These symptoms might embody headaches, seizures, issues with vision, vomiting, and mental changes. The headache is classically worse within the morning and goes away with regurgitation. alternative symptoms might embody problem walking, speaking, or with sensations. because the illness progresses, cognitive states might occur.

Thyroid cancer is cancer that develops from the tissues of the ductless gland. It's an illness during which cells grow abnormally and have the potential to unfold to alternative components of the body. Symptoms will embody swelling or a lump within the neck. Cancer may occur within the thyroid once unfold from alternative locations, during which case it's not classified as thyroid cancer.

Thyroid cancer typically presents while not symptoms and simply causes a thyromegaly or nodules that typically won't impact the thyroid's operate or cause any pain within the early stages. because it progresses and cancerous nodules grow, you'll expertise pain within the neck,

problem swallowing, or a cacophonous voice. The good news is that thyroid cancer is sometimes terribly treatable, and therefore the survival rates are high.

Cervical cancer may be a malignancy of the bottommost part of the womb which will be prevented by smear screening. Symptoms embody injury in between periods, chronic low back pain or lower abdominal pain may additionally occur. In some rare cases, there are also no symptoms the least bit.

This project attempts to classify the five different types of cancer, based on the input images which may be in the form of CT, MRI scans or histopathological images. The rest of the thesis is divided into seven chapters. Chapter I discusses the objective and the work proposed. Chapter II discusses a survey of the related works. Chapter III discusses the overall system design. While chapter IV discusses detailed module design. Chapter V lists down the implementation details and evaluation results. Chapter VI discusses the future work. The list of references are provided in Chapter VII.

## **1.2 PROBLEM STATEMENT**

Cancer is considered to be one of the root causes of a large number of deaths worldwide. Researchers and Doctors fear this issue and are also facing the challenges of fighting cancer. So we propose a system that closely matches the following objectives:

1. To aid in the detection of cancer at an early stage.
2. To use CT images or x-ray images for the detection of cancer.
3. To minimize the time taken to detect the presence of cancer.

4. To determine a cheaper way to predict the cancer presence.
5. To predict five different types of cancer.

### **1.3 PROPOSED SYSTEM**

In this project, we apply deep learning algorithms to classify the different types of cancer. We also apply the VGG16 architecture to predict the highest accuracy, which is an intelligent cancer recognition system that can automatically recognize the presence of cancer cells, combining the techniques of image processing and deep learning for determining five different types of cancer. Here, five different types of cancer such as lung cancer, skin cancer, cervical cancer, thyroid cancer and brain tumour are determined. Optimizing techniques like SGD - stochastic gradient descent, ADAM and Activation methods like ReLU help to increase the accuracy. A web application will be developed where CT, MRI scans and histopathological images of organs will be given as input and the model will predict if it's cancerous or not.

## CHAPTER II

### REVIEW OF LITERATURE

#### **2.1 INTRODUCTION**

The following section discusses the works related to cancer detection using deep learning. The most popular of the existing techniques has been discussed as below

#### **2.2 LITERATURE SURVEY**

Liu et al [5] have used feature dimensionality reduction techniques in radiomic studies with a large number of features. The conventional radiomic approach suffered from noise, and feature dimensionality reduction techniques, which were not equipped to utilize latent supervision information of patient data under study proposed by [5]. The study says that, in order to achieve robustness to noise and feature dimensionality reduction with improved discriminative power, a robust collaborative clustering method had to be developed, so that patients and radiomic features can be simultaneously clustered into distinct groups respectively, under adaptive sparse regularization. Such a method is built upon matrix tri-factorization enhanced by adaptive sparsity regularization for simultaneous feature dimensionality reduction and denoising. Particularly, latent grouping information of patients with distinct radiomic features is learned and utilized as supervision information to guide the feature dimensionality reduction. The noise in radiomic features is adaptively isolated in a Bayesian framework under a general assumption of Laplacian distributions of transform-domain coefficients. Experiments on synthetic data have demonstrated the effectiveness of the proposed approach in data clustering, and evaluation

results on an FDG-PET/CT dataset of rectal cancer patients have demonstrated that the proposed method outperforms alternative methods in terms of both patient stratification and prediction of patient clinical outcomes.

Jose et al [1]’s comparison of phantom and patient data between a conventional clinical PET-CT scanner and a dedicated high resolution brain scanner, the HRRT, showed that there is benefit of high resolution scanners to reduce the partial volume effect and spill-in from surrounding tissues for small structures and lesions. This comes with the consequence of higher voxel noise. RM image reconstruction to restore PET-CT is also provided, which proves to be beneficial but is unable to achieve the resolution performance of the HRRT, and has the potential for introducing image artifacts.

An exhaustive empirical study was developed by Mandal et al [6] to identify biomarkers using two approaches: frequency-based and network-based. This is carried on by implementing over seventeen different bi-clustering algorithms and six different cancer expression datasets. To systematically analyze the bi-clustering algorithms, a novel method of enrichment analysis is performed, followed by subtype identification and biomarker identification. Bi-clustering algorithms such as C&C, SAMBA and Plaid have also proved to be useful in detecting biomarkers by both approaches for all datasets, except prostate cancer. Thus, a total of 102 gene biomarkers using frequency-based methods are detected. Out of these, 19 are for blood cancer, 36 for lung cancer, 25 for colon cancer, 13 for multi-tissue cancer and 9 for prostate cancer. Using the network-based approach a total of 41 gene biomarkers are detected, out of which 15

from blood cancer, 12 from lung cancer, 6 from colon cancer, 7 from multi-tissue cancer and 1 from prostate cancer dataset are obtained.

Temporal Enhanced Ultrasound (TeUS), comprising the analysis of variations in backscattered signals from a tissue over a sequence of ultrasound frames, has been previously proposed as a new paradigm for tissue characterization. Azizi et al [2] proposed a system that aims at employing deep Recurrent Neural Networks (RNN) to explicitly model the temporal information in TeUS. By investigating several RNN models, the concept of Long Short-Term Memory (LSTM) networks are demonstrated, in order to achieve the highest accuracy in separating cancer from benign tissues in the prostate. Tavassolian et al[7] have presented algorithms for in-depth analysis of LSTM networks and achieved clear statistics of the area under the curve, sensitivity, specificity, and accuracies of 0.96, 0.76, 0.98 and 0.93, respectively. Their work[7] also suggested that temporal modeling of TeUS using RNN can significantly improve cancer detection accuracy over previously presented works.

Wang et al [10] have presented a new breast cancer detection method that combines thermography and high frequency excitation techniques. Their proposed method uses distribution and variation of the temperature on the breast surface in order to estimate the location and size of a malignant tissue (a cancerous tumor). Their research work[10] stems from the analysis of the radio frequency (RF) radiation effects on the human body. In their work, a description of RF effects on the human body and simulation results is provided. In order to validate their proposed method, a multilayer 3D breast model is simulated. First, breast tissue is excited with different types of RF antennas following which an electromagnetic analysis is

conducted. Next, the heat transfer equation is applied for thermal analysis to estimate the distribution of surface temperature.

Setio et al [8] employed a novel Computer-Aided Detection (CAD) system for pulmonary nodules using multi-view convolutional networks (ConvNets), for which discriminative features are automatically learnt from the training data. The network is fed with nodule candidates obtained by combining three candidate detectors specifically designed for solid, sub solid, and large nodules. For each candidate, a set of 2-D patches from differently oriented planes is extracted. The proposed architecture[8] comprises multiple streams of 2-D ConvNets, for which the outputs are combined using a dedicated fusion method to get the final classification. Data augmentation and dropout are applied to avoid overfitting. On 888 scans of the publicly available LIDC IDRI dataset, the proposed method[8] reaches high detection sensitivities of 85.4% and 90.1% at 1 and 4 false positives per scan, respectively. An additional evaluation on independent datasets from the ANODE09 challenge and DLCST was performed. According to Zheng et al [4], the study aimed at developing a new quantitative image feature analysis scheme and investigating its role along with 2 genomic biomarkers namely, protein expression of the excision repair cross-complementing 1 (ERCC1) genes and a regulatory subunit of ribonucleotide reductase (RRM1), in predicting cancer recurrence risk of Stage I non-small-cell lung cancer (NSCLC) patients after surgery. By using chest computed tomography images, a computer-aided detection scheme is developed to segment lung tumors and compute tumor related image features. After feature selection, a Naïve Bayesian network based classifier using 8 image features and a Multilayer Perceptron classifier using 2 genomic biomarkers to predict

cancer recurrence risk, respectively are trained. Fusion of prediction scores generated by the two classifiers further improved prediction performance.

# CHAPTER III

## SYSTEM MODEL

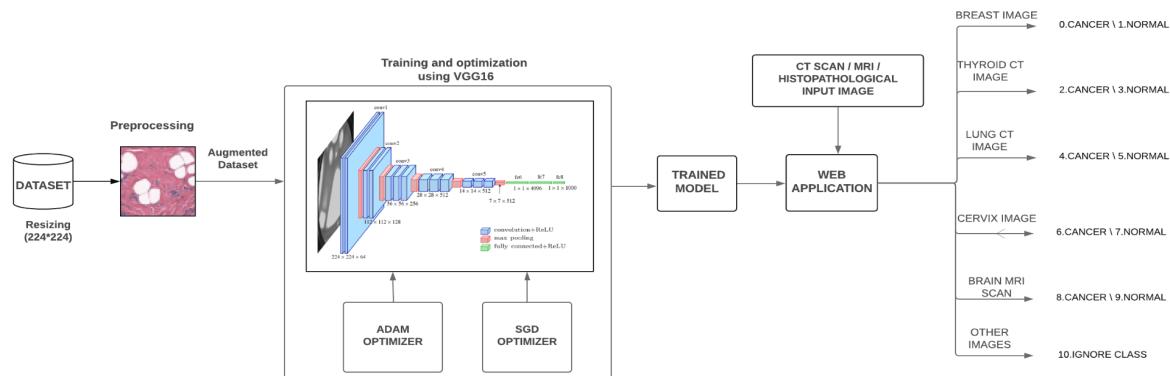
### 3.1 INTRODUCTION

This chapter explains the overall architecture and the flow of the system.

### 3.2 SYSTEM OVERVIEW

The project aims at determining the presence and type of cancer with CT scans/MRI scans/Histopathological images as inputs. Once the dataset images are gathered, they are split into training and testing set. The samples split for training are then preprocessed and augmented.

A sequential VGG16 model is built and trained. It then undergoes optimization and loss minimization. After validation and testing the model is serialized where the model's updated weights will be saved. A web application using a javascript framework reactJS is developed where an input scan or histopathological image is given to predict the type of cancer. This method provides an effective method to determine the presence and type of cancer.



**Fig 3.1- Architecture diagram**

## **CHAPTER IV**

## **MODULE DESIGN**

### **4.1 INTRODUCTION**

This chapter introduces the various modules used in the system. Sections 4.3 to 4.6 explains the algorithm, detailed diagram and processes involved in the four modules namely, preprocessing, data augmentation, cancer prediction using the VGG16 architecture and web application development modules.

### **4.2 MODULE DESCRIPTION**

The entire system is split into four modules which include:

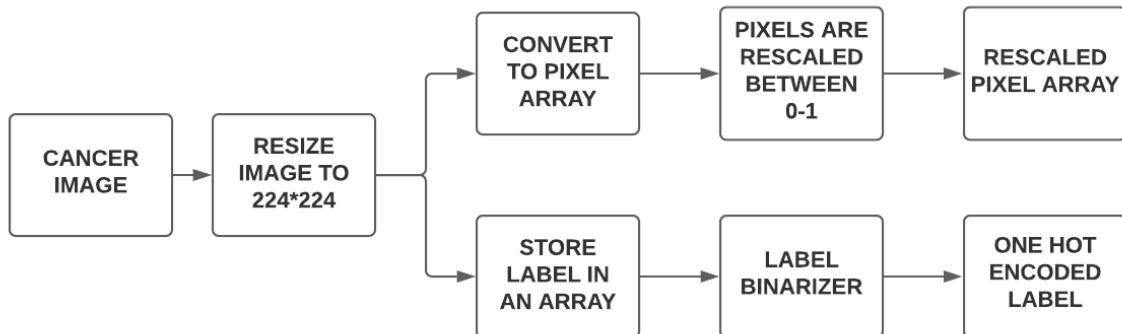
- Preprocessing
- Data Augmentation
- Cancer Prediction Using The VGG16 Architecture and
- Web Application Development

### **4.3 PREPROCESSING**

Image Preprocessing is typically only applied to the training dataset, and not to the validation or test dataset. It basically involves image resizing and pixel scaling. They must be performed consistently across all datasets that interact with the model. The intent is to apply the basic process of resizing and rescaling on the input cancer image or scan. Initially, the input image is resized to a dimension of 244\*244 and this is applied on the expanded dataset. This means that the image is subjected to a process of pixel transformation, which involves the process of

conversion of the image/scan to a pixel array. The two-dimensional grid of pixels would define the form and color of a digital input image. The pixels in the image undergo rescaling and are scaled in between a range of 0 to 1.

Another alternative process of storing the labels in an array format is performed where a label array is simply a collection of labels for the items of the array. Finally, the output is fed to a label binarizer and it undergoes the process of label binarization, where the sample labels are converted to binary. The label binarizer assigns a unique value or number to each label in a categorical Feature.



**Fig 4.1-Preprocessing flow**

---

### Algorithm 1

---

**Input:** image of various sizes

**Output:** array of resized image pixels and binarized labels

**1:** Resize input image to 224\*224 dimension

**2:**  $\text{data}[] \leftarrow \text{pixel}[]$  of input images

**3:** pixels of the each image is rescaled between 0-1 by dividing by 255

**4:** `label[]`← labels of images are stored in label array

**5:** use Label binarizer to convert the labels to binaries

---

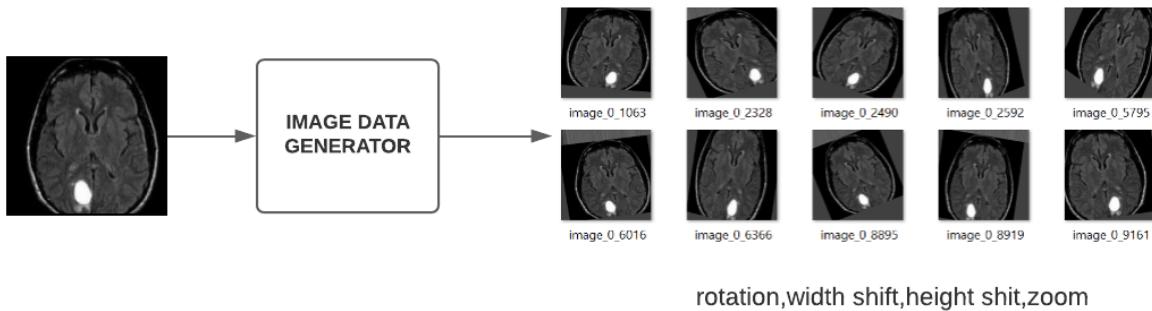
## 4.4 DATA AUGMENTATION

The performance of deep learning neural networks often improves with the amount of data available. Data augmentation is a technique to artificially create new training data from existing training data. This is done by applying domain-specific techniques to examples from the training data that create new and different training examples.

Image data augmentation is one of the most well-known types of data augmentation and involves creating transformed versions of images in the training dataset that belong to the same class as the original image.

Transformations include a range of operations from the field of image manipulation, such as shifts, flips, zooms, and much more. The intent is to expand the training dataset with new, plausible examples. This means, variations of the training set images are likely to be seen by the model.

Image data augmentation is typically only applied to the training dataset, and not to the validation or test dataset. This is different from data preparation such as image resizing and pixel scaling; they must be performed consistently across all datasets that interact with the model.



**Fig 4.2-Data augmentation**

## Algorithm 2

**Input:** training images

**Output:** augmented images

1: `aug←Imagedatagenerator(rotation_range, width_shift_range, height_shift_range, shear_range, zoom_range, horizontal_flip, flip_mode)`

- `Rotation_range←30°`
- `Width_shift_range←0.1(10%)`
- `Height_shift_range←0.1(10%)`
- `Shear_range←0.2 (20%)`
- `Zoom_range←0.2(80% - 120%)`
- `Horizontal_flip←true`
- `Flip_mode←nearest`

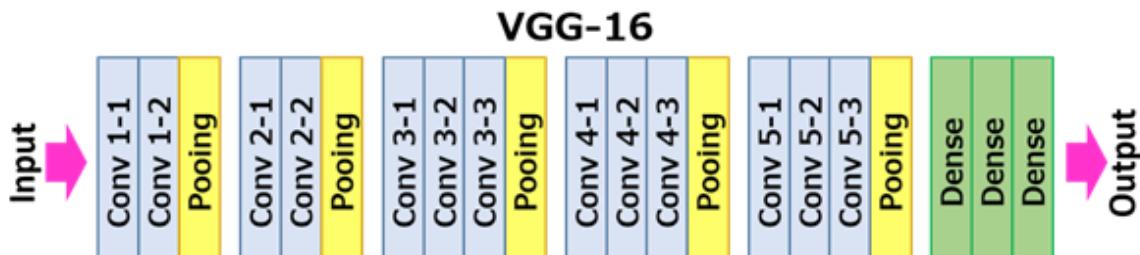
## 4.5 CANCER DETECTION USING VGG-16

### 4.5.1 VGG-16 ARCHITECTURE

VGG16 is considered to be one of the excellent vision model architectures till date. The key innovation in this architecture was the definition and repetition of what we will refer to as

VGG-blocks. These are groups of convolutional layers that use small filters (e.g.  $3 \times 3$  pixels) followed by a max pooling layer. A convolutional neural network with VGG-blocks is a sensible starting point when developing a new model from scratch as it is easy to understand, easy to implement, and very effective at extracting features from images.

Most unique thing about VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of  $3 \times 3$  filter with a stride 1 and always used the same padding and maxpool layer of  $2 \times 2$  filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it as 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters.



**Fig 4.3-VGG architecture**

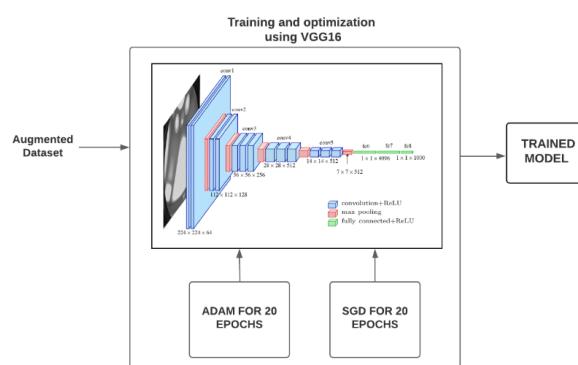
The input to the cov1 layer is of fixed size  $224 \times 224$  RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field:  $3 \times 3$  (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes  $1 \times 1$  convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed

to 1 pixel; The spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for  $3 \times 3$  conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a  $2 \times 2$  pixel window, with stride 2.

Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs actual classification and it contains as many channels as the number of classes. The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

#### 4.5.2 MODEL GENERATION

A sequential VGG16 model is built by adding 13 convolutional layers and 3 fully connected dense layers at the end. 75% of the dataset is taken for training and 25% for validation and testing. The training set is passed through the model and trained for a total of 60 epochs. The first 20 epochs is optimized with ADAM and the last 20 epochs(fine-tuning) is optimized with SGD. The model is serialized and saved. Finally, the model is validated and tested to compute the final accuracy of the model.



**Fig 4.4-Model generation**

### 4.5.3 OPTIMIZATION USING SGD AND ADAM OPTIMIZERS

#### Stochastic Gradient Descent

Stochastic gradient descent is an iterative method for optimizing an objective function with suitable smoothness properties . It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). Especially in high-dimensional optimization problems this reduces the computational burden, achieving faster iterations in trade for a lower convergence rate.

SGD with momentum- SGD has trouble navigating ravines, i.e. areas where the surface curves much more steeply in one dimension than in another , which are common around local optima. In these scenarios, SGD oscillates across the slopes of the ravine while only making hesitant progress along the bottom towards the local optimum.

Momentum is a method that helps accelerate SGD in the relevant direction and dampens oscillations. It does this by adding a fraction  $\gamma$  of the update vector of the past time step to the current update vector:

$$v_t = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta)$$

$$\theta = \theta - v_t$$

The momentum term  $\gamma$  is usually set to 0.9 or a similar value.The momentum term increases for dimensions whose gradients point in the same directions and reduces updates for dimensions

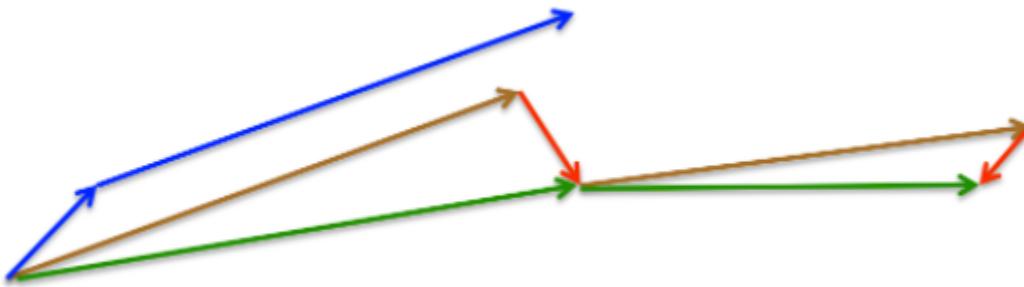
whose gradients change directions. As a result, we gain faster convergence and reduced oscillation.

Nesterov accelerated gradient:Nesterov accelerated gradient (NAG) is a way to give our momentum term a kind of prescience of where to move next in the slope by approximating the next position of the parameters instead of blindly moving down the slope.To effectively calculate the next position of the parameters look ahead the gradient is calculated not w.r.t. to our current parameters  $\theta$  but w.r.t. the approximate future position of the parameters.

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta - \gamma v_{t-1})$$

$$\theta = \theta - v_t$$

Again, we set the momentum term  $\gamma$  to a value of around 0.9.



**Fig 4.5-Nesterov weight update**

While Momentum first computes the current gradient (small blue vector in Fig4.5) and then takes a big jump in the direction of the updated accumulated gradient (big blue vector), NAG first makes a big jump in the direction of the previous accumulated gradient (brown vector), measures the gradient and then makes a correction (red vector), which results in the complete NAG update (green vector). This anticipatory update prevents us from going too fast and results in increased responsiveness.

---

### Algorithm 3 Stochastic Gradient Descent(SGD)

---

**Input:** the weights  $w_k$ , learning rate  $\eta$ , weight decay  $\gamma$ , dampening  $d$ , boolean *nesterov*

**Output:**updated weights of parameters

```
1:  $v_0 = 0$ 
2: function  $\Delta_{SGD}(w_k, \nabla, \gamma, \mu, d, nesterov)$ 
3:    $\hat{\nabla} = \nabla + w_k \cdot \gamma$ 
4:   if  $m \neq 0$  then
5:     if  $k = 0$  then
6:        $v_k = \hat{\nabla}$ 
7:     else
8:        $v_k = v_k - 1 \cdot \mu + \nabla \cdot (1 - d)$ 
9:     end if
10:    if  $nesterov = True$  then
11:       $v_k = \hat{\nabla} + v_k \cdot \mu$ 
12:    end if
13:  end if
14:  return  $v_k$ 
15: end function
16: for batches do
17:    $w_k + 1 = w_k - \eta_s \cdot \Delta_{SGD}(w_k, \nabla, \mu, d, nesterov)$ 
18: end for
```

---

### Adaptive Moment Estimation

Adaptive Moment Estimation (Adam) is another method that computes adaptive learning rates for each parameter. In addition to storing an exponentially decaying average of past squared gradients  $v_t$  like Adadelta and RMSprop, Adam also keeps an exponentially decaying average of past gradients  $m_t$ , similar to momentum. Whereas momentum can be seen as a ball running down a slope, Adam behaves like a heavy ball with friction, which thus prefers flat minima in the error surface. We compute the decaying averages of past and past squared gradients  $m_t$  and  $v_t$  respectively as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$m_t$  and  $v_t$  are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients respectively, hence the name of the method. As  $m_t$  and  $v_t$  are initialized as vectors of 0's, the authors of Adam observe that they are biased towards zero, especially during the initial time steps, and especially when the decay rates are small (i.e.  $\beta_1$  and  $\beta_2$  are close to 1). They counteract these biases by computing bias-corrected first and second moment estimates:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

They then use these to update the parameters just as we have seen in Adadelta and RMSprop, which yields the Adam update rule:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t + \epsilon}} \widehat{m}_t$$

The default values of 0.9 for  $\beta_1$ , 0.999 for  $\beta_2$ , and  $10^{-8}$  for  $\epsilon$ .

---

#### Algorithm 4 Adaptive Moment Estimation(ADAM)

---

**Input:** the weights  $w_k$ , learning rate  $\eta$ , weight decay  $\gamma$ ,  $\beta_1$ ,  $\beta_2$ ,  $\epsilon$ , boolean *amsgrad*

**Output:** updated weights of parameters

```

1:  $m_0 = 0$ 
2:  $v_0^a = 0$ 
3:  $\widehat{v}_0 = 0$ 
4: function  $\Delta_{ADAM}(w_k, \nabla, \eta, \gamma, \beta_1, \beta_2, amsgrad)$ 
5:    $\widehat{\nabla} = \nabla + w_k \cdot \gamma$ 
6:    $m_k = m_k - 1 \cdot \beta_1 + \widehat{\nabla} (1 - \beta_1)$ 
7:    $v_k^a = v_k^a - 1 \cdot \beta_2 + \widehat{\nabla} \cdot \widehat{\nabla} (1 - \beta_2)$ 
8:   if amsgrad = True then
9:      $\widehat{v}_k = \max(\widehat{v}_k - 1, v_k^a)$ 
10:     $denom = \frac{\sqrt{v_k}}{\sqrt{1 - \beta_2} + \epsilon}$ 
11:   else
12:      $denom = \frac{\sqrt{v_k^a}}{\sqrt{1 - \beta_2} + \epsilon}$ 
13:   end if
14:    $\eta_a = \frac{\eta}{1 - \beta_1}$ 

```

```

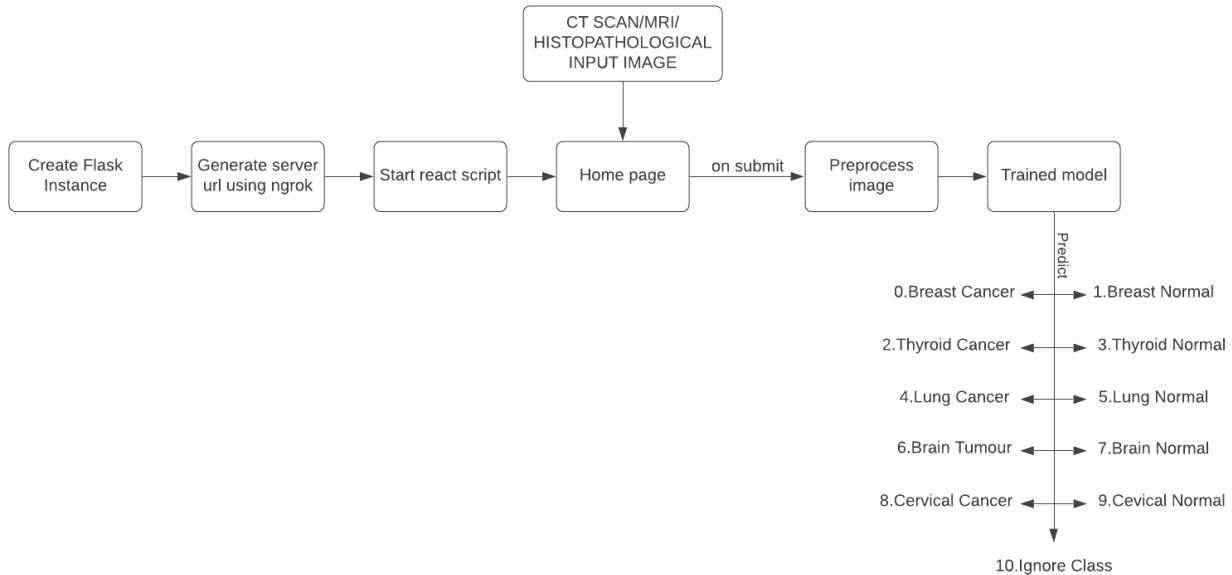
15:    $d_k = \frac{m_k}{denom}$ 
16:   return  $d_k, \eta_a$ 
17: end function
18: for batches do
19:    $d_k, \eta_a = \Delta_{ADAM}(w_k, \nabla, \eta, \gamma, \beta_1, \beta_2, \epsilon, amsgrad)$ 
20:    $w_k + 1 = w_k - \eta_a \cdot d_k$ 
21: end for

```

---

## 4.6 WEB DEVELOPMENT

A database is managed for the prediction process in the backend and the result is fetched through the web application. The Frontend is developed using ReactJS. Flask is used to integrate the trained model with the application. The fetched input is passed through the model and analyzed. The model predicts the input whether it is cancer or not.



**Fig 4.6-Web development module**

#### **4.6.1 FRONT END DEVELOPMENT USING REACT JS**

ReactJS is a JavaScript library used for building reusable UI components. React is also defined as a library for building composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. React abstracts away the DOM from you, offering a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native. React implements one-way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.

# CHAPTER V

## IMPLEMENTATION DETAILS

### 5.1 INTRODUCTION

This chapter discusses the implementation details, system requirements, step by step implementation output snapshots and results obtained. Snapshots of test cases and detailed analysis is done in this chapter.

### 5.2 SYSTEM REQUIREMENTS

The study is intended to classify cancer and non-cancer images in the medical field. The implementation is done in Python.

Programming Language	Python – 3.7.10
Google colab	-
Front-end	Visual Studio Code
API used	Keras – 2.4.1

**Table – 5.1 Software Requirements**

RAM	12GB
PROCESSOR	Core i5-2.11 GHz
GRAPHICS CARD	Colab GPU

**Table – 5.2 Hardware Requirements**

## 5.3 DATASET

For breast cancer (inductive duct carcinoma), histopathological images have been taken from kaggle having over 700 images of each class. For cervical cancer, histopathological images have been taken from the HERLEV dataset containing 917 images. HERLEV dataset contains histopathological images of 7 classes(carcinoma\_in\_situ, light\_dysplastic, moderate\_dysplastic, normal\_columnar, normal\_intermediate, normal\_superficial, severe\_dysplastic) of normal and abnormal cervical cells. For Lung cancer, CT images have been taken from Kaggle containing 1000 images. The dataset contains CT images of 4 classes (adenocarcinoma, large cell carcinoma, normal, squamous cell carcinoma) for both testing and training. For brain tumour, MRI images have also been taken from the Kaggle containing for about 250 images for both normal and cancer. CT scan images of thyroid cancer have been taken from *cancerimagingarchive.net* containing about 100 images for each class(cancer & normal). A total of about 3500 images are fed into the network.

## 5.4 MODEL

VGG algorithm is used to train the model . The model is fine tuned using optimizers like SGD, ADAM. The model was trained for 20 epochs initially using ADAM optimizer and later again for 20 more epochs to improve the accuracy of classification using SGD optimizer as shown in Fig-5.13 and Fig-5.14 respectively. Dropout layer is added to regularize the model in order to reduce overfitting. The final accuracy achieved is 94% as shown in Fig-5.12. The loss values are plotted against the every epoch as shown in Fig-5.15 and Fig-5.16. The loss tends to decrease in every epoch gradually, and a loss of 0.1 is recorded at the 40<sup>th</sup> epoch.

Model: "sequential\_1"

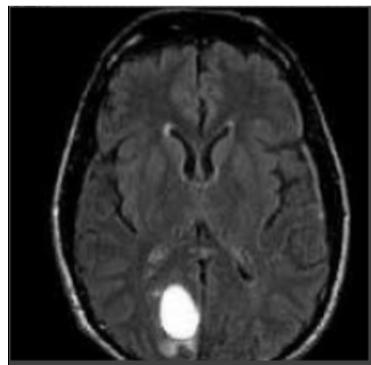
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 64)	1792
conv2d_1 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 112, 112, 64)	0
dropout (Dropout)	(None, 112, 112, 64)	0
conv2d_2 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_3 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 128)	0
dropout_1 (Dropout)	(None, 56, 56, 128)	0
conv2d_4 (Conv2D)	(None, 56, 56, 256)	295168
conv2d_5 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_6 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 256)	0
dropout_2 (Dropout)	(None, 28, 28, 256)	0
conv2d_7 (Conv2D)	(None, 28, 28, 512)	1180160
conv2d_8 (Conv2D)	(None, 28, 28, 512)	2359808
conv2d_9 (Conv2D)	(None, 28, 28, 512)	2359808
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 512)	0
dropout_3 (Dropout)	(None, 14, 14, 512)	0
conv2d_10 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_11 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_12 (Conv2D)	(None, 14, 14, 512)	2359808
max_pooling2d_4 (MaxPooling2D)	(None, 7, 7, 512)	0
dropout_4 (Dropout)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 4096)	102764544
dense_1 (Dense)	(None, 4096)	16781312
dropout_5 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 11)	45067
activation (Activation)	(None, 11)	0

Total params: 134,305,611  
Trainable params: 134,305,611  
Non-trainable params: 0

**Fig 5.1-Model summary**

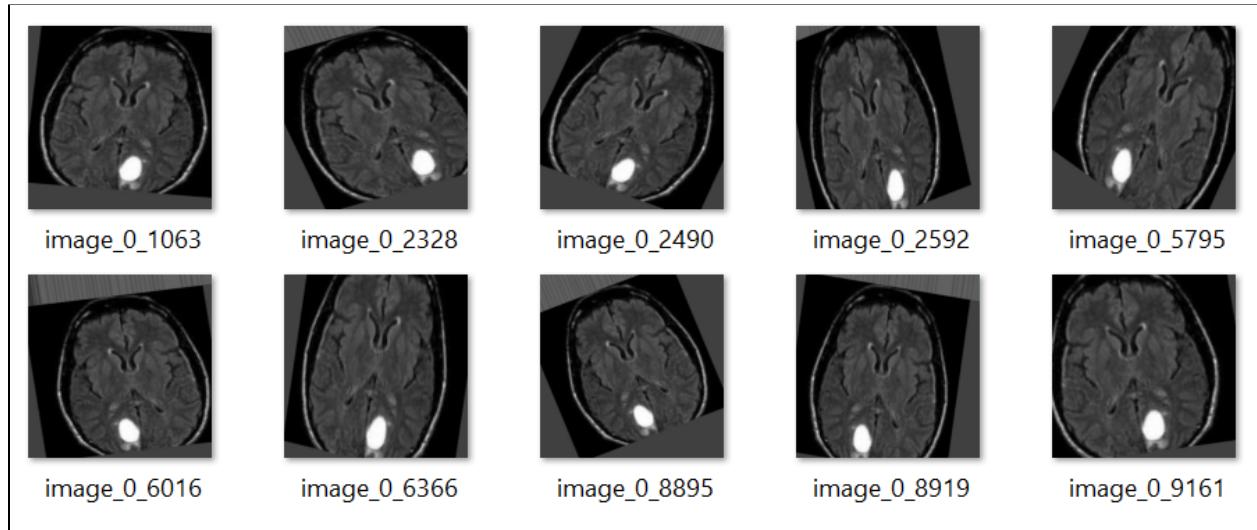
## 5.5 IMPLEMENTATION SNAPSHOT

PREPROCESSING: The dataset images are preprocessed to form an equal aspect ratio of 224\*224 so that it can be made ready for training with the model.(As shown in the below image)



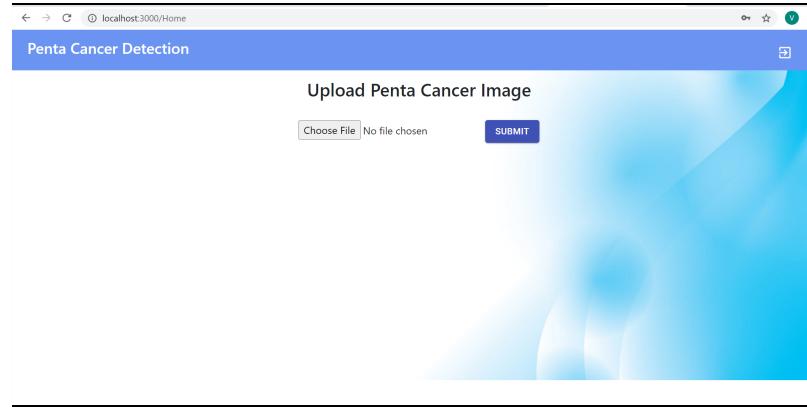
**Fig 5.2-Image before augmentation**

AUGMENTATION: Preprocessed images are rotated to 30 degrees, width shift range is set to 10%, height shift range is set to 10%, zoom range is 80-120%, shear range is set to 20% and the source image is also horizontally flipped.



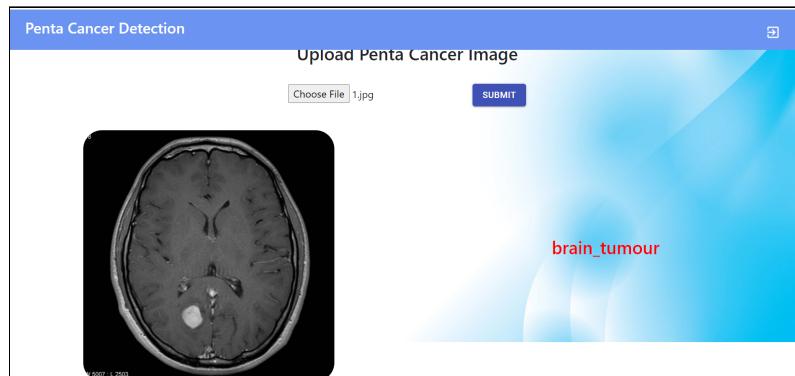
**Fig 5.3-Augmented image**

WEBSITE HOMEPAGE: The user can upload images to check for cancer. The below image portrays the user interface.



**Fig 5.4-Home page**

Once the image has been loaded from the device, it is processed by the generated model and the response showing the result is sent back to the UI. The below image shows that the uploaded image is of a cancerous brain tumour.

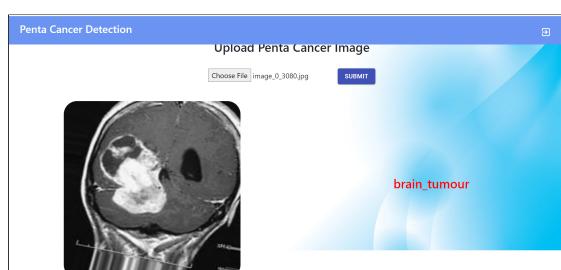


**Fig 5.5-Cancer detection**

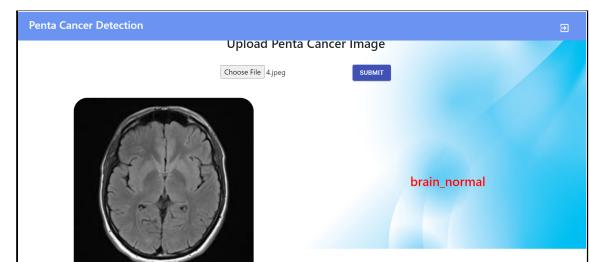
## 5.6 TEST CASES

Cancer and non-cancer images were fed into the model for every cancer, bringing out all corner cases and were detected as follows-

### BRAIN



**Fig 5.6(a)-Brain tumour**



**Fig 5.6(b)-Brain normal**

## BREAST

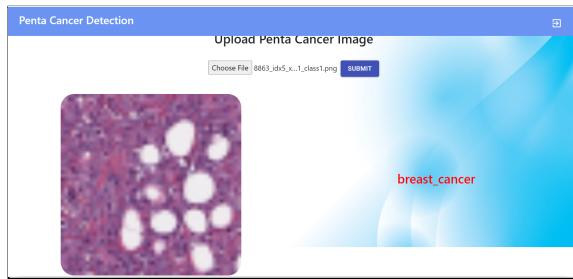


Fig 5.7(b)-Breast normal

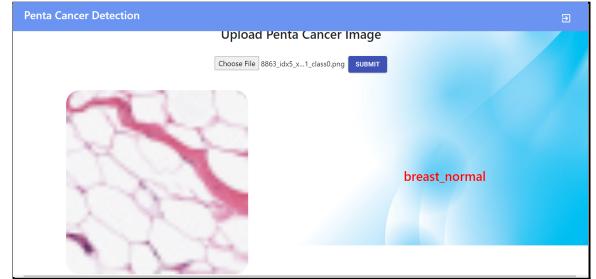


Fig 5.7(a)-Breast cancer

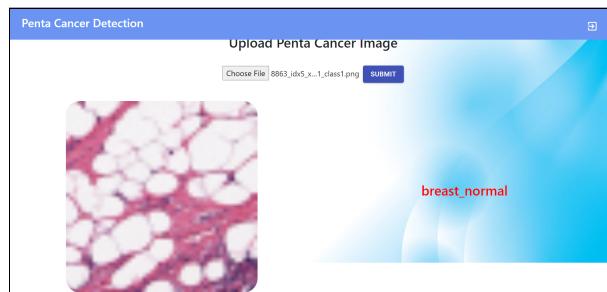


Fig 5.7(c)-Breast cancer

## CERVIX



Fig 5.8(a)-Cervical cancer

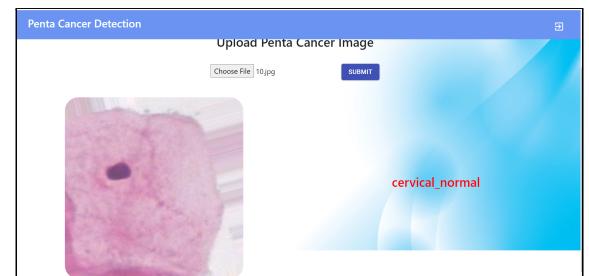


Fig 5.8(b)-Cervical cancer



Fig 5.8(c)-Wrong classification

## LUNG



Fig 5.9(a)-Lung cancer



Fig 5.9(b)-Lung normal

## THYROID



Fig 5.10(a)-Thyroid cancer

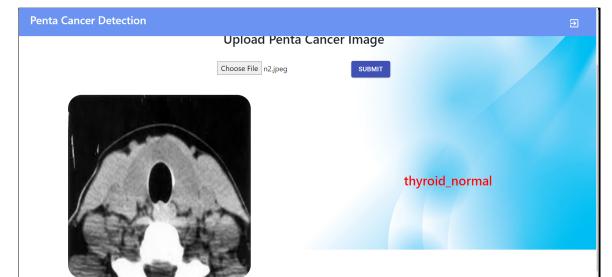


Fig 5.10(b)-Thyroid normal

## IGNORE CLASS



Fig 5.11-Ignore class

### 5.6.1 Analysis of Test Cases

- Fig 5.6(a)-the model has correctly identified the input brain tumour MRI as brain\_tumour.(True Positive).
- Fig 5.6(b)-the model has correctly identified the non\_cancer brain MRI as brain\_normal.(True Negative)

- Fig 5.7(a)-the model has correctly identified the cancerous breast histopathological image as `breast_cancer`.(True Positive).
- Fig 5.7(b)-the model has correctly identified the normal breast histopathological image as `breast_normal`.(True Negative).
- Fig 5.7(c)-the model has incorrectly identified the cancerous breast histopathological image as `breast_normal`.(False Negative).
- Fig 5.8(a)-the model has correctly identified the cancerous cervical histopathological image as `cervical_cancer`.(True Positive).
- Fig 5.8(b)-the model has correctly identified the normal cervical histopathological image as `cervical_normal`.(True Negative)
- Fig 5.8(c)-the model has wrongly identified the normal cervical histopathological image as cancerous cervical cell.(False Positive)
- Fig 5.9(a)-the model has correctly identified the cancerous lung CT image as `lung_cancer`.(True Positive)
- Fig 5.9(b)-the model has correctly identified the normal lung CT image as `lung_normal`.(True Negative)
- Fig 5.10(a)-the model has correctly identified the cancerous thyroid CT image as `thyroid_cancer`.(True Positive)
- Fig 5.10(b)- the model has correctly identified the normal thyroid CT image as `thyroid_normal`.(True Negative)
- Fig 5.11-the model has correctly identified a image that doesn't belong to any of the 5 class as `ignore_class`.(True Positive)

## 5.7 EVALUATION RESULTS

The model is initially trained for 20epochs(optimized with ADAM) and fine tuned for another 20epochs(optimized with SGD).

		Actual										
		0	1	2	3	4	5	6	7	8	9	10
Predicted	0	29	0	0	0	0	0	0	0	0	0	0
	1	2	45	0	0	0	0	0	0	0	0	0
	2	0	0	65	13	0	0	0	0	0	0	0
	3	0	0	30	243	0	0	0	0	0	0	0
	4	0	0	0	0	119	0	0	0	0	0	0
	5	0	0	0	0	5	99	0	0	0	0	0
	6	0	0	0	2	0	0	119	0	2	0	0
	7	0	0	0	0	0	0	0	129	3	0	0
	8	0	0	0	0	0	0	0	0	109	0	0
	9	0	0	0	0	0	0	0	0	0	26	2
	10	0	0	0	0	0	0	0	0	0	0	27

Confusion matrix				
		precision	recall	f1-score
Predicted	Actual			support
brain_normal	brain_normal	0.94	1.00	0.97
brain_tumour	brain_tumour	1.00	0.96	0.98
breast_cancer	breast_cancer	0.68	0.83	0.75
breast_normal	breast_normal	0.94	0.89	0.92
cervical_cancer	cervical_cancer	0.96	1.00	0.98
cervical_normal	cervical_normal	1.00	0.95	0.98
ignore	ignore	1.00	0.97	0.98
lung_cancer	lung_cancer	1.00	0.98	0.99
lung_normal	lung_normal	0.96	1.00	0.98
thyroid_cancer	thyroid_cancer	1.00	0.93	0.96
thyroid_normal	thyroid_normal	0.93	1.00	0.96

accuracy	0.94	1069
macro avg	0.95	0.96
weighted avg	0.95	0.94

Fig 5.12-Classification report

### 5.7.1 TRAINING

#### Initial 20epochs optimized with ADAM

```

Epoch 1/20
61/61 [=====] - 105s 2s/step - loss: 1.6095 - accuracy: 0.5832 - val_loss: 17.8453 - val_accuracy: 0.097
Epoch 2/20
61/61 [=====] - 103s 2s/step - loss: 0.7517 - accuracy: 0.7486 - val_loss: 13.1733 - val_accuracy: 0.097
Epoch 3/20
61/61 [=====] - 103s 2s/step - loss: 0.6426 - accuracy: 0.7691 - val_loss: 13.7582 - val_accuracy: 0.142
Epoch 4/20
61/61 [=====] - 103s 2s/step - loss: 0.5783 - accuracy: 0.7970 - val_loss: 9.4068 - val_accuracy: 0.2086
Epoch 5/20
61/61 [=====] - 106s 2s/step - loss: 0.5657 - accuracy: 0.7930 - val_loss: 6.0481 - val_accuracy: 0.2002
Epoch 6/20
61/61 [=====] - 104s 2s/step - loss: 0.4900 - accuracy: 0.8309 - val_loss: 5.5033 - val_accuracy: 0.2208
Epoch 7/20
61/61 [=====] - 103s 2s/step - loss: 0.3826 - accuracy: 0.8442 - val_loss: 1.8140 - val_accuracy: 0.5716
Epoch 8/20
61/61 [=====] - 103s 2s/step - loss: 0.3463 - accuracy: 0.8786 - val_loss: 1.1552 - val_accuracy: 0.6015
Epoch 9/20
61/61 [=====] - 103s 2s/step - loss: 0.2862 - accuracy: 0.8929 - val_loss: 1.2019 - val_accuracy: 0.6651
Epoch 10/20
61/61 [=====] - 103s 2s/step - loss: 0.2949 - accuracy: 0.8958 - val_loss: 0.6657 - val_accuracy: 0.7624
Epoch 11/20
61/61 [=====] - 103s 2s/step - loss: 0.3047 - accuracy: 0.8924 - val_loss: 0.2598 - val_accuracy: 0.9158
Epoch 12/20
61/61 [=====] - 104s 2s/step - loss: 0.2895 - accuracy: 0.8859 - val_loss: 0.3410 - val_accuracy: 0.8803
Epoch 13/20
61/61 [=====] - 103s 2s/step - loss: 0.2654 - accuracy: 0.8879 - val_loss: 0.2036 - val_accuracy: 0.9280
Epoch 14/20
61/61 [=====] - 103s 2s/step - loss: 0.2250 - accuracy: 0.9140 - val_loss: 0.5937 - val_accuracy: 0.7914
Epoch 15/20
61/61 [=====] - 103s 2s/step - loss: 0.2134 - accuracy: 0.9187 - val_loss: 0.2120 - val_accuracy: 0.9242
Epoch 16/20
61/61 [=====] - 104s 2s/step - loss: 0.2198 - accuracy: 0.9140 - val_loss: 0.7581 - val_accuracy: 0.7259
Epoch 17/20
61/61 [=====] - 105s 2s/step - loss: 0.2231 - accuracy: 0.9163 - val_loss: 0.1621 - val_accuracy: 0.9308
Epoch 18/20
61/61 [=====] - 104s 2s/step - loss: 0.1874 - accuracy: 0.9275 - val_loss: 0.2645 - val_accuracy: 0.8915

```

Fig 5.13-Initial training

## Final 20 epochs optimized with SGD

```
Epoch 1/20
61/61 [=====] - 104s 2s/step - loss: 0.2150 - accuracy: 0.9160 - val_loss: 0.2123 - val_accuracy: 0.9186
Epoch 2/20
61/61 [=====] - 101s 2s/step - loss: 0.1810 - accuracy: 0.9332 - val_loss: 0.1646 - val_accuracy: 0.9401
Epoch 3/20
61/61 [=====] - 100s 2s/step - loss: 0.1692 - accuracy: 0.9344 - val_loss: 0.1513 - val_accuracy: 0.9420
Epoch 4/20
61/61 [=====] - 101s 2s/step - loss: 0.1600 - accuracy: 0.9403 - val_loss: 0.1479 - val_accuracy: 0.9457
Epoch 5/20
61/61 [=====] - 99s 2s/step - loss: 0.1722 - accuracy: 0.9340 - val_loss: 0.1693 - val_accuracy: 0.9242
Epoch 6/20
61/61 [=====] - 99s 2s/step - loss: 0.1851 - accuracy: 0.9302 - val_loss: 0.1501 - val_accuracy: 0.9373
Epoch 7/20
61/61 [=====] - 100s 2s/step - loss: 0.1873 - accuracy: 0.9276 - val_loss: 0.1516 - val_accuracy: 0.9467
Epoch 8/20
61/61 [=====] - 100s 2s/step - loss: 0.1605 - accuracy: 0.9407 - val_loss: 0.1482 - val_accuracy: 0.9411
Epoch 9/20
61/61 [=====] - 100s 2s/step - loss: 0.1582 - accuracy: 0.9375 - val_loss: 0.1435 - val_accuracy: 0.9448
Epoch 10/20
61/61 [=====] - 100s 2s/step - loss: 0.1378 - accuracy: 0.9433 - val_loss: 0.1944 - val_accuracy: 0.9177
Epoch 11/20
61/61 [=====] - 99s 2s/step - loss: 0.1549 - accuracy: 0.9409 - val_loss: 0.1362 - val_accuracy: 0.9439
Epoch 12/20
61/61 [=====] - 100s 2s/step - loss: 0.1578 - accuracy: 0.9411 - val_loss: 0.1418 - val_accuracy: 0.9392
Epoch 13/20
61/61 [=====] - 100s 2s/step - loss: 0.1484 - accuracy: 0.9472 - val_loss: 0.1765 - val_accuracy: 0.9326
Epoch 14/20
61/61 [=====] - 98s 2s/step - loss: 0.1602 - accuracy: 0.9460 - val_loss: 0.1572 - val_accuracy: 0.9448
Epoch 15/20
61/61 [=====] - 99s 2s/step - loss: 0.1384 - accuracy: 0.9504 - val_loss: 0.1395 - val_accuracy: 0.9383
Epoch 16/20
61/61 [=====] - 98s 2s/step - loss: 0.1624 - accuracy: 0.9363 - val_loss: 0.1351 - val_accuracy: 0.9486
Epoch 17/20
61/61 [=====] - 100s 2s/step - loss: 0.1366 - accuracy: 0.9496 - val_loss: 0.1729 - val_accuracy: 0.9261
Epoch 18/20
61/61 [=====] - 100s 2s/step - loss: 0.1670 - accuracy: 0.9332 - val_loss: 0.1367 - val_accuracy: 0.9457
Epoch 19/20
61/61 [=====] - 100s 2s/step - loss: 0.1464 - accuracy: 0.9487 - val_loss: 0.1543 - val_accuracy: 0.9467
Epoch 20/20
61/61 [=====] - 101s 2s/step - loss: 0.1428 - accuracy: 0.9475 - val_loss: 0.1369 - val_accuracy: 0.9448
```

Fig 5.14-Fine tuning

## 5.7.2 LOSS AND ACCURACY GRAPH

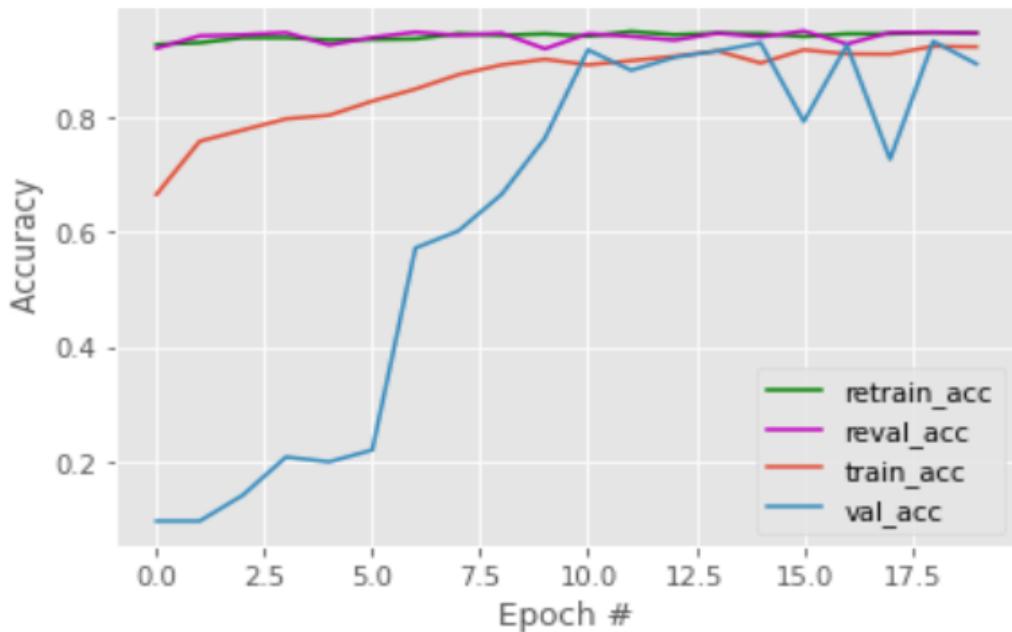
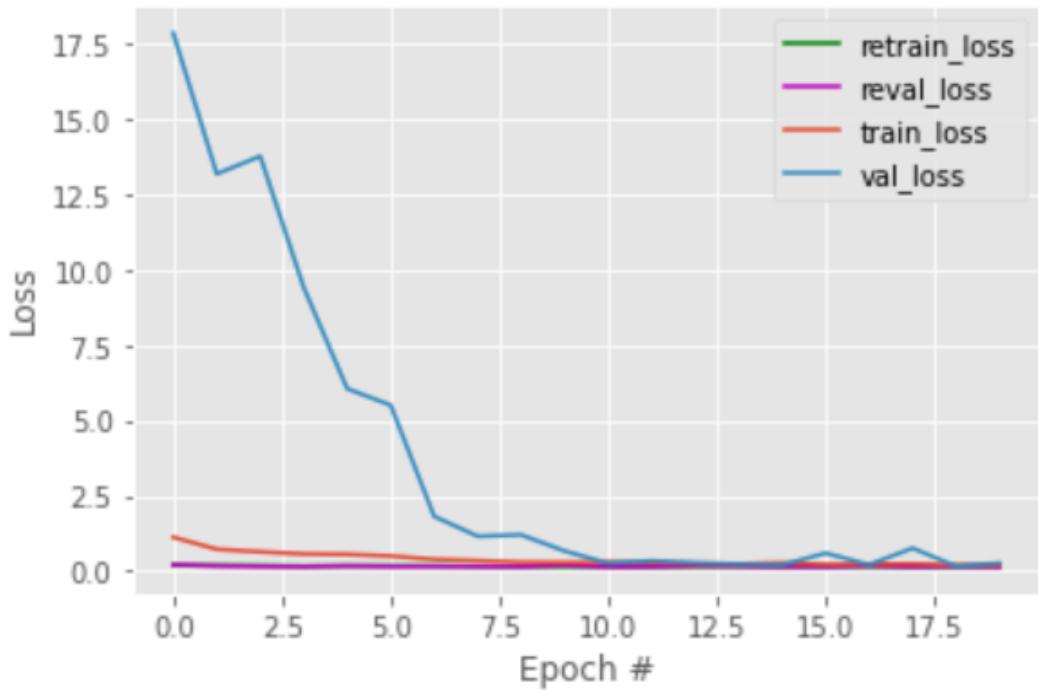


Fig 5.15-Accuracy graph



**Fig 5.16-Loss graph**

## **CHAPTER VI**

## **CONCLUSION AND FUTURE WORK**

### **6.1 CONCLUSION**

The project has been successfully implemented to detect the presence of five types of cancers and also determine its type. In the current state, the process is usually done manually which consumes more time and also involves more human error rate. So, this project aids in reducing the time required for manual classification and eliminates the human error rate by this project. The task to classify large-scale images of the dataset was made difficult by the actual fact that the datasets were extremely imbalanced, and in order to deal with this issue, we have used the paradigm of network surgery and trained the prevailing VGG-16 architecture before applying the learned information to the dataset images. Thus, this project has an efficient scope in the coming future where manual predicting can be converted to computerized production.

### **6.2 FUTURE WORK**

Despite having used five separate datasets, one for each type of cancer, we haven't extracted individual features from each separate cancer type as we have used a generalized algorithm for the same. For example, in the case of cervical cancer, one can extract and use the features from the nucleus in its histopathological image to perform the necessary training. This will reduce the overfitting in the model. The model was obtained by training approximately only 3000 images, but a more generalized model can be developed if we can implement the project with a dataset size that is three folds the size of the collected dataset.

## REFERENCES

- [1] J. M. Anton-Rodriguez, P. Julyan, I. Djoukhadar, D. Russell, D. G. Evans, A. Jackson, and J. C. Matthews, “Comparison of a Standard Resolution PET-CT Scanner With an HRRT Brain Scanner for Imaging Small Tumors Within the Head,” *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 3, no. 4, pp. 434–443, 2019.
- [2] S. Azizi, S. Bayat, P. Yan, A. Tahmasebi, J. T. Kwak, S. Xu, B. Turkbey, P. Choyke, P. Pinto, B. Wood, P. Mousavi, and P. Abolmaesumi, “Deep Recurrent Neural Networks for Prostate Cancer Detection: Analysis of Temporal Enhanced Ultrasound,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 12, pp. 2695–2703, 2018.
- [3] T. Beaumont, P. Onoma, M. Rimlinger, D. Broggio, P. Caldeira Ideias, and D. Franck, “Age-Specific Experimental and Computational Calibration of Thyroid *in vivo* Monitoring,” *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 3, no. 1, pp. 43–46, 2019.
- [4] N. Emaminejad, W. Qian, Y. Guan, M. Tan, Y. Qiu, H. Liu, and B. Zheng, “Fusion of Quantitative Image and Genomic Biomarkers to Improve Prognosis Assessment of Early Stage Lung Cancer Patients,” *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 5, pp. 1034–1043, 2016.
- [5] H. Liu, H. Li, M. Habes, Y. Li, P. Boimel, J. Janopaul-Naylor, Y. Xiao, E. Ben-Josef, and Y. Fan, “Robust Collaborative Clustering of Subjects and Radiomic Features for Cancer Prognosis,” *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 10, pp. 2735–2744, 2020.
- [6] K. Mandal, R. Sarmah, and D. K. Bhattacharyya, “Biomarker Identification for Cancer

Disease Using Biclustering Approach: An Empirical Study,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 2, pp. 490–509, 2019.

[7] A. Mirbeik-Sabzevari and N. Tavassolian, “Ultrawideband, Stable Normal and Cancer Skin Tissue Phantoms for Millimeter-Wave Skin Cancer Imaging,” *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 1, pp. 176–186, 2019.

[8] A. A. Setio, F. Ciompi, G. Litjens, P. Gerke, C. Jacobs, S. J. van Riel, M. M. Wille, M. Naqibullah, C. I. Sanchez, and B. van Ginneken, “Pulmonary Nodule Detection in CT Images: False Positive Reduction Using Multi-View Convolutional Networks,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1160–1169, 2016.

[9] R. Singh, T. Ahmed, A. Kumar, A. K. Singh, A. K. Pandey, and S. K. Singh, “Imbalanced Breast Cancer Classification Using Transfer Learning,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–1, 2020.

[10] Y. Wang, N. Wang, M. Xu, J. Yu, C. Qin, X. Luo, X. Yang, T. Wang, A. Li, and D. Ni, “Deeply-Supervised Networks With Threshold Loss for Cancer Detection in Automated Breast Ultrasound,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 4, pp. 866–876, 2020.