

PREDICTION OF LUNG CANCER USING CT IMAGE

Submitted in partial fulfillment of the requirements for the award of
**Bachelor of Engineering Degree in
Computer Science and Engineering**

BY

SANA PREM KUMAR (38110495)

RYAPATI SAI PRASANNA KUMAR (38110483)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI

SALAI, CHENNAI – 600119.

MARCH 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
Chennai - 600119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONA FIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SANA PREM KUMAR (Reg. No. 38110495), RAYAPATI SAI PRASANNA KUMAR (Reg.No. 38110483)** who carried out the project entitled “**PREDICTION OF LUNG CANCER USING CT IMAGES**” under our supervision from Nov 2021 to April 2022

Internal Guide

Dr. S. Jancy MCA.,MBA., M.Tech.,ph.D.,

Head of the Department

Dr.L.Lakshmanan M.E.,Ph.D.,

Dr.S.Vigneshwari M.E.,Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

We **SANA PREM KUMAR (Reg. No. 38110495), RAYAPATI SAI PRASANNA KUMAR (Reg.No.38110483)** hereby declare that the Project Report entitled **“PREDICTION OF LUNG CANCER USING CT IMAGES”** done by me under the guidance of **Dr.S.Jancy.MCA.MBA.,M.Tech.Ph.D.,** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

We convey our thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean, School of Computing, Dr. S. Vigneshwari M.E., Ph.D. and Dr. L. Lakshmanan M.E., Ph.D., Heads of the Department**, Department of Computer Science and Engineering for providing us the necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr.S.Jancy.MCA.,MBA.,M.Tech.Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **COMPUTER SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

ABSTRACT

There is lot of progress made in the field of treatment of lung cancer in the last years (adjustment chemotherapy, radio therapy, individualized therapy). Nonetheless, lung cancer is still remained the threat of society and cause of death of thousands of people in all over the world. This paper is all about detection of lungs cancer. Here Computer Tomography (CT) images are used to detect lungs cancer. There are several algorithms are used to detect Lungs cancer accurately. Here sharp masking filter is use to filtering the image. Adaptive Canny edge detection algorithm is used to detect the edges and cancer affected areas. Neural network is used to classify the features and predict the probability of lung cancer. K-Nearest Neighbors is used to segment the cancer from lungs.

Past years have experienced increasing mortality rate due to lung cancer and thus it becomes crucial to predict whether the tumor has transformed to cancer or notif the prediction is made at an early stage then many lives can be saved and accurate prediction also can help the doctors start their treatment. Computed tomography plays a vital role in ensuring the condition of tumor that by checking the size of tumor, location of tumor, etc

TABLE OF CONTENTS

CHAPTER NUMBER	TITLE	PAGE NUMBER
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF TABLES	ix
1	INTRODUCTION	1
	1.1 OVERVIEW OF DEEP LEARNING	1
	1.2 PURPOSE OF DEEP LEARNING	2
	1.3 PROBLEM STATEMENT	2
	1.4 OBJECTIVES	3
2	LITERATURE SURVEY	4
3	SYSTEM ANALYSIS	42
	3.1 DRAWBACKS OF EXISTING SYSTEM	42
	3.2 PROPOSED SYSTEM	42
	3.3 SYSTEM REQUIREMENTS	42
	3.3.1 HARDWARE REQUIREMENTS	42
	3.3.2 SOFTWARE REQUIREMENTS	43
4	SYSTEM DESIGN	44
	4.1 ARCHITECTURE DIAGRAM	44

SYSTEM IMPLEMENTATION

		48
5.1	MODULES	48
5.2	IMAGE PRE-PROCESSING	48
5.3	IMAGE SEGMENTATION	48
5.4	FEATURE EXTRACTION	49
5.5	RESULTS	50
	OUTPUT SCREEN SHOTS	51-54
output		
6	CONCLUSION	55
	6.1 CONCLUSION	55
	APPENDIX	54
	A. SOURCE CODE	56
	REFERENCES	65

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF DEEP LEARNING

According to Andrew The theory of Deep Learning has been around for decades but it has only been in the past few years that it has risen and taken off. In our society we have gone from having a relatively small amount of data to huge amounts to the point that we don't know what to do with them. According to , Deep Learning has gained large popularity within different fields of interest ranging from a theoretical research in academia to practical application in industry. Newly emerging algorithms have expanded applications of Learning Machines to be able to compute large amounts of data .with the help of hardware implementations and optimized parallel computational techniques available today. With this Deep Learning Architectures are able to be generalized, customized and configured to advance efficient and accurate models. Compared to other techniques such as standard Machine Learning algorithms ,Deep Learning is a preferred option as performance increases as the data scales. Machine Learning algorithms such as Support Vector Machines, Random Forest and etc, performance plateaus when data increases in these algorithms. Deep Learning Neural Networks works well for my lung cancer detection project as our dataset is large and can iteratively grow larger as more people infer new data to it which will result in a boost in performance. In this chapter, the author explores the intuitions made into creation deep learning models. Finally the result is evaluated using a dice coefficient and confusion matrix metrics. The author reaches a test set of positive and negative samples.

PORPUSE OF DEEP LEARNING

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

How does deep learning attain such impressive results

In a word, accuracy. Deep learning achieves recognition accuracy at higher levels than ever before. This helps consumer electronics meet user expectations, and it is crucial for safety-critical applications like driverless cars. Recent advances in deep learning have improved to the point where deep learning outperforms humans in some tasks like classifying objects in images. While deep learning was first theorized in the 1980s, there are two main reasons it has only recently become useful:

Deep learning requires large amounts of **labeled data**. For example, driverless car development requires millions of images and thousands of hours of video.

Deep learning requires substantial **computing power**. High-performance GPUs have a parallel architecture that is efficient for deep learning. When combined with clusters or cloud computing, this enables development teams to reduce training time for a deep learning network from weeks to hours or less.

1.2 PROBLEM STATEMENT

lung cancer, a bronchoscope with a built-in ultrasound probe may be used to check the lymph nodes in the chest. This is called endobronchial ultrasound (EBUS) and helps doctors determine the appropriate treatment. EBUS may be used for other types of cancer to determine if the cancer has spread. a medical test that lets doctors look into your lungs and airways. The procedure uses a thin tube with a small camera and light at

the end. The doctor inserts it through your nose or mouth, down your throat, and into your lungs. Most of the time, the tube is soft and flexible.

1.3 OBJECTIVE

- There is various cause of lung cancer like smoking, exposure to radon gas, secondhand smoking, and exposure to asbestos etc. Lung cancer is of two type small cell lung cancer (SCLC) and non small cell lung cancer (NSCLC).
- Non-small cell lung cancer is more common than SCLC and it generally grows and spreads more slowly.
- SCLC is almost related with smoking and grows more quickly and form large tumors that can spread widely through the body.
- They often start in the bronchi near the center of the chest.

CHAPTER 2

LITERATURE SURVEY

2.1 Cloud-Based Lung Tumor Detection and Stage Classification Using Deep Learning Techniques

ABSTRACT

Artificial intelligence (AI), Internet of Things (IoT), and the cloud computing have recently become widely used in the healthcare sector, which aid in better decision-making for a radiologist. PET imaging or positron emission tomography is one of the most reliable approaches for a radiologist to diagnosing many cancers, including lung tumor. In this work, we proposed stage classification of lung tumor which is a more challenging task in computer-aided diagnosis. As a result, a modified computeraided diagnosis is being considered as a way to reduce the heavy workloads and second opinion to radiologists. In this paper, we present a strategy for classifying and validating different stages of lung tumor progression, as well as a deep neural model and data collection using cloud system for categorizing phases of pulmonary illness. The proposed system presents a Cloudbased Lung Tumor Detector and Stage Classifier (Cloud-LTDSC) as a hybrid technique for PET/CT images. The proposed Cloud-LTDSC initially developed the active contour model as lung tumor segmentation, and multilayer convolutional neural network (M-CNN) for classifying different stages of lung cancer has been modelled and validated with standard benchmark images. The performance of the presented technique is evaluated using a benchmark image LIDC-IDRI dataset of 50 low doses and also utilized the lung CT DICOM images. Compared with existing techniques in the literature, our proposed method achieved good result for the performance metrics accuracy, recall, and precision evaluated. Under numerous aspects, our proposed approach produces superior outcomes on all of the applied dataset images. Furthermore, the experimental result achieves an average lung tumor stage classification accuracy of 97%-99.1% and an average of 98.6% which is significantly higher than the other existing techniques.

d categorization strategies. The latest lung nodule diagnosis, localization, and classifiers with standard datasets LIDCIDRI, LUNA 16, and Super Bowl Dataset 2016 are familiar with supervised learning algorithms such as SVM, KNN, and CNN. These are the most frequent and typical threshold CT data used for disease diagnosis, according to the authors in [9]. Kalaivani et al. introduced the DenseNet model, a binary classifier based

on a deep CNN network for detecting malignant or benign lung cancer patients [17]. The researchers employed a dataset of 201 lung scans, with 85 percent of the photos being used for training and 15 percent being used for testing and classification. The proposed method obtained an accuracy of 90.85% in tests, according to the results. The authors evaluated histopathological pictures; a classification system was developed to distinguish between 5 types of lung and colon tissues (2 benign and 3 malignant). The obtained results suggest that the proposed model can accurately detect cancer tissues up to 96.33 percent of the time [18]. Qin et al. described how to evaluate EGFR mutation status via computer-assisted diagnostics, which includes extracting, analyzing, and fusing multitype interdependent features. A new hybrid network model built of CNN-RNN architecture is involved in this study. Image quantitative characteristics are extracted using CNN, and the relationship between different types of features is modelled [19]. When compared to the conventional features extracted, their findings showed that multitype dependency-based feature representations outperform single-type feature representations (accuracy = 75%, AUC = 0.78). For supporting radiologists, Masood et al. developed an approach using computer-aided diagnosis support systems for lung nodule diagnosis based on 3D-DCNN. The LUNA16, ANODE09, and LIDC-IDRI datasets were used to train and validate a computer-aided diagnosis (CAD) system in this study [20]. To categorize distinct forms of cancer based on tumor RNA sequences of genomic data, ElNabi et al. offer a unique optimised deep learning strategy based on binary particle swarm optimization with decision tree (BPSO-DT) and convolutional neural network (CNN). The performance criteria such as recall, precision, and F1-score were mentioned in this study [21]. Label-free approaches, according to Abdulgani and Al Ahmad, do not induce cell harm or lead to any changes in cell composition or inherent traits. This research used advances in optical measurements with Prony approaches to improve cell classification using measured optical profiles [22]. By the improvement of Tobacco Exposure Pattern (TEP) classification models and uncovering their interaction linkages at multiple biological levels, He et al. were able to identify signature genes. TTZ is a new method for extracting existing features as an input variable to the TEP classification model [23]. To train and evaluate the TEP classification model, with two independent LUAD datasets, 34 genes were recognised as tobacco-related mutation signature gene, with an accuracy of 94.65 percent for training data and 91.85 percent for validation data. Table 1 represents the summary of the recent models and dataset used for performing various algorithms.

3. Proposed Methodology

To achieve accurate result in terms of “accuracy, precision, and recall,” the recommended Cloud-LTDSC hybrid framework includes lung tumor detector, lung

tumor segmentation, and stage classifier modules. Figure 2 shows a summary of the suggested model, followed by descriptions.

3.1. Dataset

3.1.1. Image Acquisition. F-FDG PET/CT scans of 94 individuals with NSCLC were used in this investigation. Between March 2010 and April 2014, the images were taken. Siemens Biograph 6, HiRez, was the PET/CT imaging device used. FFDG was administered into the patients in doses of 10 to 15 mCi, and PET/CT scans took about 60 minutes. PET scan was done for every 2 to 3 minutes of the 8 or 9 bed positions. The photos were reconstructed using a three-dimensional iterative reconstruction method. Using eSoft software (eSS) platform, there are two experts in the field of nuclear medicine that reviewed the photos and for an entire body projection (Siemens, USA). The Research Ethic Committee of KRTH accepted this work. All study participants were given the opportunity to give their informed consent. There were seven female patients and 92 male patients (average age was 59; interval of age is between 36 and 82). There has been no research on tumor variability in male and female NSCLC patients until now. As a result, we did not examine the effect of gender on cancer subtype characteristics when designing this study. Forty-five patients were found to have an ADC, while the others were found to have SqCC. The Lung Image Database Consortium (LIDC-IDRI) data collection the DICOM CT lung images [24]. Each DICOM lung CT scan is saved in the DICOM file formats, which is having 512×512 pixels in size. Images of tissue slices with a thickness of 1.15 mm to 2.75 mm and a diameter of 0.45 mm to 0.75 mm are included in this dataset [25]. Each radiologist saw all CT scans individually and labelled lesion that fit into one of three categories nodules, seminodules, and nonnodules. Four radiologists looked at the CT scans and identified the tumor types. This LIDC-IDRI dataset included 1024 cases, 358 CT scans of lung nodules, and an XML file with the results of a 2phase of image-annotation processes by a radiologist. A total of 623 pictures of lung nodule-free and nodule-filled slices were used. In this model, nodules with diameters ranging from 4 to 20mm ($125 \text{ pixels} \times 125 \text{ pixels}$) were used. The LIDC-IDRI data collection was used to create all DICOM CT lung images with solid nodules, semisolid nodules, and nonsolid nodules referred by few radiologists. Selected nodules were classified as “well-circumscribed, juxtavascular, juxtapleural, pleural tail.” The ACM approach centred a region of interest (lung tumor) with a nodule size of $75 \times 85 \times 45 \text{ mm}$. As described by a few radiologists, the size of the nodule may be retrieved, and the number of voxels that engaged all dimensions can be determined. Then, they pointed to the nodule’s centre, and 55 squares were suggested as the zero-level set using the Signed Distance level set Function (SDF), which aids in the conversion of

lung images into 3D shapes. The real and simulated nodule characteristics are listed in Table 2. The total number of images with

respect to solid nodules, semisolid nodules, and nonsolid nodules is given in Table

3.2. Lung Tumor Segmentation Using Active Contour Model. In comparison to other organs such as the brain, detecting lung nodule in a DICOM CT lung imaging is tough. Likely, bronchus and intensity of lung nodule identification in a CT DICOM lung image are tedious. The bronchus and intensities of blood vessels within the tumor area are then measured, and the region of lung parenchyma is partitioned from back to front. The steps involved for the detection of lung tumor are as follows:

(1) Remove the region of mediastinum and thoracic wall
if the lung parenchyma was not reconstructed

(2) The tumor part of the lung picture is segmented
using the active contour model (ACM)

(3) A total of 925 lesions from different patients were
collected, together with nodule information

The number of different forms of benign nodules is shown in Table 2. Due to a gradient value, the edge-based ACM causes some segmentation modifications. The leaking problem occurs when there is a weak boundary in the image [26]. The approach of edge-based segmentation is not possible due to noise in the image border. According to Rouhiet al., border leakage does not occur in region-based ACM, and undesired sections of the picture are removed. When

the roundness rule was applied to the objects, the result was approximately 1, resulting in a solid nodule [27]. Snake's model was utilized to develop a curve that aids in the detection of a tumor portion in the relevant photos. The

curve must be drawn around the identified item, and then, it must alter its location towards the interior and end at the object borders [28]. For each contour line that gathers a similar amount of control point and distinct point, the initial prediction model of active contour is utilized; it predicts contour point that helps for the contour lines and then generates the active contour model's first prediction [29]. The proposed approach generates 3D features from

2D stochastic characteristics, which are then fed into the CNN classifier.

Considering gray-scale DICOM image $\frac{1}{2}dy$, $z \in R^2$, the contour of the segmented portion was mentioned as In the proposed ACM, we reduced the number of points to fit the curve in the tumor portion which is segmented. Let

the sample contour C_k be into the given n points, and the overall curve U_k was given by

The modified ACM is evaluated using a gradient value which helps for an edge detection. It is derived from the image as α , β , and μ are constant, and E_k refers to the parametric curve with the k th image. Using the Mumford-Shah model, we derived an

expression for intensity outside and inside curves with level set method and energy approximation AE is given by,

3.3. Cross-Validation Set. In this paper, 10-fold crossvalidation method is widely utilized for the LIDC-IDRI/ PET dataset with performance metrics which is shown in Figure 5. The dataset was erratically splitting in this work into stratified 10-fold cross-validation, which has been employed in many deepconvolutional neural networks and traditional machine learning-based bioimage and biosignal studies. M-CNN model was trained from scratches

and also starts with some random weight using a processor NVIDIA GeForce GTX 1650 PC for a High-Performance Computing (HPC). The LIDC/IDRI and PET scan images such as 1018 DICOM CT scans and 355 PET scans were grouped as 10-fold cross-validation sets. In this set, 9 sets are for training and 1 set is for validation with a repeat of 15 epochs. The number of epoch for overall training processes will be 150 with a batch size of 10 from the given dataset, and early stopping approaches and also regularization were used to prevent from an overfitting. In the crossvalidation set, we fold our dataset into 10-fold set of images. In fold 3, fold 5, and fold 7, we have used PET scan images for testing; we got less accuracy than other fold sets as we achieved. From Figure 6, the performance analysis of training data loss and validation data loss with respect to 60 epochs for 4-fold validation set is clearly shown. And also, we clearly show the accuracy analysis with respect to 60 epochs for training and for 4-fold validation data in Figure 7. Table 4 shows the performance metrics of accuracy, precision, and recall for 10-fold cross-validation with the mean value.

3.4. Lung Tumor Stage Classification. Artificial neural networks are stimulated by the human brain which meets machine learning for solving complex problems. So, machine learning has a subset called deep learning (DL). DL method is used to draw out features from bulk data; collecting useful information from big data using DL algorithms is useful in many aspects. Since detecting a feature is time-consuming and costs a lot, DL methods do not require labelled data for learning objectives. In this case, we may have both types of data labelled and unlabelled in

healthcare like CT Scan images regardless of a medical condition, huge data that is not labelled, etc. The classification based on image database is shown in Figure 8. tion, we discussed some of the widespread techniques among them. (1) Artificial neural network (ANN) is a DL technique that contains several organized layers which have perceptron, the neurons. The convolutional layer is one of the most important components of CNN. Convolution is a

mathematical procedure for combining two sets of data. Convolution is connected to an input data using convolution filters to create a feature map. Convolution is accomplished by sliding this filter over the input. We conduct elementwise matrix

multiplication and total the results at each position. Pooling is frequently done after a convolution process to minimize dimensionality. This allows us to limit the number of parameters, reducing training time and preventing overfitting. Each feature map is down sampled independently using a pooling layer, lowering the height and breadth while maintaining the depth. Dropout is only applicable to input and hidden layer nodes. The edges to and from the nodes have been thrown. The use of multilayer network is for the hidden layer in which neurons are not directly linked to the output layer and also used in multilayer network to address the classification issue for a nonlinear data. The hidden layers will be understood geometrically as an extra

hyperplane that increases the network layer separation capability. Multilayer networks were once utilized in computer vision, but convolutional neural networks have now taken their place. This network is no longer considered as adequate for sophisticated computer vision applications. Each perceptron was connected to any other perceptron, giving it the property of completely connected layer. The total number of parameters might get quite large which are all the disadvantages. Even though there is duplication in such large

dimensions, this is ineffective. It also ignores geographical information, which is a drawback. Its inputs are flattened vector. Multiple layers are starting from the input layer then the hidden layer; actually, this works as the training layer and output layer. If hidden layers are increased vastly, then it may not assure improved results. Overfitting also takes place if many layers are added at the same time; as a result, many distortions forward too much interference in data

being captured. Convolutional neural network (CNN) until now highlighted beneficiary technique in healthcare (Figure 9) [27]. A static extent flight path is using as an input. It can be applied to medical data like in image processing for the detection of the lung tumor. Perceptron interconnected with each other and assigned a weight to them that can be adjusted after each iteration. As the propagating waves are transmitted in one direction, i.e., starting from input meanwhile ending on the output layer, this is known as a feed-forward network. Convolutional neural network (CNN) is enthused by the human cortex and is one of the supreme prevalent deep learning methods among others. It is considered as a feed forward network containing multiple layers that implies in one way from input to output. When data is passed through the process layer, useful features are extracted from input data, and the result is shown in output layers. In healthcare, it is used for disease detection from sample tissues. It promotes read structures usually difficult to understand by human medical experts. In the proposed CNN architecture, it results in the classification of lung tumor stages as 9 classes which is described from Figure 10. The DICOM lung image and PET scan image were given as input to our proposed M-CNN method. Herewith, we have used six CNN layers and each layer consists of 1 convolutional layer and 1 max-pool layer. The proposed M-CNN algorithm helps to classify the lung tumor stages with 3×3 kernel function which were

examined. We will train our model in each of the 30 epochs. We can reduce our learning rate μ in each of the 10 epochs with the factor value of 10. At first, our input image of (height $\times$ width $\times$ depth) $256 \times 256 \times 8$ pixels is given to our M-CNN with dropout 0.2 and 4096 units of fully connected layer used for classification and leads to 97% accuracy in the lung tumor stage classification. Therefore, our proposed method helps to reduce the error function and for accurate lung tumor that, CNN with 2 layers was developed, and their findings were evaluated. The method had been used until the model's output was effective and more accurate. Six convolutional layers with max-pooling made up with ReLU and got a final M-CNN model, which would have been extremely viable based on output results. In the outcome portion, the parametric results of every model increment have also been provided with the more accurate result in lung tumor stage classification. In terms of stage classifications, lung tumor is distinct among other tumor types. The majority of lung tumor stage classification methods are mostly based on experience and consensus. The International Association for the Study of Lung Cancer (IASLC) created a lung tumor stage classification system based on detailed statistical methods of a global database of more than 100,000 cases. This research was presented with TU, LN, and DM components, as well as the stage categories (Tables 5 and 6), methodology, and testing for non-small-cell lung cancer and small cell lung cancer. In Table 7, classification results of performance metrics such as sensitivity, specificity, and accuracy were given by the MCNN technique with ACM and without the ACM model. These results show that the stage classification with the ACM model achieves more accuracy than without ACM. In this work, we used 9 stages of classification of lung tumor with respect to size and location. If we send the CT/PET scan images to the M-CNN model, we got the less accuracy in stage classification. Therefore, we used the ACM model for tumor portion segmentation and sent that segmented image to our M-CNN model to achieve more accuracy.

3.5. LC-Cloud Framework.

This section described about the proposed new LC-Cloud system as a lower-level server based on "Saas" used for analysis and storing the patient's diagnosis records for study purpose and for virtual diagnosis purpose (i.e., E-treatment). The main motivation of this LC-Cloud is (i) to improve the diagnosis performance of E-treatment in regional medical care system, (ii) to make use of this patient's records for second opinion or clinical study purpose, and (iii) to make fast processing in real time with low cost. Generally, cancer patients are diagnosed through multiple physicians from various sectors such as neurology, surgeons, and cancer specialist with the help of their personal records which have been stored in their local servers. Conventionally, their record is comprised of numerous images (either DICOM or PET or thermal imaging formats) which are massive computational intensive and demand more memory for storage purpose. In the current scenario, many local workstations are used for such analysis and storing the medical records which greatly rises the cost for installation, maintenance, and reconfiguration process. To address these complexities, researchers are initiated to

adopt the cloud computing services due to its ability of fast processing, high-level storage, flexibility in handling the data, and also easily accessible through the internet worldwide. This significance of cloud systems has been adopted in the medical care industry. Liu et al. developed a new cloud server called “iMage” for particularly processing the medical data [30]. The proposed cloud has three distinct layers for analysis, processing, and communication purpose in order to provide the software services. The limitation of this cloud system is mainly focused on security issues in transferring the medical data. Similarly, Ojog et al. proposed the “m3Dicom” server which is dedicated for handling the dental segmentation process. Parsonson et al. also developed a dedicated cloud server called “bCloudronics3D” model which stored the volume rendering data [32]. These existing servers are focused on either analysis or storage purpose of medical data but not mentioned for lung cancer detection process. Motivated by these models, we developed a dedicated LC-Cloud framework for lung tumor analysis, processing, and storage purpose as a lower level public cloud server. The proposed LC-Cloud system is developed with multiple nodes and communication layer for transferring the data through the internet. The LCC loud server architecture is comprised of three layers called “input, processing, and storage layer” which is connected with the firewall and LC-server. Figure 11 illustrates the conceptual architecture of the proposed LC-Cloud system. The input layer is designed to adopt any category of lung tumor data such as “DICOM images or PET images or EMRs” directly accessible through users or radiologists through the internet. These data are feed-forwarded into the next layer for processing under certain predefined conditions. The new user data is checked with the LC-Cloud engine for availability match. The conditions are (i) if the new user record is matched with existing file, then it is directly forwarded to physicians and (ii) if the data is not matched, then it is sent to the processing layer for diagnosis and sent to the physician. Finally, the physician results and patient records are gathered in the storage layer for clinical study and for virtual treatment process. The processing layer consists of intellectual algorithms called “ACM segmentation algorithm and M-CNN classifier” for diagnosis purpose.

3.5.1. LC-Cloud Significance

- (i) Useful for patients in terms of fast processing particularly in emergency situation
- (ii) The records are analyzed worldwide due to its distributed sharing
- (iii) Beneficial for both physicians and patients in terms of cost
- (iv) Avoiding the data collection and transferring repetition process which saves the analysis time and improves the performance
- (v) Easily accessible
- (vi) Lower-level structure with limited maintenance cost
- (vii) Used for clinical study purpose or for future references

The LC-Cloud system is beneficial for both physicians and patients in real time. For example, with the use of 4G and 5G connectivity; patients can take precautions from the

doctor without visiting and it provides real-time data to the doctor for best treatment. By availing sources like mobile networks, data can be retrieved and sent over devices remotely to physician, radiologists, and external specialists concurrently. Figure 12 illustrates the working structure of the proposed model for new or unknown lung tumor patient record for analysis. LC-Cloud specifications: the proposed cloud is comprised of three clusters “input cluster, processing cluster, and storage cluster” where each cluster includes different numbers of processors. Each layer includes 4-core processor with 64 GB memory, and these nodes are accessed through a main server. The input data is accessed from a standard database and also adopted from Amazon web service cloud and processed through hardware of LC-Cloud engine. The limitation of the proposed cloud system is capable to analyse only 65 users’ record concurrently due to its memory limitation which is not much efficient for large hospital structure. The proposed system can be extended with multiple nodes in the future for large hospitality frameworks with high level cost, and security issues can be focused in the future.

4. Experimental Results and Discussion

Simulation environment: the proposed Cloud-LTDSC framework is developed in “Python IDE” with deep learning modules, and various performance metrics are analyzed. Experimental setup: LIDC-IDRI and PET scanned images are adopted for pre training and testing purposes. The initial analysis such as noise reduction and image segmentation is performed in the hardware PC with i7 processor with “NVIDIA JETSON GPU System-on-module with 256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores” and memory 64 GB 128-bit LPDDR4 Memory 1866 MHz -59.7 GB/s.” Cloud service: the proposed models with two standard databases are created, and stage classification is performed on Amazon web cloud service due to large database and high-level processing speed is required. The AWS cloud has been used in LC-Cloud system that processes and collects the classification reports that are stored permanently for future diagnosis .Each layer includes 4-core processor with 128 GB memory, and these nodes are accessed through a main server. The input data is accessed from a standard database and also adopted from Amazon web service cloud and processed through hardware of LC-Cloud engine. From Figure 13, it gives the information about the estimated performance metrics for validating the proposed Cloud-LTDSC model. From Table 8, we compared our performance metrics with the results of an existing system in terms of accuracy, sensitivity, specificity, FPR, and AUC. The performance analysis of testing experiments TS-1, TS- 2, TS-3, TS-4, and TS-5 with accuracy (%) of TP, FP, and prediction accuracy from the Figure 14. Each test set is varied with training and testing samples in the range of 20-70% images in training and 10-40% images in testing. The comparative analysis of our proposed model with existing SVM models to deep learning networks is proposed by other authors mentioned above in related works. The proposed model outperformed “10 distinct modules—“AlexNet, CNN, SVM, KNN, RF, 3D-CNN, DenseNet, BPSO-DT, TEP, RNN,”

proposed by other authors mentioned in summary table. The metrics are measured for all 11 modules including the M-CNN with ACM segmented model with the performance measures like “accuracy, sensitivity, and specificity” which are shown in Figures 15–17 Figure 18 illustrates the classification accuracy obtained by M-CNN and other deep learning classifiers. The range of accuracy is varied in between 97% and 99.1% for all five test sets with varying training and testing samples. This level of accuracy is achieved in lung tumor stage classification because of exact segmented inputs feed-forwarded into the multilayer network which detected the different classes based on their independent features of each and every segmented image. The test set 2 contained few class 3 images and more class 1 images. Similarly, multistage classification was performed on five distinct test sets which accurately categorized 9 distinct classes. Figure 15 shows the comparative analysis of sensitivity 95.9% with existing methods. From Figure 16, it shows the comparative analysis of specificity 93.9% with existing methods. After comparing the performance metrics of our proposed cloud-based healthcare application to existing cloud methods, it can be shown in Figure 17. This happens due to an employment of excellent lung tumor segmentation and also for tumor stage classification approaches, which aid in creating an accurate tumor prediction choice using both traditional lung tumor images and cloud data gathered from remote lung tumor patients. The total elapse time of the proposed method is completed within 2700 seconds, and the time complexity is $O(n^2)$, where n is the number of pixels in the segmented image.

5. Conclusion

In order to lower the lung tumor death rate, this paper focused on constructing a cloud-based lung tumor detector that includes a segmentation module, detection module, and stage classifiers. The suggested Cloud-LTDSC module used unsupervised learning neural networks as a predictor and stage classifier to more precisely identify the tumor. The segmented images are feed-forwarded into the MCNN model in order to identify its classes and severity level for each patient. The e-record has been generated with patient name, severity level, and tumor details which are transferred through cloud to doctors for virtual monitoring and E-diagnosis automatically. The performance metrics are estimated in terms of accuracy, sensitivity, and specificity and compared with the existing techniques. The results Proposed method Various metrics prove that 97% accuracy has been achieved in stage classification for tested segmented images. These deep learning and smart lung tumor models detect the tumor in the earlier stage using the effective virtual monitoring and E diagnosis. In future work, we will develop and analyse this proposed system for large database and then publish the findings on how the system is capable of safely managing hospital large data and also, this prototype is planned to develop as the complete cloud hardware module in the future work.

Data Availability

The LIDC-IDRI data used in the findings of this study is free publicly available dataset

from National Cancer Institute and the Foundation for the National Institute of Health.

2.2 Prediction of Cancer Disease using Machine learning Approach

Abstract

Cancer has identified a diverse condition of several various sub types. The timely screening and course of treatment of a cancer form is now a requirement in early cancer research because it supports the medical treatment of patients. Many research teams studied the application of ML and Deep Learning methods in the field of bio medicine and bio informatics in the classification of people with cancer across high- or lowrisk categories. These techniques have therefore been used as a model for the development and treatment of cancer. As, it is important that ML instruments are capable of detecting key features from complex datasets. Many of these methods are widely used for the development of predictive models for predicating a cure for cancer, some of the methods are artificial neural networks (ANNs), support vector machine (SVMs) and decision trees (DTs). While we can understand cancer progression with the use of ML methods, an adequate validity level is needed to take these methods into consideration in clinical practice every day. In this study, the ML & DL approaches used in cancer progression modeling are reviewed. The predictions addressed are mostly linked to specific ML, input, and data samples supervision. 2021 Elsevier Ltd. All rights reserved. Selection and peer-review under responsibility of the scientific committee of the International Virtual Conference on Advanced Nanomaterials and Applications.

1. Introduction

The main weight of ailment overall is as Lung malignancy that is the most inescapable disease in the two men and women [1]. A few other reports estimate some 221,200 new cases of pulmonary cancer occur and represent approximately 13% of all cancer diagnoses in 2015. Approximately 27 percent of all cancer deaths are attributed to lung cancer [2]. Lung nodules must therefore be closely examined and monitored when at an early stage. In this study, the ML & DL approaches used in cancer progression modeling are reviewed. The predictive models discussed here are based on different supervised ML techniques, input and data samples. A Local Binary Pattern (LBP) is an image operator that transforms an image into an array or picture of integer labels that describe the appearance of the small picture. These labels are then used for further image analysis, most frequently in the histogram. The LBP texture operator has become a popular approach to various applications thanks to its discriminative power and computational simplification [2]. A Binary Local Pattern (LBP) is a picture administrator that changes over a picture into

a variety of number names speaking to its essence. These markers are then more commonly used in

the histogram for further image processing. In the last three decadent years the prevalence of prostate and breast cancer in male and female cancer has been the largest, but lung cancer remains the highest in cancer-patient mortality [3]. One of the main reasons for this is that prostate and breast cancer prognostic models are comparatively more advanced and systemic than pulmonary cancer. Thus, it is urgently necessary to establish an effective early

stage lung cancer forecast model. In linear and non-linear probweight of the misjudgment for the majority is higher than that of the minority class and is likely to cause misjudgment. Traditional

algorithms of classification are not successful and excellent. 1.1. Topology of machine learning & deep learning algorithms In order to predict the various types of diseases, different deep learning & machine learning algorithms are used, such as Support vector machine (SVM), Neural Network (NN), LR, Nevin biases (NB), Fuzzy logic, transfer learning, ensemble learning, Transduction learning, KNN, and Adaboost are mostly utilized in diverse contributions. Moreover, SVM is categorized into Boosted SVM & MLSVM for predicting distinct diseases in the earlier contributions. Similarly, NN is classified as Dynamic Neural Network (DNN) & Convolution Neural Network (CNN) which are employed for diagnosing different diseases in different contributions. Moreover, GBDT is a modified form of DT, CVIFLR is the modified form of LR that are used for detecting diseases. Moreover, RF and Fuzzy logic is grouped into HRFLM and Fuzzy SVM, respectively in order to predict discrete diseases in various contributions. So, for predicting lung cancer in an efficient manner with the help of improved machine learning techniques can be uselems, SVM has superior predictor performance and is widely used in various fields including in medical matters. Even if SVM is a

superior classifier, the field of cancer prognosis models is relatively immature [4]. The mutation test [5] has become an important tool for deciding the right therapy options for patients in clinical tests. Direct sequencing is an alternative approach for unknown mutations based on screening. The Mutation Test for Epidermal Growth Factor Receptors (EGFR) has been identified for lung cancer genetic mutation testing [4]. A contrast with their non-ensemble variants of two types of categorizing equipment Artificial Neural Network (ANN) and Support Vector Machine (SVM) is published. The 2.1. Analysis of ML applicability's in cancer A comprehensive search of ML techniques in cancer sensitivity, recurrence and survival predictions was conducted. The PubMed, Scopus, has entered two online databases. Further review was required due to the large no of blogs which were found by the search queries Most of the studies use different input data types: Clinical, genetic, histological, imaging, socioeconomic, epidemiological or mixed details. According to [13] and their survey based on ML use

of cancer prediction, we have seen a large rise in documents released in the last decade. Discuss, we selected from the first category of papers a representative list following a welldefined structure. We have selected such studies in order to prevent the desired effects, especially with the implementation of recognisable ML techniques and integrated data through heterogeneous data. [Tables 1](#) indicate some publications in this study. Each suggested approach specifies the type of cancer, the ML treatment, number of patients, type of data and overall precision achieved. Each table corresponds to a particular scenario of study. i.e., prediction of cancer sensitivity, forecast of cancer occurrence and prediction of cancer survival. It should have been acknowledged to introduce the most precise predictive model here in publications which apply more than one ML technique to prediction. Different study projects have attempted to forecast cancer regeneration after remission and have managed to boost predictions correctly in contrast to alternative statistical techniques. In addition, molecular and clinical data have been used to estimate the large bulk of such papers. The implementation of observed behaviors such as input data is a growing phenomenon, based on the growth of HTTs.

2.2. Case study 1

Application of machine learning to predict the susceptibility of cancer risk from the 79 papers surveyed in this study are relative limited (only 3). The development of a retrospective methodology

to predict the presence of 'spontaneous' breast cancer using single nucleotide polymorphism (SNP) steroid metabolizing enzymes (CYP 450) is among the interesting documents. Close. Sporadic and non-family breast cancers account for 90% (Dumitrescu and Cotarla, 2015). This trial was conducted with the theory that environmental toxins or hormones were accumulated in breast tissue and that some combinations of the SNP gene were at increased risk

of breast cancer. The authors have obtained data on 63 breast cancer patients and 74 breast-free (controls) patients from the SNP (98 SNP from 45 cancer-associated Genes). It was vital to the of this research that researches used various methods to minimize a sample-per-feature ratio and analyzed several processes of machine training in order to find optimum classification. In particular, the authors rapidly reduced this set from a start set of 98 SNPs to just 2–3 SNPs that appeared to be as informational as possible. Instead of almost 3:2 (with all the 98 SNPs used), the specimen ratios were reduced to 45:1 (for 3 SNPs) and 68:1 (for 2 SNPs). This made it possible to prevent the “dimensionality curse” from being affected (Bellman, 1961; Somorjai et al., 2013). When the testing sample gets minimized, a number of machine learning techniques, consisting of a naïve Bayes model, various decision-making methods and a sophisticated SVM were applied. With just a set of three SNPs the SVM and naïve

Bayes classifier were maximum in precision. The decision-tab classifier achieved maximum accuracy with a set of two SNPs. The SVM classification was the optimum, along with a precision of 69%, and 67% and 68%, respectively, were found in the naive Bayes and Decision Tree classification systems. The outputs are about 23–25% better than original. The extensive level of cross validation and confirmation conducted was another notable feature of this study. At least three ways have been validated for each model's predictive power. Firstly, model training with 20-fold cross validation has been evaluated and monitored. A bootstrap resampling approach was used when the cross validation is performed 5 times and the outputs were averaged to keep the stochastic dimension in the division of samples to a minimum. In addition, the selection process was carried out for 100 times in each fold (5 times for each of 20 folds) in order to reduce unequality in function selection (i.e. selecting the most informative SNP

sub-ensemble). Thus, the outputs are then matched with an altered permutation test that, had 50 percent predictive precision. While the researchers tried to reduce the stochastics in sample partitioning, it could have been better to use leave-one-out cross-validation that shall have completely deleted this stochastic element. This trial was conducted with the theory that environmental toxins or hormones were accumulated in breast tissue and that some combinations of the SNP gene were at increased risk of breast cancer. The authors have obtained data on 63 breast cancer patients and 74 breast-free (controls) patients from the SNP (98 SNP from 45 cancer-associated Genes). It was vital to the progress of this research that researchers used various models to minimize a sample-per-feature ratio and analyzed several methods of machine training in order to find optimum classification. It also points out the way in which machine learning can disclose significant information into the biology of spontaneous or non-familial breast cancer and polygenic risk factors.

2.3. Case study 2

Cancer Survival Prediction Almost half (or 1 year or 5 years survival rates) of all the machine study studies on cancer prediction. One report of a specific interest (Futschik et al., 2013) was used to predict the outcomes of diffused large-B-cell lymphoma (DLBCL) by hybrid machine learning. In particular, both clinical and genomic (microarray) information was gathered into creating one clinical classification to predict DLBCL survival. The approach is somewhat different from the Listgarten et al. study (2014) in its classification scheme, which only used genomic data (SNP). Futschik et al. hypothesized, rightly, that clinical knowledge may improve data on microarrays to a better output than a microarray-alone or clinical data-based classifier. In addition, different kinds of “Evolving Neural Network” (EFuNN) classifiers have been produced to manage genomic data, separate from the Bayesian classification system. A mixture of 17 genes from the microar

ray data is used by the best EFuNN classifier. The accuracy of this best EFuNN was 78.5%. In order to achieve consensus prediction, the EFuNN and the Bayesian classifiers were mixed in a hierarchical modular structure. This hybrid classifier has a precision of 87.5 per cent, significantly improving both classifications' performance alone. This was 10% good than the excellent-performing classification for machine learning (77.6% by VMS). A cross-validation strategy for the EFuNN classifier was applied. Possibly because the sample was small. No external validation collection was present to check for the overall system, as with Case Study # 1. The Sample per Feature Ratio (SFR) is above 3 with just 56 patients (samples) categorized using 17 gene features. SFR below 5 do not always necessarily ensure a robust classification (Somorjai et al., 2013). Moreover, it is obvious that the researchers were known of this problem and spent enough time explaining in depth the internal functioning of their classifier to justify their approach. This consisted of a summary of how the Bayesian classification was constructed, how the EFuNN works and how the two classification systems cooperate to make one prediction. Also, the researchers tested the independence of the micro-array knowledge from clinical data and subsequently verified it. This eye for detail is particularly outstanding in such a study. The whole research reveals how well the capacity to use both genomic and clinical data significantly improves cancer prediction accuracy.

2.4. Case study 3

The Laurentian and the other in a particularly good example, and also discusses some of the inconveniences observed in existing researches. The authors wanted to predict the possibility of recurrence in patients with breast cancer for five years. Seven predictive variables have been combined, comprising of clinical information like patient age, tumor size and no. of axilla metastases. Protein biomarkers, like oestrogen and progesterone receptor levels, also received information. The focus of the research was to produce an automated, quantitative predictive approach more precise than those of the classical metastasization of the tumor node (TNM). TNM is a group of medical experts that rely majorly on the professional judgment of a pathologist or clinician.

The researchers used an ANN model, were using information from 2441 breast cancer patients (each time seven data points). A sample to feature ratio remained significantly higher than the recommended minimum of five (Somorjai et al., 2013). The whole dataset was divided into 3 classes: training (1/3), testing (1/3) and test sets (1/3). Furthermore, the authors have collected 310 separate samples from another organization to carry out an external assessment of breast cancer patients. This helped the researchers to test the generalization of their system out beyond ones institution — a phase not

taken in the 2 experiments discussed above. The analysis demonstrates not only the volume of data and the thoroughness of validation, but also the level of quality control for data processing. The information, for example, was decided to enter and collected autonomously in a connection database and was autonomously checked to keep the referring doctors

in good standing. The samples of 2441 patients and 17 000 data points were sufficiently large for a typical breast cancer population demographics when subdivided into the data sequence. However,

by examining data distributions for patients in each set (training, monitoring, testing and external), the authors explicitly verified this assumption and demonstrated that distributions are quite same. The Authors built an extremely accurate and robust classifier through consistency and attention to detail Since the study's aim is to produce a system that better predicted re-currence of breast cancer than the traditional TNM stalemate method, comparing the ANN model with the TNM stalemate predictions was important. This was done by using an Operator Characteristic (ROC) curve to compare the performance. The ANN model (0.726), calculated by the portion in the ROC curve, exceeded the TNM system (0.677). This research is an brilliant illustration that machinery is

well articulated and tested. A large enough set of data was obtained and data was tested for performance assurance and precision for each sample independently. In addition, blinded validation sets were available for assessing the generality of the machine learning system both from the original data set and through an external point. Finally, the precision of the model has

been contrasted directly with that of the traditional TNM projection scheme. Thus the only challenge to this analysis was that ing a new variant. The ability of machine learning to solve compos

ite tasks with dynamic environment and knowledge has contributed to its success in prediction research especially lung cancer, enabled with novel met-heuristic algorithms. Although there are many advantages for predicting the lung cancer, but still there are few defects with the existing methodologies so that a new method needs to be implemented. Adaboost [31] has attained high sensitivity and best performance, and it is very simple to implement. But, it is very sensitive to noisy data. DT [32] is simple to interpret, it should be taken as the minimal decision standard of work-relatedness for lung cancer, is the best predictor by attaining high accuracy, and it automatically prunes to a very short three-level depth. However, the running time of training algorithms do not scale well with the size of the training set. SVM [34] is used to build n-hyperlanes and n-features for dividing each different class apart from maximal margin, and it improves the classification power and robustness. Yet, many parameters need to be set accurately for attaining the best results. Gene Expression Programming [35] has the better solution for predicting lung cancer difficulties, and has high accuracy. However

there are some disadvantages such as if they are easy to manipulate, they lose in functional complexity. Dynamic Bayesian Networks [36] has demonstrated high discrimination and predictive power, and it is used to acquire the probability of positive outcome of a biopsy for the given individual. Though there are few challenges like if there is longer search time, the performance might be affected. POMDP [38] optimizes the lung cancer prediction during the improvement of test specificity, and it reduces the false positive rates. But, the performance needs to be improved. Hence, the new model needs to be introduced for providing best performance so that the above conflicts are useful for the new development method.

4. Research objectives

The objective of this research work is discussed as follows.

1. To review on various state-of-the-art lung cancer prediction models and develop a new feature extraction model.
2. Compare the symptoms of cancer for early notification.
3. To design and develop a deep learning model to predict the lung cancer.
4. To validate the proposed model by comparing it with other conventional models.
5. Sending Automatic notification for detecting the cancer.

.

Discussion

In The latest research on predicting cancer using ML & DL techniques are discussed in this study. Further through the short details of the ML & DL field and the preprocessing data techniques, the selection techniques and the classification algorithms were employed, we discussed three specific case studies based on popular ML tools, concerning foretell of the susceptibility of cancer, cancer recurrence and cancer survival. Clearly, a huge number of ML & DL concepts released over the past decade produce precise outputs regarding particular cancer predictions. Moreover, it is crucial for

the separation of clinical decisions to identify potential problems including experimental design, collecting suitable samples of data and validating classified results. Moreover, despite claims to have contributed to appropriate and efficient decision-making by the ML classification methods, very few have in fact entered clinical practice. Recent advances in omics technology have led us further to better understand a wide range of diseases, but validation results need to be accurate before signatures of gene expression shall be used in hospitals. Only a few marked samples in general. The small amount of data samples is a majorly frequent drawback observed in the research surveyed in this article. The size of training data sets that need

to be large enough is a basic requirement in the use of classification schemes to model a disease. A relatively large dataset makes it possible to divide enough into training and trial sets and therefore to validate the calculators reasonably. A small training sample can result in misclassifications compared with the dimension of the data, while estimators can develop unstable and partial techniques. It's clear that a more wealthy group of patients could predict their survival may improve predictive model capacity. The quality of the dataset and the selection schemes are important for efficient ML and DL and then for precise cancer foretell except for data size. Using feature selection methods to select the maximum informative characteristics subset for training the technique could lead to sturdy models. Reproducible values are also characterized as characteristic sets consisting of histology and pathology studies. Given the lack of static entities, it is essential that a multiple feature sets are adapted to the ML & DL technology over time. We also discovered which SVM and ANN classifiers are commonly utilized for cancer forecasting results as one of the most frequently used ML algorithms [35]. As discussed in our introductory section, ANNs are widely used for nearly 30 years [40]. SVMs are also a newer method to cancer prediction but have already been widely included in their trustworthy predictive results. However, the selection of the best algorithm is dependent on a large number of parameters, which include data types collected, sample size, time limits and the type of prediction results. New methods for overcoming the above-mentioned limitations should be explored regarding the future of cancer modeling. More accurate results and reasoned conclusions would be obtained through efficient quantitative research of the heterogeneous data sages used. Further research on the basis of more public databases, which gather valid cancer data for all diagnosed patients, is needed. Their use by scholars will allow their modeling studies to generate relevant outputs and integrated clinical decisionmaking.

Conclusion

The whole study explains and compares the findings of various machine learning and in-depth learning implemented to cancer prognosis. Specifically, several trends related to those same kinds of machines techniques to be used, the kinds of training data to be incorporated, the kind of endpoint forecasts to be made, sorts of cancers being investigated, and the overall performance of cancer prediction or outcome methods have been identified. While the ANNs are common, it is clear that a broader variety of alternative learning approaches is also used to predict at least three different cancer types. ANNs continue to be prevalent. Furthermore, it is clear that machine

training methods typically increase the efficiency or predictable accuracy of most pronostics, in particular when matched with conventional statistical or expert systems. Although most researches are usually excellently-designed and fairly validated, more focus is quite desirable for the planning and implementation of experiments, in particular with regard to quantity and quality of biological data. Improving the experimental design and the biological validation of several device classification systems would undoubtedly increase the general Quality, replicability and reproductivity of many systems. In total, we believe that the usage of the devices education & deep learning classificatory will probably be quite common in many clinical and hospital settings if the quality of study continues to improve. The assimilation of multifaceted heterogeneous data, which can offer a promising tool for cancer infection and foresee the disease, the researchers evaluated only one form of ANN algorithm. Because of the type and the amount of data used, another machine learning technique can well have exceeded their ANN model.

3. Research gap

Lung cancer is the second largest human illness, which refers to deaths from cancer worldwide. The average survival rate of 5 years for patients with lung cancer in other organs such as the breast, cervix, bladder, prostate or colon does not exceed 14 percent, which is significantly less than the rate of patients with cancer [18]. Thus, early prediction of lung cancer is very important for the appropriate treatments for decreasing the deaths. In big data, healthcare is one of the significant sources. Accurate examination of healthcare information is mostly demanded for detecting lung cancer in an early stage. Multiple researches are being designing newly to recognize lung cancer with more quality using big data. As there is a necessity to classification approach for improving the detection accuracy with respect to time. In addition, machine learning techniques are modelled for enhancing the detection accuracy in big data. Specifically, lung cancer is not well known that means which kind of approaches will give high detection data and which data attributes must be employed for the detection purpose. With the help of huge datasets, prediction methods for breast cancer survivability were introduced by implementing two famous data mining techniques such as Artificial Neural Network (ANN) and Decision Tree (DT) and also utilized a common statistical approach. For measuring the unbiased assessment of three detection models, ten-fold cross validation mechanisms were used for the performance comparison. The outcomes

have proven that DT was the well performing classifier for predicting the disease with an accuracy of 93.6% on the holdout sample; ANN was standing the second best position with an accuracy of 91.2%. Similarly, logistic regression has attained the accuracy of 89.2%. A research was done in [20] for developing detection techniques to know the survivability of prostate cancer, using SVM along with that three methods that were mentioned earlier. Here, the outputs have revealed that the singled out SVM acquired high accuracy, next to that ANN and DT attained more accuracy. Moreover, prostate cancer survivability was examined by ANNs, DTs, and LR methods in [21]. Multiple methods were contrasted in [22] by SEER colon cancer patient dataset for predicting survival rate, and recognized that NNs were best for predicting the survival rate. Ensemble voting of three outperformed classifiers present in [23] was resulted in optimal prediction, and AU-ROC curve to colon cancer survival rate. In some researches, the survival of lung cancer patient was examined by evaluating the SEER database using machine learning algorithms, consisting of SVM, LR [24], unsupervised approaches [25], and clustering-based techniques [26]. In [27], data classification approaches were assessed for finding the chances of patients with definite indications for the growth of lung cancer. The performance of DT and NB classifiers were compared in [28], and implemented for lung cancer data acquired from SEER database. This attained approximately 90% precision in detecting the survival of patient. Ensemble voting of five DTs and meta-classifiers existing in [29] was resolute for acquiring the best prediction survival rate of lung cancer regarding precision and AU-ROC curve. Many challenges related to the machine learning algorithms are associated with manual training. The significant thing is complexity in accurate recognition of nature for pre- processing them correspondingly before subjected to machine learning algorithms. The time and the experts linked with this job were majorly high. According to the research, it was manifested that there is lack of consistency in detection accuracy of machine learning techniques over classical prediction techniques. With the present literature, this was made reliable. Many investigations that compared the machine learning models with classic statistical model have been confirmed that their outcomes were different. Even though multiple strategies were utilized for predicting different types of diseases, the predictive models using the machine learning algorithms reported in the literal works are less for lung cancer detection with IoT integration. There is a high scope to implement more well- performing deep learning models that might produce best prediction outcome. Moreover, the enlarged availability of adequate historical data of patients has paved the way for the development of novel deep learning algorithms for lung cancer prediction. In addition, the optimization algorithms have the ability to improvise the deep learning models. However, there are few disadvantages such as it is not able to find the optimal solution to the problem defined, and it is complex to select

parameters. Moreover, the benefit of PSO is its ability to solve is not applicable. Some of the positives of SMO are useful for solve also demonstrates the incorporation in the application of different analytical and classification methods. In future, by using the proposed framework, we would like to use other state of the art machine learning algorithms and extraction methods to allow more intensive comparative analysis. Declaration of Competing Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in quadratic problems that occurs in the training of SVM, and also it reduces the memory storage.

2.3 Prediction of Lungs Cancer using Machine Learning

There is lot of progress made in the field of treatment of lung cancer in the last years (adjuvant chemotherapy, radio therapy, individualized therapy). Nonetheless, lung cancer is still remained the threat of society and cause of death of thousands of people in all over the world. This paper is all about detection of lungs cancer. Here Computer Tomography (CT) images are used to detect lungs cancer. There are several algorithms are used to detect Lungs cancer accurately. Here unsharp masking filter is use to filtering the image. Adaptive Canny edge detection algorithm is used to detect the edges and cancer affected areas. Neural network is used to classify the features and predict the probability of lung cancer. K-Nearest Neighbors is used to segment the cancer from lungs. Finally achieved the classification accuracy near about 99.5% by using Bayesian Regularization Neural Network (BRNN) and performance measure by Mean square error (MSE) noted as 0.0166.

Keywords: Lungs Cancer, K-Nearest Neighbors, Bayesian Regularization Neural Network

1 Introduction

Another name of lung cancer is carcinoma. It's actually a malignant tumor which is characterized by the uncontrolled growth of cell tissue. Lung cancer happened mainly because of tobacco smoking for a long period. It's a fact that 85% of lung cancer happened because of tobacco smoking. But it's a fact that 10%-15% of lung cancer

affected people are never smoke in their life. Actually, it's happened because of passive smoking, air pollution, asbestos, radon gas [1]. Doctors can able to examine the interior body parts by the help of several testing process. Lung cancer is the most dangerous cause for man and women both. It's not an easy task to detect lung cancer in earlier stages. But there are some major facts that indicate that if we can detect it on earlier stage then it can help to reduce the mortality rate. Approximate 80% of lung cancer affected people are diagnosis correctly at the middle or advance stage. There is an estimation that 1.2 million of people every diagnosis with this disease which is 12.3% of total number of cancers diagnosed and approximate 1.1 million peoples are dying because of this disease which is 17.8% of total number of cancer death [3]. Image processing is an effective way to detect lung cancer easily because there are several useful tools which help in detection. In this proposed work multiple filtering and segmentation are used to detect lung cancer on earlier stage and analyze the data more accurately.

2 Literature survey

Prediction and detection of cancer in its early stages is main target of all researchers. For this they always try to develop a system which helps to predict and detect cancer in its primary stage. Different algorithms are used to improve the accuracy of the Early Prediction and Detection system by preprocessing, segmentation feature extraction and classification techniques. K. Senthil Kumar, Venkatalakshmi and K. Karthikeyan proposed in their work that cancer is a malignant tumor. Several algorithms like K-means clustering, Particle Swarm Optimization (PSO), InertiaWeighted, Particle Swarm Optimization (IWPSO) and Guaranteed Convergence Particle Swarm Optimization(GCPSO) also used for the experiment to compare the accuracy between them on different images. But here Adaptive median filter and GCPSO gives the highest accuracy which is 95.8079% [1]. A. Asuntha, A. Brindha, S.Indirani, Andy Srinivasan proposed in their work that there is multiple choice for choosing images which may be a CT,MRI or Ultrasound image. Gabor filter, Layer separation technique, Super pixel segmentation technique is use to separate the object or the region of interest from the other part of image. PSO, Genetic Algorithm and SVM are used to get a better result. Here accuracy is about 89.5% with reduction in false positive [2]. Babita Rani, Ashok Kumar Goel, Ravneet Kaur proposed in their work that modified approach of some algorithm which can achieve the better result. In this paper hybridization of three techniques are utilized that is Bacterial Forging Optimization (BFO), Principle Component Analysis (PCA) and Back Propagation Neural Network (BPNN). Here accuracy is 98.1881% [3]. Manasee Kurkure, Anuradha Thakare used CT images for the experiment. They used canny edge detection algorithm where canny filter used in feature extraction and output value output value gives to naïve bayes classification

which is optimized by Genetic Candidate Group Search (GCGS) algorithm. Genetic Algorithm (GA) used to identify genes that help to classify patient lung cancer status with a notable predictive performance [4]. Neha Panpaliya, Neha Tadas, Surabhi Bobade, Rewti Aglawe, Akshay Gudadhe proposed a system where they use Histogram Equalization and neural network classifier to check the state of patient whether it is normal or abnormal. This will generate very accurate result for detection and prediction of lung cancer. So, early detection and prediction of lung cancer should play a important role in the diagnosis process and also improve the survival rate of patient [5]. Fatma Taher, Naoufel Werghi, Hussain Al-Ahmad, Rachid Sammouda had proposed the early detection of lung cancer is a challenging problem, due to the structure of the cancer cells, where most of the cells are overlapped with each other [6]. This paper presents KNN algorithm for segmentation and BRNN for classification.

3 Proposed Model for Lungs Cancer Detection

To make an image sharp “unsharp mask” is used, though the name means opposite. Sharpening used to emphasize detail and texture, also it applied at the time of post processing of digital image. It is a technique for image sharpen which is more advantageable for the segmentation process. Unsharp masking used several purposes of an image preprocessing where we used for the blur image convert to the original sharpen or smooth image. We compute the equation for unsharp masking,

Many more researchers are used different thresholding algorithm for the calculate the image background and foreground color intensity level. We were used Otsu’s thresholding because of some advantage make it flexible. Basically, Otsu’s thresholding method compute the three parameters which are weight, mean, variance of background and foreground of each level. It calculates the threshold with in class variance and end of the calculation we add two variances after that result multiplied their corresponding weight. Otsu’s method compute with in class variance to the following equation, Where w_b is the weight and σ_b^2 is the variance of the background pixel. For the foreground pixel is represent the weight and σ_f^2 is represent the variance.

3.3 Adaptive Canny

In this section we were propounded the adaptive canny algorithm for detection of cancer region or find the actual effected area. Generally, edge map is little bit of necessary for the proper results, some techniques are failed to find the proper edge. In this paper we used adaptive canny which successfully perform of our lungs cancer detection model. Adaptive canny compute through Otsu’s global thresholding techniques.

3.4 Propounded K-Nearest Neighbor for Segmentation

This section we proposed the KNN algorithm which successfully perform by the segmentation process and achieve the better segmentation result. In pattern recognition KNN needs to predicted value of K-based and training data for computation. It has some powerful application, such as pattern recognition, machine learning, text categorization, data mining, object recognition etc. In this segmentation process the KNN used to predict the model as well as used for predict the segmented result.

3.5 Bayesian Regularization

Here the neural network has become independent, using BR neural network we get the classifications results. Using BRNN we compute the classification process where BRNN is the most outstanding learning ability. NN provide some specific parameter which necessary to classify dataset. It contains layers and neurons, which get some inputs and generate corresponding output. this out passed through the layer into next layer of neurons and finally BRNN compute the specific fitting model which has less loss over the results. BRNN is predictive learning ability which is helps to get the better accuracy almost than some others method. It also handles gradient based optimization problem and measurement the performance using MSE which value noted as 0.0166. finally, we achieve the classification accuracy which near about 99.5%.

4 Result & Discussion

Fig 2: Using proposed algorithms detect lungs cancer (result). Here all the figures show the different step of processing. JPG format 512 X 512-pixel image of lung be taken as for the experiment. Here lung image is sharpening for the easier edge detection. Otsu global thresholding is used for set the threshold value of canny edge detection where proposed edge detection algorithm can easily detect the edges from the binary image. But there are lots of small pixels and lung edges also detected which is not required to show the result. So, these edges are removed and the affected area is filled using whole filling algorithm.

4.1 Evaluate the Performance

Evaluate based performance show the result of graphical form which include the training, testing and validation-based performance .BR algorithm is including probabilistic distribution, sum of squares of network weight and gradient based optimization. fig 3A shown as the training and testing state, fig 3B shown as the error histogram. Fig 3C and 3D shown as the gradient based optimization of the given data and the training, testing and target set visualization.

2.4 Lung Cancer Detection using Deep Learning

Abstract

Pulmonary cancer also known as lung carcinoma is the leading cause for cancer-related death in the world.

Early stage cancer detection using computed tomography (CT) could save hundreds of thousands

of lives every year. However analysing hundreds of thousands of these scans are an enormous burden for

radiologists and too often they suffer from observer fatigue which can reduce their performance. Therefore,

a need to read, detect and provide an evaluation of CT scans efficiently exists.

In this paper, the author proposes a method of detecting lung cancer in a CT scan using a 2D-UNet

model on a web application. The author cropped 2D cancer masks on its reference image using the center

of the lung cancer given in the dataset and trained a model with different techniques and hyperparameters.

Introduction

1.1

Introduction

This section aims to give a brief overview into the project, the goals of the project, the challenges faced and

the structure of the report.

1.2

Project Overview

1.2.1

Lung Cancer Detection Application

A web application has been developed to demonstrate a proof of concept. The application requires a user

to upload a CT Scan. The application then processes the file and displays the images to the user. The user

then chooses which scan he or she wants to predict then the application pre-processes the CT Scan and infers

the image to the predictive model. The output of the model is then displayed to the user.

The user has the

choice to view the images via a carousel or a gallery mode.

1.2.2

Research Element

This report contains many aspects of research that support deep learning's ability to find lung cancer within CT Scans. The data used to train the model was gathered from a deep learning competition called LUNA 2016 which are CT Scans that have been annotated by radiologists. The radiologists determined whether a nodule is lung cancer or not and this location has been specified in 3D coordinates. Deep Learning research has also been conducted to ensure that the correct architecture would be used to train the model. The architecture used is called U-Net. It acts as a deep learning segmenter of images. Using our evaluation methods we could see that the deep learning architecture is able to segment regions of interest relative to its accuracy but found that the model creates many false positives which means that the model required more regularization and training time, although it was trained for 40 hours.

121.3

Project Goals

The completed project should accomplish the following Objectives:

- **Self Education on Deep Learning** A large part of this project contains a lot of self education, initially the author knew very little about deep learning, as part of this project the author should have a good grasp on deep learning concepts.
- **User Research and Evaluation** The project should have a user-centred design aspect. This means that the system should be designed to help certain users. In this project the user would be medical professionals who work in diagnosing lung cancer.
- **System: Upload CT Scans** The system should be capable of getting CT Scans from Users that will be utilized by the Deep Learning Model.
- **System: Detection of Lung Cancer** The system should be able to detect the lung cancer within the CT scan images that users have uploaded.
- **System: Display Results** The system should be able to give information that our user can appropriately understand and gain insight from it.

1.4

Project Challenges

This section discusses the challenges that were overcome to complete this project. Each of the challenges are briefly introduced and detailed in later sections of the report. This section can be useful for those who aim to do a deep learning project with large image datasets.

1.4.1

Large 3D Dataset Management and Preprocessing

Working with the dataset is the most difficult challenge that had to be overcome for this project. The dataset is very large (around 70 Gigabytes) which makes managing and analysing the dataset and training the model very computationally heavy. [Figure 1.1](#) shows one instance of a patient's CT Scan which is approximately 60 Megabytes. A potential solution to this is to migrate the computation workload to a cloud service such as Google Cloud, AWS or Floydhub which could result in a more efficient workflow for the project. A single CT scan is also 3 Dimensional which can be complex to work with especially during feature selection and data preparation.

1.4.2

Neural Network Architecture Design Choices

Neural networks are architectures that are designed in of itself and there are many types of architectures out there that exist to solve different problems. A key challenge of the author is to use an architecture that is able to find malignant tumour patterns in the data. To overcome this, the author has to undertake necessary research to implement the correct model design prior to training. There are many neural network architectures such as multi-layer perceptrons, convolutional networks and sequence models.

1.4.3

Understanding Deep Learning Concepts

Deep learning is an exciting but new concept for the author. To overcome this challenge the author has performed research and study to deliver the project goals. A useful resource that has been used is Stanford's professor Andrew Ng's deep learning course on Coursera. This course enables the

learners to gain a foundation to deep learning.

1.4.4

Training the Model

Deep learning models require hours of training time, the best performing model that has been trained in this project required 40 hours of training time on a Tesla K80 GPU available on Floydhub. Compared to standard machine learning algorithms, performing experiments and getting results take much longer. This problem can stop the project's progress. Careful planning before running experiments can minimize this problem.

Using `keras.Callback` features such as `CSVLogger`, `ModelCheckpoint`, `LearningRateScheduler` and etc can be used to debug the models quicker when models don't improve which directly increase productivity

Deep Learning Integration

Integrating deep learning models into applications using Python is different compared to standard machine learning algorithms. Standard machine learning algorithms can be serialized into a file and loaded into an application. Deep learning models use Tensorflow as a framework. Integrating the model requires additional code to ensure that the model runs sequentially on the same thread as the application. This is further explained in the implementation chapter.

1.5

Report Structure

The research chapter details research about lung cancer, impact statistics, existing solutions, challenges that medical professionals face, technologies used and the dataset for the project. The main aim of this chapter is to introduce how lung cancer is an incredibly complex problem and that the current ways its being diagnosed.

The deep learning chapter deals with the Deep Learning intuitions needed to develop and train the deep learning model. This chapter introduces deep learning concepts including the research papers associated with the concepts to help debug neural network issues and identify problems during training.

The approach and methodology chapter deals with standard practice used to deliver the project.

The two mentioned is Agile Software Development and CRISP-DM. The idea of this chapter is to ensure

that the project is managed using standard industry practices.

The design chapter deals with the design methodologies used, user centered design arte facts such as

persona, scenarios, design wireframes and technical design arte facts such as use case diagrams and system

architecture designs. The goal of this chapter is to demonstrate what kind of user would be using this

application and how it is specifically designed for that type of user.

The implementation chapter details the process of creating the project, following the approach and

methodology, adhering to the designs created and performing deep learning experiments drawing from

relevant research aforementioned.

The project plan chapter outlines how the project has evolved since the interim throughout the entire

development process and how the tasks were estimated.

14The conclusion chapter contains results gained, a proof of concept evaluation, future and final thoughts

about the project overall.

1.6

Conclusion

The project integrates different topics in Computer Science to try and solve a real world problem in the

medical domain. The project consists of data mining and software development to deliver a proof of concept.

The application is a lung cancer detection system to help doctors make better and informed decisions when

diagnosing lung cancer.

In the next chapter, the author outlines the relevant research for the project.

Deep Learning

3.1

Overview

According to Andrew Ng[34],The theory of Deep Learning has been around for decades but it has only been

in the past few years that it has risen and taken off. In our society we have gone from having a relatively

small amount of data to huge amounts to the point that we don't know what to do with them

According to Huang et al[26], Deep Learning has gained large popularity within different fields of interest

ranging from a theoretical research in academia to practical application in industry.

Newly emerging

algorithms have expanded applications of Learning Machines to be able to compute large amounts of data

with the help of hardware implementations and optimized parallel computational techniques available today.

With this Deep Learning Architectures are able to be generalized, customized and configured to advance

efficient and accurate models.

Compared to other techniques such as standard Machine Learning algorithms, Deep Learning is a

preferred option as performance increases as the data scales[34]. Machine Learning algorithms such as

Support Vector Machines, Random Forest and etc, performance plateaus when data increases in these

algorithms. See [Figure 3.1](#). Deep Learning Neural Networks works well for my lung cancer detection project

as our dataset is large and can iteratively grow larger as more people infer new data to it which will result

in a boost in performance.

In this chapter, the author explores the intuitions made into creation deep learning models.

Finally the result is evaluated using a dice coefficient and confusion matrix metrics. The author reaches a

65.7% accuracy on the dice coefficient and an average 0.88% true positive rate and 0.71% false positive rate

on a test set of positive and negative samples.

Neural Network

A neural network is a model that has been inspired by the brain, the brain consists of nearly 10 billion

neurons with 60 trillion connections between each other. A neuron consists of a cell body called the soma

where the nucleus is found, many dendrites where input signals are received and transmitted and a synapse

which is basically connections between neuron to neuron.[13]

Artificial Neuron

Like a brain neuron, neural networks consists of an artificial neuron. An artificial neuron consists of a weight

to determine the strength of a connection, a linear function that needs to be computed and an activation

function that computes the weighted sum of the linear function that is then compared to a threshold value.

It is also worthy to note that artificial neurons like [Figure 3.3](#) can be modified to contain different linear

function algorithms such as Gradient Descent, Logistic Regression, etc and also different activation functions

such as ReLU (Rectified Linear Unit) or TanH functions which depends on the problem to solve.

3.2.2

Artificial Neural Network

An Artificial Neural Network is an interconnected architecture where there exists an input layer where input

data is placed, a hidden layer(s) where artificial neurons are stacked on on top of each other and an output

layer where the prediction or classification is made.

3.2.3

Forward Propagation

Forward propagation is technique in which data moves through from the corresponding input layer, hidden

layers and output layer sequentially.

Figure 3.4: An Artificial Neural Network also known as a Multi Layer Perceptron

A hidden layer usually consists of an weight which is used in an optimization algorithm such as Gradient

Descent, an activation function such as Sigmoid, TanH, LeakyReLU etc and a Loss Function to calculate the

loss or error of the function which we use to back propagate to adjust the weights.

3.2.4

Back Propagation

Back propagation is the opposite of Forward propagation as it feeds the network backwards. Back propagation

is used to adjust the weight of the neural network after the errors have been computed by forward propagation

algorithms. In later sections of this chapter I detail how Gradient Descent is used in back propagation to

adjust the weights of the neural network.

3.2.5

Activation Functions

Activation functions are an important part of a neural network. They allow neural networks to create non linear functions to solve problems. The 3 most used activation functions are Sigmoid, TanH and ReLU.

Activation functions allow or stop neurons from firing into the next layer by comparing it into the activation function thresholds.

Activation functions are used both in the forward and backward propagation where in the forward propagation

an activation function is used to calculate the loss where the output of a function is compared to the a real

number and in backward propagation they are used to update the parameters of the neural network.

.

Convolutional Neural Networks

3.3.1

Overview

Convolutional Neural Networks is all about using Deep Learning with Computer Vision. A good way to gain

intuition about this is to think about a Neural Network Architecture and how it is applied to visual tasks

i.e. Images and Video. Convolutional Neural Networks have allowed us to create Facial Recognition, Object

Architecture

Like a Neural Network, a typical Convolutional Neural Network consists of a multiple hidden layers called

a Convolutional Layer where the linear function computes the strided convolutions over an image to extract

features. It also consists of a pooling layer that computes another function such as Max Pool or Average

Pool to reduce the size of the image in the neuron to speed up the computation. It does it by extracting the

features of the neuron image and ignoring the rest, this makes the network more robust.

There is also fully

connected layer which is like a hidden layer in a neural network where the sum of the outputs of each layer

are flattened and where each value is an input to the next layer followed by an activation function

Convolutional Layer

In a Convolutional Neural Network the linear function that is used is called a convolutional layer. Each node in the hidden layer extracts different features by using image processing feature detectors. For example, in the first layer, the first node may extract the horizontal edges of an image, the second node may extract vertical edges and etc. These features are extracted using a kernel. [Figure 3.11](#) shows an example of how strided convolutions work on an image. The bottom is the original image and the top is the output of the convolutions. It is also worth noting that the output of the convolutions reduce the dimension of the original image

Pooling Layer

The pooling layer happens tends to be computed after the convolutional layer. The reason why pooling is done is to further reduce the dimensions of the convolutional layer and just extract out the features to make the model more robust. There are two types of pooling done: max pooling and average pooling. Max pooling extracts out the highest pixel value out of a feature while average pooling calculates the average pixel value that has to be extracted.

Architecture

Ronneberger et al 2015,[\[36\]](#) mentions that a U-Net model consists of a contracting path (left side). This is the same as standard Convolutional Networks outlined in the last section(3x3 Conv with ReLU and 2x2 Max Pooling). As earlier discussed, the great idea about this is that the network is able to create features necessary to detect patterns that we are looking to predict(In this case its lung cancer). The problem with this on its own is that convolutional layers down sample the size of the image. Hence data is lost as it feed forwards. The unique design of the U-Net model lies in its expanding path (right side) which consists of up-convolutions (size 2x2) and merge layers. The up-convolutions take a downsampled sized image and expands the borders to increase the size of the image. This is done before the merge happens, the merge

layers then takes output from an earlier convolution layer and concatenate it with the previous upsampled image. This concatenated data or 'Merge' is then fed through another convolution layer. The network is able to send data from an earlier network to a later one through skip connections. This is how the network is able to segment the images after it has found the features. See

Improving Deep Neural Networks

3.5.1

Overview

This section discusses the decisions made into improving the U-Net model. Deep Learning is a highly iterative process and training is extremely slow and can get stuck on plateaus. i.e. the model is not getting better.

I've had this experience many times while training the U-Net for hours and getting bad results. To avoid

this, different techniques have been applied on the model that helped tremendously in training and reaching

better convergence. It's also worth mentioning that there are many ways to improve a neural network and

that it is constantly evolving as new techniques are created. It is completely infeasible to explain each one.

The author has decided to only introduce techniques that are effective and currently state of the art.

3.5.2

Regularization using Dropout

Neurons in a neural network have a high tendency to adapt co-dependency among each other during training

time. This means that certain activations become more heavily weighted and results in a network becoming

over fit to the training set. To stop the network from over fitting Dropout is used.

Weight Initializations

One of the main problems encountered in deep learning is vanishing / exploding gradients. This is when the

activations become smaller or larger at each layer. To summarise briefly, if the weights in a network are too

small, then the signals shrink as it passes through each layer until its too small to be useful conversely if the

weights in the network start too large then the signals becomes too large to be useful.

This also has terrible consequence for the parameter updates on the weights as the network back propagates either giving very small updates or very large into the network. This has terrible consequence for large neural networks as it means the larger the network is the more likely it will be to be slow to train and inaccurate.

Gradient Descent

Gradient descent is the learning algorithm used in Deep Learning. It is mainly used in the back

propagation step in the training of the neural network. In a neural network the cost function measures

the inconsistency between a predicted value and the actual value. The idea behind gradient descent is

that we increase the quality of the weights so that the weights minimize a cost function in the neural network.

Gradient descent takes the loss function and calculates a derivative. This is also known as calculating the

gradient of a loss function. The idea behind calculating the gradient is that it computes the best direction

which the weights must be updated to guarantee a best outcome for minimizing the cost function.

The learning rate parameter then is used to determine the size of the steps we take given the direction. The

closer the network gets to convergence i.e. lowest point in the cost function, the more accurate the neural

network becomes

Variants of Gradient Descent

Over time there have been many variations of gradient descent that have improved the algorithm's capability.

The main ones are batch gradient descent, stochastic gradient descent and mini-batch gradient descent.

Sebastian Ruder wrote a great paper on gradient descent optimization algorithms[[37](#)] and according to Ruder

the trade-off between the algorithms is the accuracy of the parameter update and the time it takes to perform

the update.

3.5.5.1

Batch Gradient Descent

Batch gradient descent computes the gradient of a cost function relating to the parameters for the entire dataset. This basically means that batch gradient descent has to look at the entire dataset before an updating the parameters at a set number of epochs.

An epoch is a pre-defined number of times a neural network will a forward and backward pass during training. It also means predefines how many times the parameters gets updated. 1 epoch = 1 forward pass
= 1 backward pass

Conclusion

These learning algorithms help in training the neural networks however on their own they cannot guarantee good convergence. In the next section I introduce Optimization algorithms that help gradient descent learning algorithms perform better.

3.5.6

Gradient Descent Optimization Algorithms

3.5.6.1

Overview

Although there are different gradient descent algorithms out there with their own advantages and disadvantages, there are still some issues regarding training the network that has to be solved in a different way. In this section I introduce some ways we can optimize our gradient descent algorithm.

According to Ruder, choosing a learning rate can be difficult as statically predefining the learning rate can either mean slow convergence when its too small or fluctuate around the minimum when the learning rate is too large. To solve this problem there needs to be a dynamic way to choose the learning rate.

Choosing the same learning rate also means that an update applies to each parameter, this can be a problem if we don't want to update the weights that find less frequent features in our data. This means that its not a good idea to update weights that find rare occurring features compared to commonly occurring ones.

The last challenge is minimizing non-convex error functions aka saddle points. Bengio

et al[22], mentions that minimizing over saddle points give great difficulty for a neural network to find the global minimum which means learning can be extremely slow and give an impression that a network is stuck on a local minimum. The reason for this is because a saddle point's gradient is extremely small which makes it difficult for the gradient descent algorithm to minimize the cost function. [Figure 3.18](#) shows gradient descent traversing through a saddle point plateau According to Ruder[37], the momentum term increases for dimensions whose gradients point in the same direction and reduces updates for dimensions whose gradients change directions. This means that a neural network can converge faster by reducing oscillations. The default value for momentum is 0.9.

3.5.6.3

RMSprop

RMSProp is an adaptive learning rate technique proposed by Geoffrey Hinton. Ruder mentions that RMSprop divides the learning rate by exponentially decaying average of squared gradients. This means that the learning rate dynamically decays exponentially as the network trains. This means that the network can converge much better towards the global minimum. A good default value for learning rate for RMSprop is 0.001.

3.5.6.4

Adaptive Moment Estimation (ADAM)

Adam is another method that computes adaptive learning rates. This method was invented by Diederik Kingma and Jimmy Ba[1].

Adam is similar to RMSprop and Momentum where it keeps an exponentially decaying average of the squared gradients and exponentially decaying average gradients. Since these two values are 0 when initialized, Adam uses bias correction for the first and second moment estimation to correct the algorithms initial bias towards 0. These two values are then used to update the parameters. According to Kingma et al[1], Adam when used with Dropout, outperforms the other optimization algorithms.

Conclusion

With these intuitions in mind, One would be able to get a better idea of what is going on inside a deep learning model while it's training. There is an element of black-box randomness that happens when training deep learning models but these concepts gives me a better idea of what could be happening when a network doesn't work as intended. In short, it helps to know these concepts when debugging. In the next chapter the author, outlines the designs for the project.

CHAPTER 3

SYSTEM ANALYSIS

3.1 DRAWBACKS OF EXISTING SYSTEMS

In some cases, the application still does not have accurate results. Further optimization is needed.

Priority information is needed for segmentation. Database extension is required for greater accuracy.

Only a few diseases are covered. Therefore, the work must be expanded to cover more diseases.

Possible causes that can cause misclassification can be: Symptoms of the disease vary from cigarettes, optimizing the characteristics needed, more training patterns are needed to cover and predict more cases - the actual disease

3.2 PROPOSED SYSTEM

As the medical images are full of noise and due to the present of noise it becomes very difficult for prediction.

Machine learning is used for classify the tumor whether the tumor is benign or malignant.

The algorithms in context to Lung Cancer Prediction.

Support Vector Machine- SVM help in prediction of cancer by separating the dataset into two classes by using kernel functions as the image data are high dimensional.

As the images are arranged in non linear manner so it will plot the image in 3-D plane by using kernel function like polynomial kernel, Gaussian kernel, radial basis function etc, and separate the class using a hyper plane.

3.3 SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

System : Pentium i3 Processor.

Hard Disk : 500 GB.

Monitor : 15'' LED

Input Devices : Keyboard, Mouse

Ram : 2 GB

SOFTWARE REQUIREMENTS

Operating system: Windows 7.

Coding Language: python

Tool: Pycharm

Libraries: sklearn, nltk

CHAPTER 4

SYSTEM DESIGN

4.1 ARCHITECTURE DIAGRAM

The following algorithm describes the step by step approach for the proposed model.

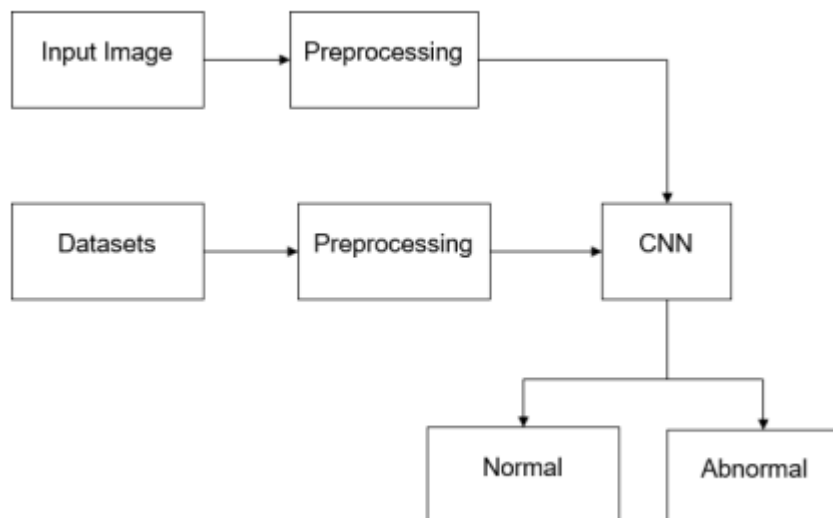


Fig 4.2 SYSTEM ARCHITECTURE

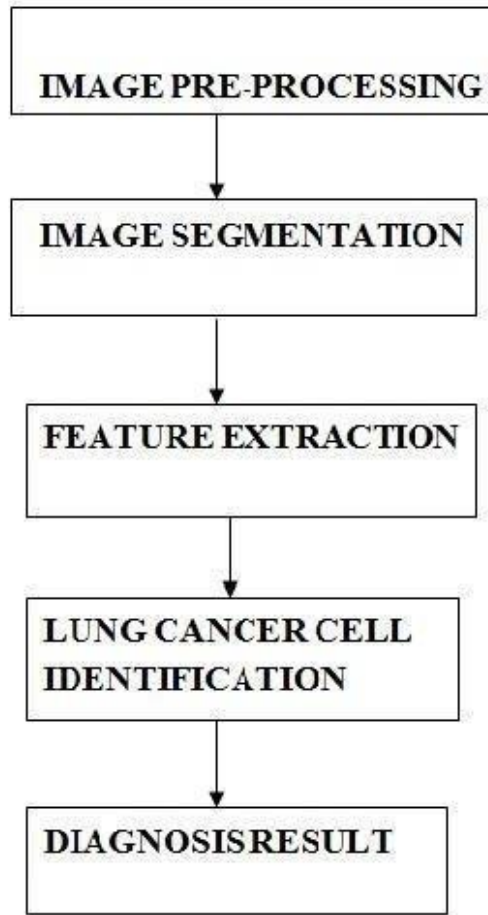


FIG:4.2: WORK FLOW DIAGRAM

4.2 DATA SET

Dataset for training is obtained from Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI).

LIDC and IDRI consist of 1000 CT scans of both large and small tumors saved in Digital Imaging and Communications in Medicine (DICOM) format

4.3 DATA ACQUISITION

- The histopathology images are obtained from LC25000 Lung and colon histopathological image dataset
- .Three classes of benign tissue, Adenocarcinoma, and squamous carcinoma cells of lungs with 5000 histopathology images in each category, are considered for our work.

4.4 DATA FORMATTING

- The obtained dataset was RGB color histopathology images with .jpeg format.
- The images were resized to maintain a uniform aspect ratio of one with (180, 180) pixel size for the CNN operation.
- All the pixel values for the images were converted in range of (0, 1) to make convergence faster. We have implemented the image acquisition technique like horizontal and vertical flip and zooming to increase the image number and variation in the data pattern.
- The neural network tends to over-fit in case of a limited number of training data samples trained with a higher number of epochs show Adenocarcinoma's histopathology image and its augmented images, respectively, with the horizontal and vertical flip and a zoom range of 0.2 applied.

4.5 MODEL TRAINING, TESTING, AND PREDICTION

- A liner stack of layers was used to create the Convolutional Neural Network (CNNs or ConvNets) for the image classification and recognition.
- Training and testing images were passed through convolutional layers with kernel filters, max pooling, and fully connected layers.
- The softmax function was applied to classify the given object.
- The model was trained and tested using Google Colaboratory GPU named as a device: GPU: 0.

4.6 TRAINING

- Back-propagation algorithms used to train the Deep CNN to detect lung tumors in CT image of size $5 \times 20 \times 20$. It consists of two phases. In the first phase, a CNN consists of multiple volumetric convolution, rectified linear units (ReLU) and max pooling layers is used to extract valuable volumetric features from input data.
- The second phase is the classifier. It has multiple FC and threshold layers, followed by a SoftMax layer to perform the high-level reasoning of the neural network.
- No scaling was applied to the CT images of the dataset to preserve the original values of the DICOM images as much as possible.
- During training, the random sub-volumes extracted from the CT images of the training set and are normalized according to an estimate of the normal distribution of the voxel values in the dataset.

4.7 ALGORITHMS AND METHODS

CNN ALGORITHM

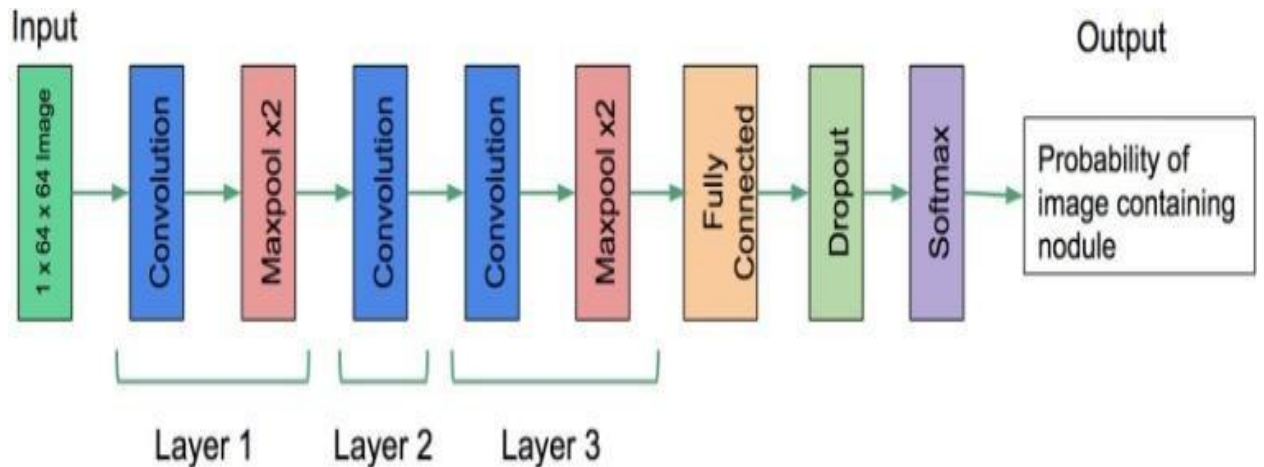


FIG:4.7: CNN FLOW CHART

CNN is influenced by the biological process, where the animal optical cortex system representation is used to display the connectivity pattern between the neuron elements.

This algorithm is appropriate for a classifying multi-class problem and binary classification (for example — predicting whether a medical image has a malignant tumor or not).

It comprises of a convolutional layer, activation, pooling, and output layer. The CNN changes the pixel values of input CT image through weighing in convolutional layers and sub-sampling layers are sampled alternatively.

The outcome of this algorithm is a recursive function with weighted input values.

CNN in the medical area is employed auspiciously to aid disease diagnosis like breast cancer, lung cancer, cardiovascular diseases, neurological disorders, etc

CNN to change the image pixel values into the pixel categories.

The main variation among FCN's and CNN is that deconvolutional and upsampling layers in CNN are replaced with a fully connected layer in FCN.

It generates a score map that is similar to the size of the input image for each class and it also categorizes every pixel value of the image.

To enhance the prediction accuracy this algorithm merges the preceding layer results along with end layers deep convolution and upsampling layer results

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 MODULES

5.1.1 IMAGE PRE-PROCESSING

image pre processing :Image preprocessing are the steps taken to format images before they are used by model training and inference. This includes, but is not limited to, resizing, orienting, and color corrections.

Image augmentation are manipulations applied to images to create different versions of similar content in order to expose the model to a wider array of training examples. For example, randomly altering rotation, brightness, or scale of an input image requires that a model consider what an image subject looks like in a variety of situations

5.1.2 IMAGE SEGMENTATION

image segmentation : image segmentation is the process of partitioning a digital image into multiple image segments, also known as image regions or image objects (sets of pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.[1][2] Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property[3] , such as color, intensity, or texture.

5.1.3 FEATURE EXTRACTION

Feature Extraction: Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups.

So when you want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables.

These variables require a lot of computing resources to process. So Feature extraction helps to get the best feature from those big data sets by selecting and combining variables into features, thus, effectively reducing the amount of data.

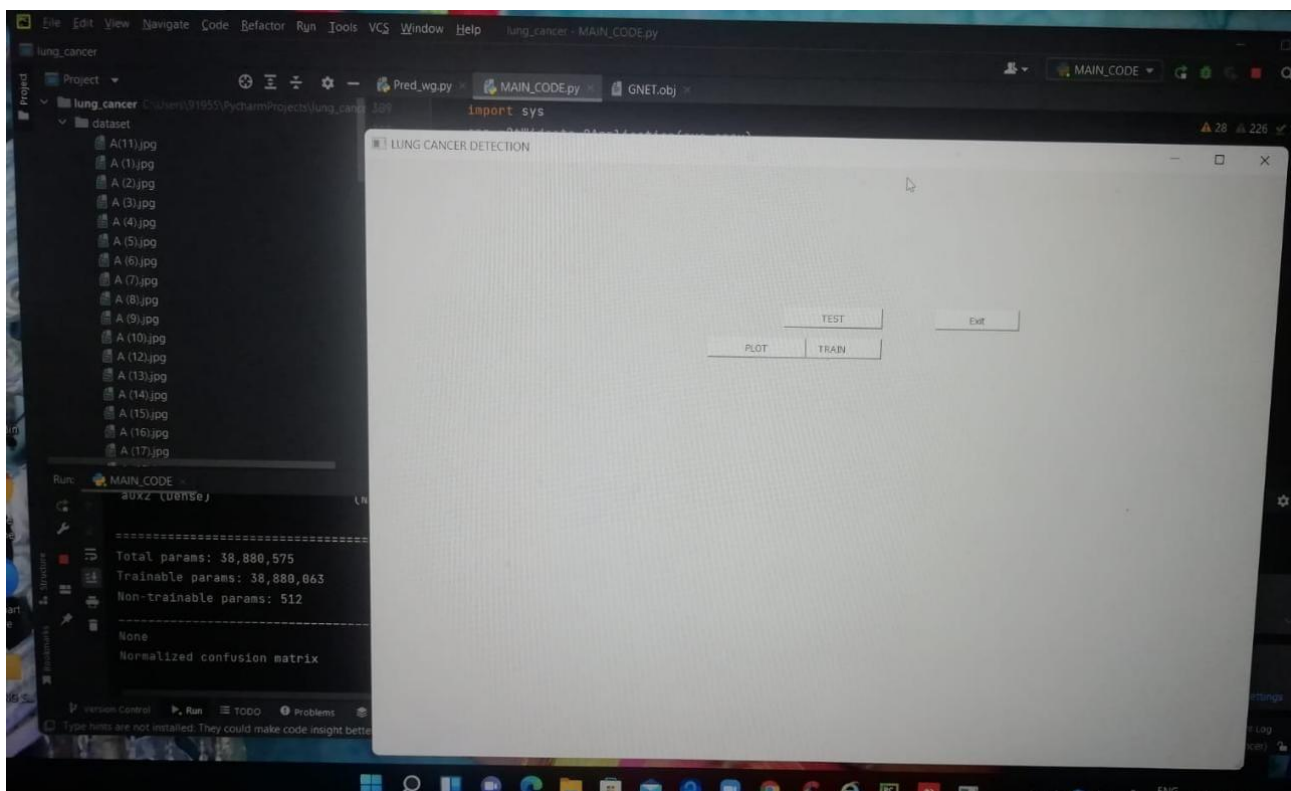
These features are easy to process, but still able to describe the actual data set with accuracy and originality.

Image Processing –Image processing is one of the best and most interesting domain. In this domain basically you will start playing with your images in order to understand

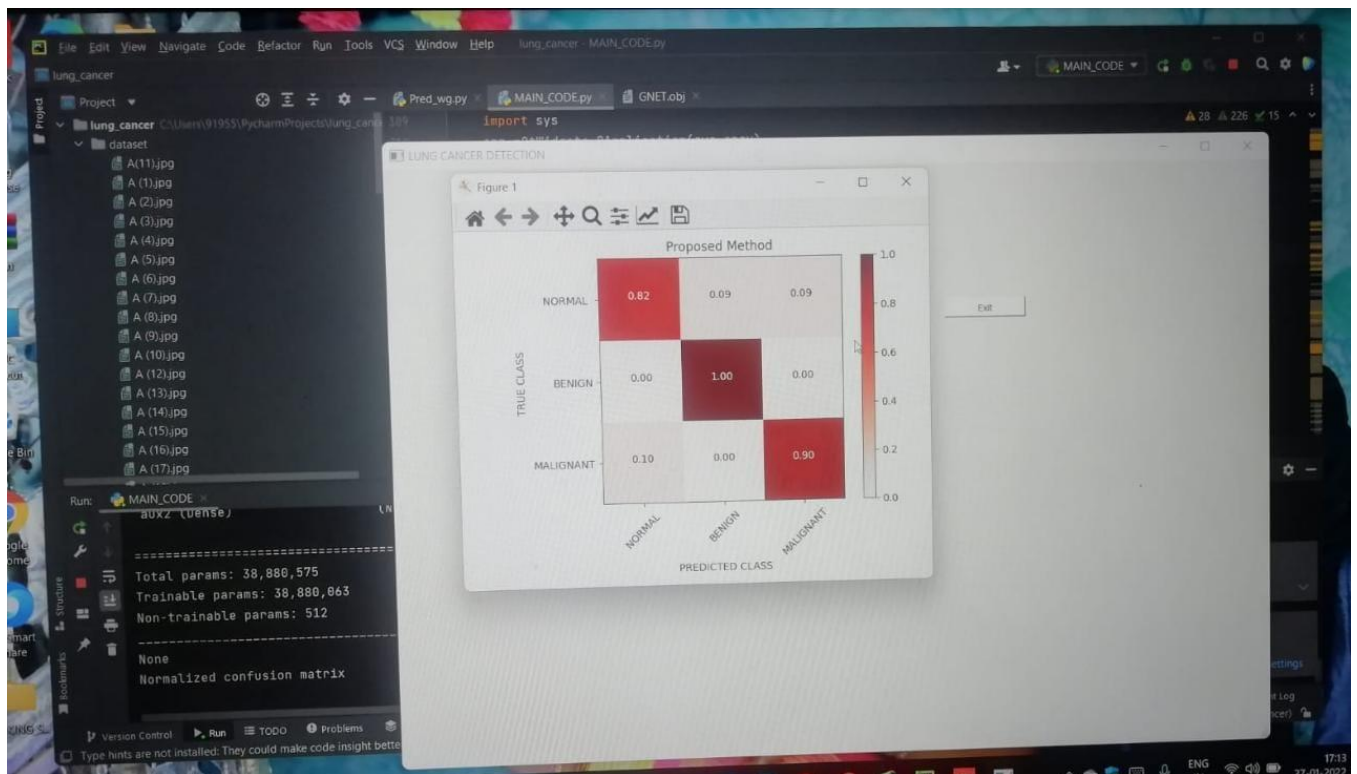
5.2 RESULTS

From the institutional database, a cohort of 586 patients was selected by applying the abovementioned criteria. Patients with distant metastases or histology different from adenocarcinoma and squamous cell carcinoma were excluded from the present analysis. ,therefore, 472 patients were included in the study (T1-T2 353 patients, T3-T4 119 patients). Staging was clinical and pathological in 97 and 375 of cases, respectively. Subsequently, the patients achieved an accuracy of 69%, a recall of 70%, and a specificity of 67% in the test set, for the identification of T1-T2 and T3-T4 lung cancer, in the final model analysis. ,eAUC was 0.83, 0.73, and 0.68 in the training, validation and test sets, respectively. Results of all metrics for the final model in the training, validation, and test sets are

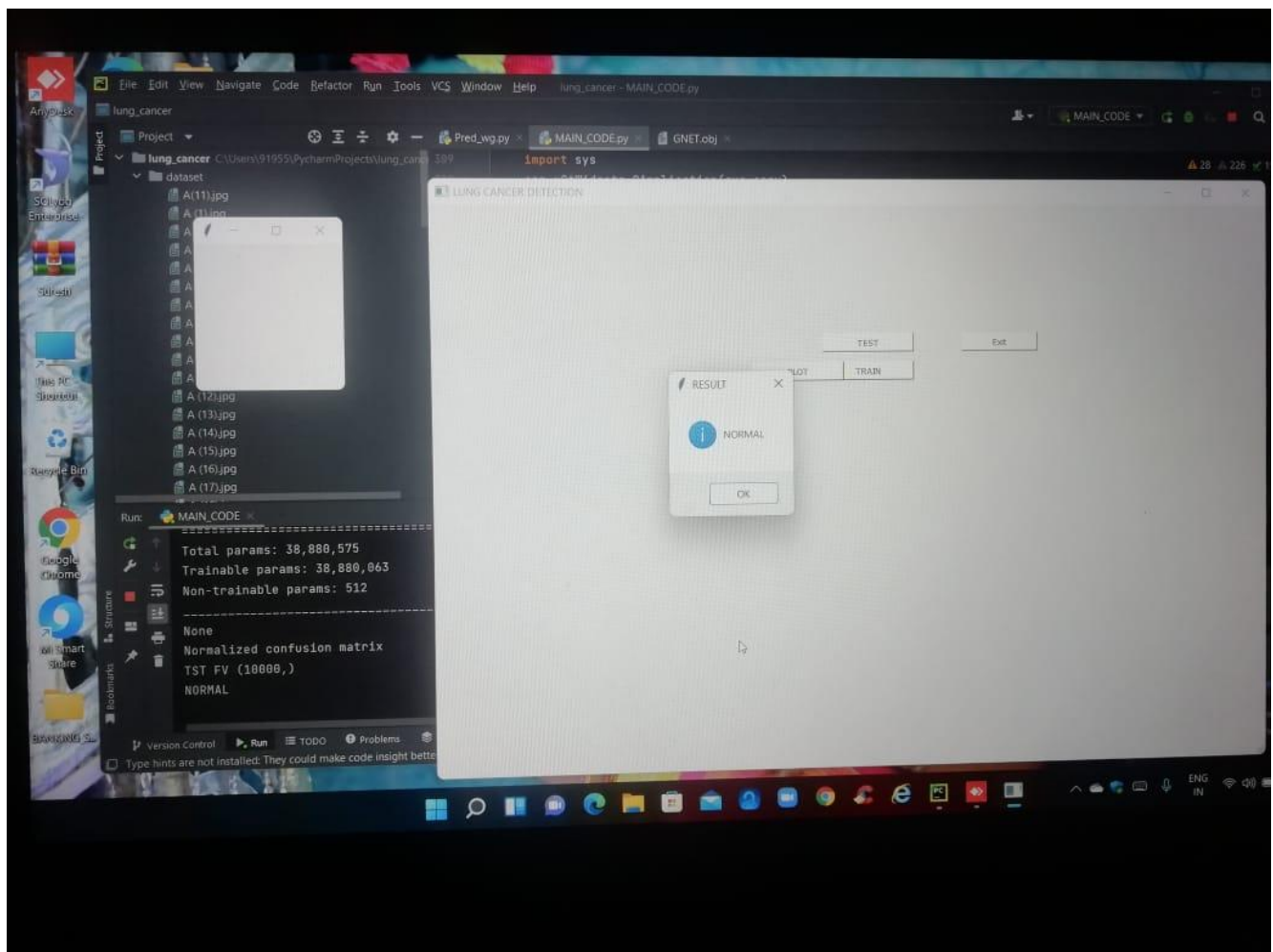
OUTPUT SCREENSHOTS



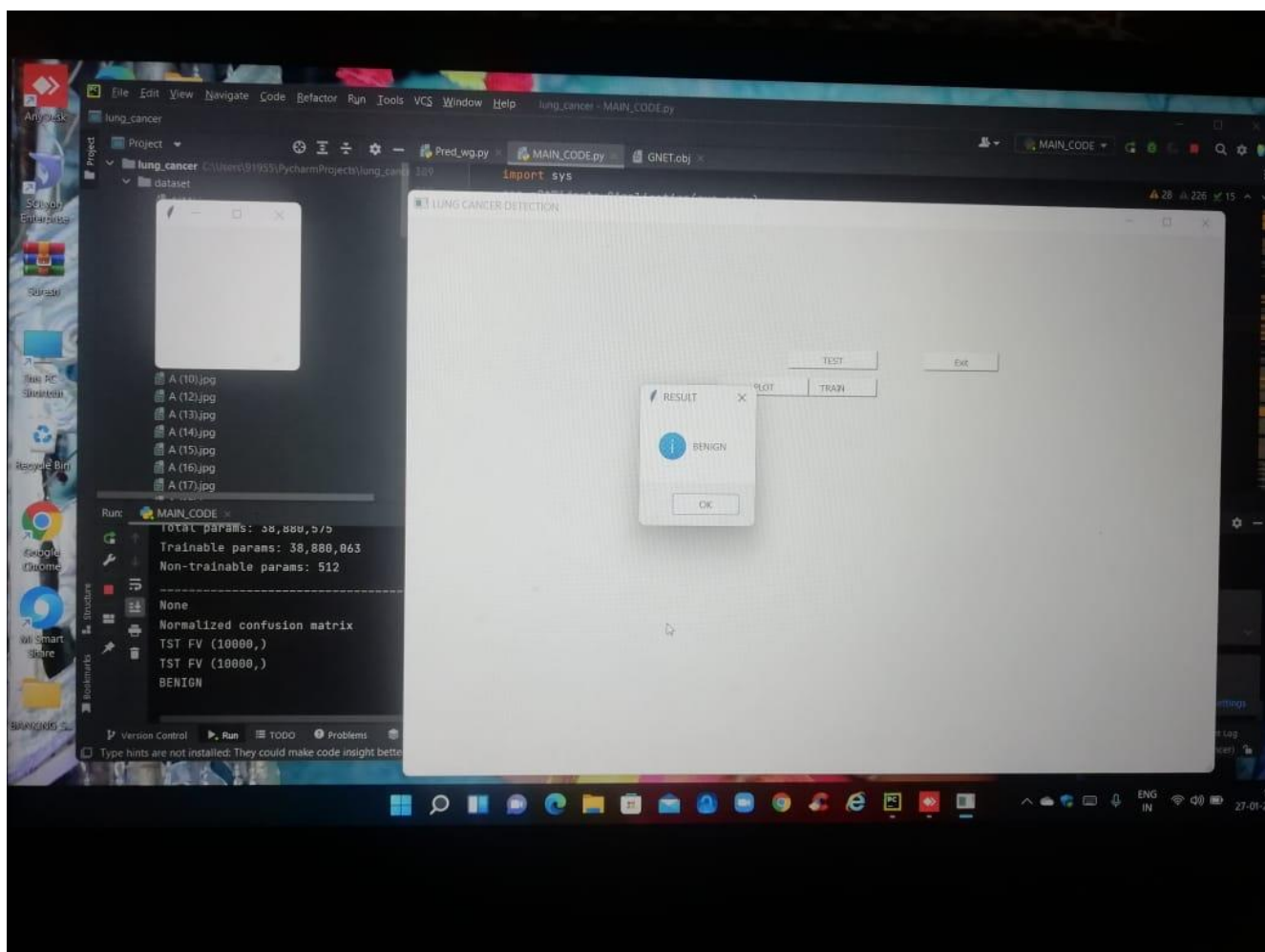
The above picture will be displayed after the execution of code it contains all the process that we have to do



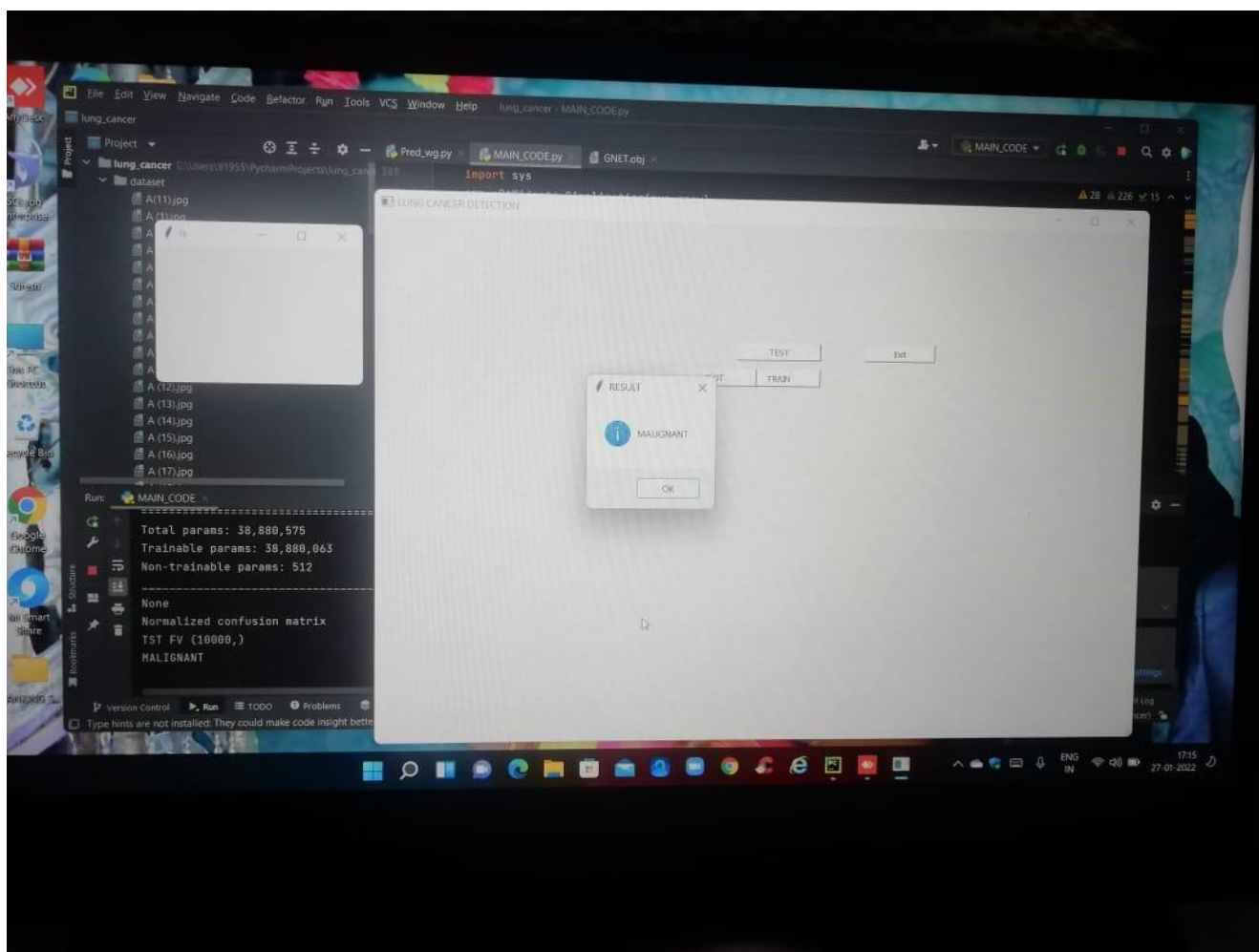
The above picture is the confusion matrix after training the model the confusion matrix will display



The above picture is showing the Result is NORMAL



The above picture is showing the Result is BENIGN



The above picture is showing the Result is MALIGNANT

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 CONCLUSION

- The proposed model shows the overview of prediction of lung cancer at an early stage. After prediction of the tumor begins malignant or benign, we generate a confusion matrix for each machine learning technique and based on the confusion matrix we calculate accuracy, Recall, precision and F1 score.
- From the result we can say that our proposed model can distinguish between benign and malignant, and it can be seen that artificial neural network is providing more accuracy

APPENDIX

A.SOURCE CODE:

```
from _future_ import print_function

##import tkinter
##import tkinter.filedialog
##from tkinter import messagebox
#root=tkinter.Tk()
from tkinter import filedialog

import os
from os import listdir
from os.path import isfile, join
import time
from PyQt5 import QtCore, QtGui, QtWidgets
import cv2

import math
import random
import string
import numpy as np
import numpy
import cv2
from array import array
from numpy import linalg as LA
from Pred_LSTM import Calc_Wt
import pickle

import matplotlib.pyplot as plt

from tkinter import *
from tkinter import messagebox
root = Tk()
```

```
root.geometry("300x200")
```

```
import itertools
```

```
def process(img, filters):
    accum = np.zeros_like(img)
    for kern in filters:
        fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
        np.maximum(accum, fimg, accum)
    return accum
```

```
def thresholded(center, pixels):
    out = []
    for a in pixels:
        if a >= center:
            out.append(1)
        else:
            out.append(0)
    return out
```

```
def get_pixel_else_0(l, idx, idy, default=0):
    try:
        return l[idx,idy]
    except IndexError:
        return default
```

```
def plot_confusion_matrix(cm, classes,
                           normalize=False,
                           title='Confusion matrix',
                           cmap=plt.cm.Reds):
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
```

```
print('Confusion matrix, without normalization')
```

```
#print(cm)
```

```
plt.imshow(cm, interpolation='nearest', cmap=cmap)
```

```
plt.title(title)
```

```
plt.colorbar()
```

```
tick_marks = np.arange(len(classes))
```

```
plt.xticks(tick_marks, classes, rotation=45)
```

```
plt.yticks(tick_marks, classes)
```

```
fmt = '.2f' if normalize else 'd'
```

```
thresh = cm.max() / 2.
```

```
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
```

```
    plt.text(j, i, format(cm[i, j], fmt),
             horizontalalignment="center",
             color="white" if cm[i, j] > thresh else "black")
```

```
plt.ylabel('TRUE CLASS')
```

```
plt.xlabel('PREDICTED CLASS')
```

```
plt.tight_layout()
```

```
print('***Start**')
```

```
try:
```

```
    _fromUtf8 = QtCore.QString.fromUtf8
```

```
except AttributeError:
```

```
    def _fromUtf8(s):
```

```
        return s
```

```
try:
```

```
    _encoding = QtWidgets.QApplication.UnicodeUTF8
```

```
    def _translate(context, text, disambig):
```

```
        return QtWidgets.QApplication.translate(context, text, disambig,
```

```
_encoding)
```

```

except AttributeError:
    def _translate(context, text, disambig):
        return QtWidgets.QApplication.translate(context, text, disambig)

class Ui_MainWindow1(object):

    def setupUi(self, MainWindow):
        MainWindow.setObjectName(_fromUtf8("MainWindow"))
        MainWindow.resize(1200, 800)
        MainWindow.setStyleSheet(_fromUtf8("\n""background-image:
url(III3.jpg);\n"""))
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName(_fromUtf8("centralwidget"))
        self.pushButton = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton.setGeometry(QtCore.QRect(750, 180, 111, 27))
        self.pushButton.clicked.connect(self.quit)
        self.pushButton.setStyleSheet(_fromUtf8("background-color:
rgb(255,255,255);\n"
"color: rgb(0, 0, 0);"))

        self.pushButton.setObjectName(_fromUtf8("pushButton"))
#####
#####

        self.pushButton_2 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_2.setGeometry(QtCore.QRect(550, 180, 131, 27))
        self.pushButton_2.clicked.connect(self.show1)
        self.pushButton_2.setStyleSheet(_fromUtf8("background-color:
rgb(255,255,255);\n"
"color: rgb(0, 0, 0);"))
        self.pushButton_2.setObjectName(_fromUtf8("pushButton_2"))

        self.pushButton_4 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_4.setGeometry(QtCore.QRect(550, 220, 131, 27))

```

```
self.pushButton_4.clicked.connect(self.show2)
self.pushButton_4.setStyleSheet(_fromUtf8("background-color:
rgb(255,255,255);\n"
"color: rgb(0, 0, 0);"))
```

```
self.pushButton_5 = QtWidgets.QPushButton(self.centralwidget)
self.pushButton_5.setGeometry(QRect(450, 220, 131, 27))
self.pushButton_5.clicked.connect(self.show2)
self.pushButton_5.setStyleSheet(_fromUtf8("background-color:
rgb(255,255,255);\n"
"color: rgb(0, 0, 0);"))
```

```
self.pushButton_5.setObjectName(_fromUtf8("pushButton_4"))
```

```
MainWindow.setCentralWidget(self.centralwidget)
self.statusbar = QtWidgets.QStatusBar(MainWindow)
self.statusbar.setObjectName(_fromUtf8("statusbar"))
MainWindow.setStatusBar(self.statusbar)
```

```
self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)
```

```
def retranslateUi(self, MainWindow):
    MainWindow.setWindowTitle(_translate("MainWindow", "LUNG
CANCER DETECTION", None))
    self.pushButton_2.setText(_translate("MainWindow", "TEST",
None))
    self.pushButton_4.setText(_translate("MainWindow", "TRAIN",
None))
    self.pushButton_5.setText(_translate("MainWindow", "PLOT",
None))
    self.pushButton.setText(_translate("MainWindow", "Exit", None))
```

```
def quit(self):
```

```
print ('Process end')
print ('***End***')
quit()
```

```
def show1(self):
```

```
    # -----
    #         MAIN CODE
    # -----
```

```
    # READ IMAGE
```

```
    image_path= filedialog.askopenfilename(filetypes = (("BROWSE
CT IMAGE", ".jpg"), ("All files", "")))
```

```
    img=cv2.imread(image_path)
```

```
    #img=cv2.imread('Train/B (5).jpg')
```

```
    img = cv2.resize(img,(100,100),3)
```

```
    img1 = cv2.GaussianBlur(img,(5,5),0)
```

```
    img1 = cv2.medianBlur(img,5)
```

```
    gray = cv2.cvtColor(img1,cv2.COLOR_BGR2GRAY)
```

```
    GF1=build_filters()
```

```
    TST1=process(gray, GF1)
```

```
    img = cv2.resize(TST1,(100,100))
```

```
    TST=np.reshape(img,(10000))
```

```
    TST=np.transpose(TST)
```

```
    #TST=np.transpose(TST)
```

```
    print('TST FV',np.shape(TST));
```

```
    TRR=[];
```

```
    #TRR=np.loadtxt('a.txt')
```

```
    file= open("GNET.obj",'rb')
```

```
    TRR = pickle.load(file)
```

```
    file.close()
```

```
    SUPLBL=[1,2,3,4,5,6,7,8,9,10]
```

```
    IND=Calc_Wt(TRR,TST)
```

```
    #print('INITIAL INDEX',IND)
```

```
    IND=int(np.ceil(IND/5));
```

```
    #print('CLASSWISE INDEX',IND)
```

```
    ind=IND
```

```
    #print(ind)
```

```

if ind==0:
    print('NORMAL')
    messagebox.showinfo(title='RESULT', message='NORMAL')
if ind==1:
    print('NORMAL')
    messagebox.showinfo(title='RESULT', message='NORMAL')
if ind==2:
    print('BENIGN')
    messagebox.showinfo(title='RESULT', message='BENIGN')
if ind==3:
    print('MALIGNANT')
    messagebox.showinfo(title='RESULT', message='MALIGNANT')

```

```

root.destroy()

```

```

def show2(self):

```

```

    import tensorflow as tf
    import matplotlib.pyplot as plt
    from keras.layers import Input, Dense
    from keras import regularizers
    from keras.models import Sequential, Model
    from keras.layers import Input, Flatten, Dense, Dropout,
BatchNormalization
    from keras.layers import Conv2D, MaxPooling2D,
AveragePooling2D, ZeroPadding2D
    from keras.layers import Concatenate
    from keras.preprocessing.image import ImageDataGenerator
    from tensorflow.keras.optimizers import Adam, SGD
    import pickle

    # define parameters
    CLASS_NUM = 5

```

```

BATCH_SIZE = 16
EPOCH_STEPS = int(4323/BATCH_SIZE)
IMAGE_SHAPE = (512, 512, 3)
IMAGE_TRAIN = 'Gnet'
MODEL_NAME = 'Gnet.h5'
def inception(x, filters):
    path1 = Conv2D(filters=filters[0], kernel_size=(1,1), strides=1,
padding='same', activation='relu')(x)
    path2 = Conv2D(filters=filters[1][0], kernel_size=(1,1),
strides=1, padding='same', activation='relu')(x)
    path2 = Conv2D(filters=filters[1][1], kernel_size=(3,3),
strides=1, padding='same', activation='relu')(path2)
    path3 = Conv2D(filters=filters[2][0], kernel_size=(1,1),
strides=1, padding='same', activation='relu')(x)
    path3 = Conv2D(filters=filters[2][1], kernel_size=(5,5),
strides=1, padding='same', activation='relu')(path3)
    path4 = MaxPooling2D(pool_size=(3,3), strides=1,
padding='same')(x)
    path4 = Conv2D(filters=filters[3], kernel_size=(1,1), strides=1,
padding='same', activation='relu')(path4)
    return Concatenate(axis=-1)([path1,path2,path3,path4])
def auxiliary(x, name=None):
    layer = AveragePooling2D(pool_size=(5,5), strides=3,
padding='valid')(x)
    layer = Conv2D(filters=128, kernel_size=(1,1), strides=1,
padding='same', activation='relu')(layer)
    layer = Flatten()(layer)
    layer = Dense(units=256, activation='relu')(layer)
    layer = Dropout(0.4)(layer)
    layer = Dense(units=CLASS_NUM, activation='softmax',
name=name)(layer)
    return layer
def googlenet():
    layer_in = Input(shape=IMAGE_SHAPE)
    layer = Conv2D(filters=64, kernel_size=(7,7), strides=2,
padding='same', activation='relu')(layer_in)

```

```

        layer = MaxPooling2D(pool_size=(3,3), strides=2,
padding='same')(layer)
        layer = BatchNormalization()(layer)
        layer = Conv2D(filters=64, kernel_size=(1,1), strides=1,
padding='same', activation='relu')(layer)
        layer = Conv2D(filters=192, kernel_size=(3,3), strides=1,
padding='same', activation='relu')(layer)
        layer = BatchNormalization()(layer)
        layer = MaxPooling2D(pool_size=(3,3), strides=2,
padding='same')(layer)
        layer = inception(layer, [ 64, (96,128), (16,32), 32]) #3a
        layer = inception(layer, [128, (128,192), (32,96), 64]) #3b
        layer = MaxPooling2D(pool_size=(3,3), strides=2,
padding='same')(layer)
        layer = inception(layer, [192, (96,208), (16,48), 64]) #4a
        aux1 = auxiliary(layer, name='aux1')
        layer = inception(layer, [160, (112,224), (24,64), 64]) #4b
        layer = inception(layer, [128, (128,256), (24,64), 64]) #4c
        layer = inception(layer, [112, (144,288), (32,64), 64]) #4d
        aux2 = auxiliary(layer, name='aux2')
        layer = inception(layer, [256, (160,320), (32,128), 128]) #4e
        layer = MaxPooling2D(pool_size=(3,3), strides=2,
padding='same')(layer)
        layer = inception(layer, [256, (160,320), (32,128), 128]) #5a
        layer = inception(layer, [384, (192,384), (48,128), 128]) #5b
        layer = AveragePooling2D(pool_size=(7,7), strides=1,
padding='valid')(layer)
        layer = Flatten()(layer)
        layer = Dropout(0.4)(layer)
        layer = Dense(units=256, activation='linear')(layer)
        main = Dense(units=CLASS_NUM, activation='softmax',
name='main')(layer)
        model = Model(inputs=layer_in, outputs=[main, aux1, aux2])
        return model

```

model= googlenet()

```
model.summary()
```

```
print(model.summary())
```

```
file= open("TRNMDL.obj",'rb')
```

```
cnf_matrix = pickle.load(file)
```

```
file.close()
```

```
plt.figure()
```

```
plot_confusion_matrix(cnf_matrix[0:3,0:3], classes=['NORMAL',  
'BENIGN','MALIGNANT'], normalize=True,title='Proposed Method')
```

```
plt.show()
```

```
if __name__ == "__main__":
```

```
    import sys
```

```
    app =QtWidgets.QApplication(sys.argv)
```

```
    MainWindow =QtWidgets.QMainWindow()
```

```
    ui = Ui_MainWindow1()
```

```
    ui.setupUi(MainWindow)
```

```
    MainWindow.move(550, 170)
```

```
    MainWindow.show()
```

```
    sys.exit(app.exec_())
```

REFERENCES

- [1] Krishnaiah, V., G. Narsimha, and Dr N. Subhash Chandra. "Diagnosis of lung cancer prediction system using data mining classification techniques." *International Journal of Computer Science and Information Technologies* 4.1 (2013): 39-45.
- [2] Zhang, Junjie, et al. "Pulmonary nodule detection in medical images: a survey." *Biomedical Signal Processing and Control* 43 (2018): 138- 147.
- [3] Fenwa, Olusayo D., Funmilola A. Ajala, and A. Adigun. "Classification of cancer of the lungs using SVM and ANN." *Int. J. Comput. Technol.* 15.1 (2016): 6418-6426.
- [4] Daoud, Maisa, and Michael Mayo. "A survey of neural network-based cancer prediction models from microarray data." *Artificial intelligence in medicine* (2019).
- [5] Gurcan, M. N., Sahiner, B., Petrick, N., Chan, H.-P., Kazerooni, E. A., Cascade, P. N., & Hadjiiski, L. 2002. Lung nodule detection on thoracic computed tomography images: Preliminary evaluation of a computeraided diagnosis system. *Medical Physics*, 29(11), 2552– 2558. doi:10.1118/1.1515762
- [6] Janee Alam, S., & Hossan, A. Multi-Stage Lung Cancer Detection and Prediction Using Multi-class SVM Classifier. 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)...
- [7] M.Gomathi, Dr.P.Thangaraj. 2010. A Computer-Aided Diagnosis System for Detection of Lung Cancer Nodules using Extreme Learning Machine. *International Journal of Engineering Science and Technology*. Vol. 2(10), 2010, 5770-5779.
- [8] Kakeda, S., Moriya, J., Sato, H., Aoki, T., Watanabe, H., Nakata, Doi, K. 2004. Improved Detection of Lung Nodules on Chest Radiographs Using a Commercial Computer-Aided Diagnosis System. *American Journal of Roentgenology*, 182(2), 505– 510. doi:10.2214/ajr.182.2.1820505.
- [9] Awai, K., Murao, K., Ozawa, A., Komi, M., Hayakawa, H., Hori, S., & Nishimura, Y. 2004. Pulmonary Nodules at Chest CT: Effect of Computer-aided Diagnosis on Radiologists' Detection Performance. *Radiology*, 230(2), 347352. doi:10.1148/radiol.2302030049.