A DEEP LEARNING MODEL FOR PREDICTION OF MALIGNANT TUMORS IN HUMAN BODY WITH SPECIAL REFERENCE TO MULTIMODAL IMAGING TECHNIQUES

A thesis submitted in partial fulfillment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

Dipanjan Moitra

Under the Supervision of

Dr. Rakesh Kumar Mandal

(Associate Professor)

Department of Computer Science & Application,

University of North Bengal

August, 2020

A DEEP LEARNING MODEL FOR PREDICTION OF MALIGNANT TUMORS IN HUMAN BODY WITH SPECIAL REFERENCE TO MULTIMODAL IMAGING TECHNIQUES

A thesis submitted in partial fulfillment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

Dipanjan Moitra

Under the Supervision of

Dr. Rakesh Kumar Mandal

(Associate Professor)

Department of Computer Science & Application,

University of North Bengal

August, 2020

DECLARATION

I declare that the thesis entitled ***A Deep Learning Model for prediction of Malignant Tumors in human body with special reference to Multimodal Imaging Techniques*** has been prepared by me under the guidance of Dr. Rakesh Kumar Mandal, Associate Professor, Department of Computer Science & Application, University of North Bengal. No part of the thesis has formed the basis for the award of any degree or fellowship previously.

Dipanjan Moitra

(Dipanjan Moitra)

Department of Management, University of North Bengal,

Raja Rammohunpur, P.O. NBU, District: Darjeeling, West Bengal,

India, PIN-734013

Date: 26/08/2020

CERTIFICATE

I certify that Dipanjan Moitra has prepared the thesis entitled ***A Deep Learning Model for prediction of Malignant Tumors in human body with special reference to Multimodal Imaging Techniques*** for the award of the Ph.D. degree of the University of North Bengal, under my guidance. He has carried out the work at the Department of Computer Science & Application, University of North Bengal.

(Dr. Rakesh Kumar Mandal)

Associate Professor, Department of Management, University of North Bengal,

Raja Rammohunpur, P.O. NBU, District: Darjeeling, West Bengal,

India, PIN-734013

Date: 26/08/2020

Associate Professor
Department of Computer Science & Application
University of North Bengal

Dedicated

To

My Grandparents

Acknowledgments

I feel greatly honoured to express my deepest and most sincere gratitude to my respected supervisor Dr. Rakesh Kumar Mandal, Associate Professor, Department of Computer Science & Application, the University of North Bengal for providing continuous encouragement, thought-provoking suggestions and thorough guidance throughout the work. His foresightedness helped me a lot to complete my thesis. Without his kind supervision, this work would not have been materialized.

I also express my thanks to all my teachers, colleagues, and fellow researchers for extending their valuable suggestions, and help throughout my work.

I would like to offer my special thanks to the University Authority and UGC for allowing me to carry out this work at this university.

I would like to express my gratefulness to my beloved parents, Shri Pinaki Maitra and Late Lila Maitra. Last but not the least, I am indeed indebted to my wife Smt. Tihi Moitra and my son Master Tirthadip Moitra for their constant support, encouragement, and patience, without which the thesis would not have been completed in time.

Date: 26/08/2020

Dipanjan Moitra

(Dipanjan Moitra)

Abstract

Several attempts have so far been made to detect malignant tumors by combining biomedical imaging and machine learning techniques. The machine learning methods employed in such studies are often found to be quite off-the-shelf in nature. Types of tumors being considered in these studies are organ-specific. Thus, there is a need to implement the latest machine learning methodologies along with multimodal imaging techniques while developing an automated tumor diagnostic model. In this way, a more general yet improved machine learning model should be developed for the detection of malignant tumors found in the human body.

The present research work has developed a deep neural network model for automated staging and grading of malignant tumors. The steps include data acquisition, image segmentation, feature extraction, feature selection, classification, and evaluation of results.

Data Acquisition has been done by The Cancer Imaging Archives (TCIA). Eight different types of tumors have been considered in the study. A new heterogeneous imagery database has been thus prepared and used in the study.

Image Segmentation has been done by conducting a benchmark survey with the leading segmentation methods and the texture-based method has been selected as the pivotal technique.

Feature Extraction is done mainly by using Speeded Up Robust Features (SURF) and Maximally Stable Extermal Regions (MSER) techniques.

Feature Selection is carried out by using Independent Component Analysis (ICA) method.

Classification has been accomplished by developing a new one-dimensional and two-dimensional convolutional neural networks combined with recurrent layers. Both sequential and non-sequential methods have been applied to develop new models. A one-dimensional model is a semi-automated classifier and the two-dimensional model is a fully-automated classifier.

Evaluation has been done by using different statistical parameters. The results of classification obtained from different machine learning models used in the study have been compared and the final model is selected.

Preface

Computer-aided systems when combined with machine learning algorithms often found useful in medical informatics. Such systems may also be employed for classifying malignant tumors. This helps oncologists and radiologists to plan effective treatment strategies. In today's context, Deep Learning has almost outperformed other machine learning techniques, especially in the domain of computer-assisted visualization. Deeper hidden layers may lead to more accurate feature extraction and classification of the target set of tumor images. The present research work has used different datasets to prepare a non-uniform collection of multi-modal tumor images which are not organ-specific. After reviewing the existing literature, research gaps are identified. Tumor classification has been done in two varieties: semi-automated and fully automated. In a semi-automated method, images are pre-processed, segmented, and then features are extracted. Important attributes are selected and classified by using available clinical information as the class variables successively. In the case of a fully automated method, no manual preprocessing, segmentation, feature extraction, and selection are required. Image pixel arrays combined with clinical information are directly fed into the model. Results are evaluated by using different statistical parameters. A new technique has been developed where non-sequential branches of convolutional layers with occasional re-injection of input acts as pre-processor followed by bidirectional recurrent layers. The model performed satisfactorily and expected to contribute significantly to automated tumor prognosis.

Date: 26/08/2020

Place: NBU Campus

(Dipanjan Moitra)

Table of Contents

1. Introduction

Cancer is one of the most frightening and deadly diseases on Earth. It is one of the leading causes of untimely mortality in developed countries. The tale is not much different in developing countries. As per the Indian Council of Medical Research (ICMR), More than 1300 people in India die every day because of cancer. As per World Health Organisation (WHO) [1], deaths from cancer are likely to go beyond 13.1 million in 2030 worldwide. Except for cancers of the blood, most of the other cancers form solid tumors in the human body irrespective of locations [2]. Tumors are results of uncontrolled and rapid cell division in living organisms. Tumors are basically of two types: benign and malignant. Benign tumors are typically non-cancerous and malignant tumors are prone to grow cancer cells within it. The difference between them is subtle and a benign tumor may anytime turn into a malignant one. Cancerous tumors grow at a quick pace and may damage nearby and far-off organs besides affecting the originating organ [3]. Treatments include surgery, chemotherapy, and radiation. Proper treatment of such a tumor needs early and correct diagnosis. Early diagnosis increases the chances for cure and assures longer survival of the patient [4]. Normally patients experience painful biopsy during the manual tumor diagnosis process. Such manual identification of tumors is not fully error-free. Human error and bias may sometimes affect the prognosis. Moreover, cancerous tumors are often detected at an advanced stage and this limits the chances for getting ample tissue samples during surgical resection [5]. Thus, the correct prognosis of tumors also becomes uncertain, especially if it is very small in size in case of an early-stage tumor or a recurrent one. Here, bio-medical imaging techniques may play a significant role when used with a Computer-Aided System (CAD) capable enough to classify tumor images. After extracting important attributes from a series of tumor images, a pattern recognition technique may be applied for correctly figuring out the type or stage of a tumor. This may give birth to a painless and bloodless auxiliary decision support system which will help radiologists and oncologists in determining the treatment plan more affirmatively. If the orthodox clinical method results in a doubtful diagnosis, medical personnel may seek the help of such an automated tumor classification system to confirm the prognosis.

In fine, it may be stated that to classify malignant tumors successfully, a typical machine learning model may incorporate a variety of cancer datasets or prepare a heterogeneous tumor dataset. The dataset should contain diversified imaging modalities. While experimenting, different leading machine learning techniques may be explored for developing a new efficient model. After comparing the outcome with the existing models, the new model may be finalized. The performance of the model may also be compared with the related contemporary studies. In this way, the study may become more reliable and also stand workable as far as its application in the real world is concerned. While automating the clinical diagnosis for malignant tumors in humans, most happening technologies like Deep Neural Networks (DNN) may play an important role in developing an improved scientific model. Upon developing such a scientific prototype, its commercial extensions can easily be implemented.

1.1 Categorization of Tumors

Although tumors are broadly categorized as benign and malignant, this is not sufficient to plan a proper treatment strategy. Tumors can also be graded as less aggressive to very aggressive, but such a classification may not be very informative. Different agencies have categorized tumors differently. For example, the World Health Organisation (WHO) has classified brain tumors into 120 types according to their location and cell type [6]. Such a huge classification may not be operationally feasible for feeding in the machine learning process. Fortunately, the American Joint Committee on Cancer (AJCC) has propounded a more technically feasible and uniform bifurcation on tumor types. The most popular categorization of the tumor is pathological Tumor-Node-Metastasis (TNM) staging and histopathological grading. Staging shows how much a tumor has already spread in the body and grading depicts how fast it may spread in the future. For example, the basic TNM staging nomenclature for malignant tumors is as follows [7]:

T: Size of the primary tumor and if cancer has spread in nearby tissues {Ti; i= 0 to 4}. May be further classified by using the lowercase letters a, b & m

N: If cancer has spread into nearby lymphatic nodes {Ni; i= 0 to 3}

M: If cancer has spread to other organs of the body {Mi; i= 0 to 1}

A more generalized version of TNM staging is also available, and this is typically known as the AJCC staging. On the other hand, a typical 4-tier histopathological grading system for malignant tumors is as follows [8]:

G1: Well differentiated

G2: Moderately distinguished

G3: Poorly distinguished

Other (Type I: Well to moderately distinguished and Type II: Moderately to poorly distinguished)

Histopathological grading results from histology subtype cataloging, e.g., NSCLC has major subtypes like lung adenocarcinoma, large carcinoma, and squamous cell carcinoma. The most common primary brain tumors found in adults are Gliomas, Meningiomas, Schwannomas, Pituitary Tumors, and CNS Lymphoma [9].

The automation of the aforementioned staging and grading systems may lead to the development of an efficient machine learning model which may, in turn, be used as a malignant tumor classification system.

1.2 Medical Imaging Techniques

A few medical imaging techniques are structure-oriented, such as Computed Tomography (CT), or Magnetic Resonance Imaging (MRI), and a few are function-based like Positron Emission Tomography (PET) [10]. Both the structural and functional techniques have their pros and cons. Structural imaging techniques rely on the anatomical structure of a tumor, whereas functional imaging techniques depend on biochemical changes taking place in and around a tumor. Normally, structural techniques efficiently demarcate tumor location, size, and boundary. The problem arises when there is a leakage in the boundary wall of a tumor or when the tumor size is tiny, especially with recurrence. In the aforementioned cases, structural imaging techniques may not work efficiently as they do in normal situations. Functional imaging techniques may pick up a minute change in bio-metabolism and they can perform well even in such adverse situations [11]. However, there are problems with functional techniques, too. Hyper-metabolism may sometimes be misjudged as a tumor by the functional imaging, but it can hardly deceive a structural one. Thus, it may be concluded that functional and structural imaging techniques are balancing each other. Therefore multimodal imaging techniques,

where both structural and functional techniques work in tandem, have gained immense popularity. Multimodal imaging techniques, such as PET/CT or MRI-PET, keep the benefits of both structural and functional techniques and neutralize the deformities of each other (Figure 1). Among the available multimodal imaging techniques, PET/CT is the most widely used one. So, PET/CT may be considered as the pivotal imaging modality in the study alongside PET, MRI, and CT-scans, etc.

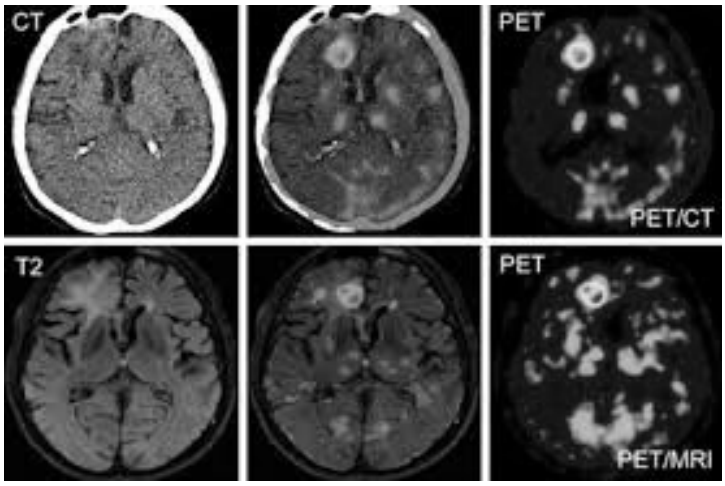


Figure 1: Fusion of PET with CT & MRI [160]

1.3 Medical Imaging Database

The Cancer Imaging Archive (TCIA) is an open repository of international repute [12]. It hosts large records of bio-medical images of different cancer accessible by researchers worldwide. The imagery data are organized as *Cases* indexed by disease, image modality, or research focus. DICOM is the default imagery file format used by TCIA for image storage. DICOM (Digital Imaging and Communications in Medicine) is a standard for embedding imaging and clinical information together (Figure 2) [165]. TCIA may be selected as the key source of data while conducting a study to classify tumors found in human beings.

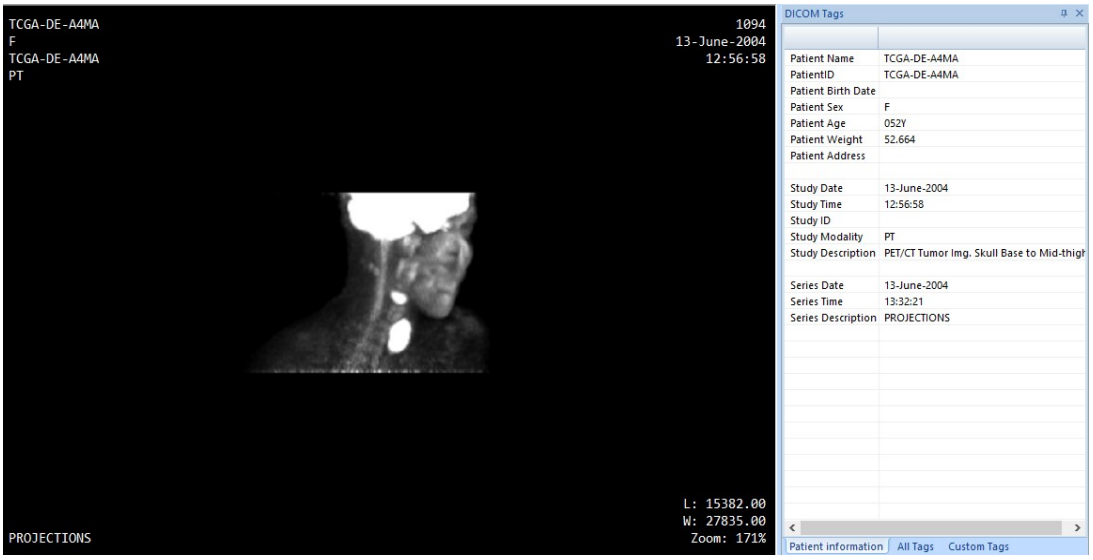


Figure 2: A DICOM image depicting a Head & Neck Tumor image and the associated clinical information fetched during experiments carried out in the present research work

In TCIA, clinical data including pathological stage or grade related information, treatment strategy adopted, recurrence, gender, survival status, etc. are also provided along with the image dataset (Figure 3). Clinical information is also reliable as they are being prepared by internationally acclaimed oncologists and scientists.

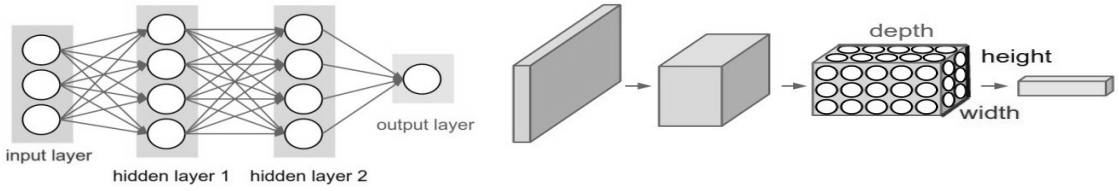
Case ID	Gender	Histology	Pathological T stage	Pathological N stage	Pathological M stage	AJCC Staging (Version 7)	Histopathological Grade	Chemotherapy	Radiation	Recurrence	Survival Status
R01-001	Female	Adenocarcinoma	T1a	N0	M0	IA	G2 Moderately differentiated	No	No	no	Alive
R01-002	Female	Adenocarcinoma	T1a	N0	M0	IA	G1 Well differentiated	No	No	no	Alive
R01-003	Male	Adenocarcinoma	T3	N0	M0	IIB	Other, Type I: Well to moderately	No	No	no	Alive
R01-004	Male	Squamous cell carcinoma	T1b	N2	M0	IIIA	G2 Moderately differentiated	Yes	Yes	yes	Alive
R01-005	Male	Adenocarcinoma	T2a	N0	M0	IB	G3 Poorly differentiated	No	Yes	yes	Dead
R01-006	Male	Adenocarcinoma	T1b	N0	M0	IA	G1 Well differentiated	No	No	no	Alive
R01-007	Male	Squamous cell carcinoma	T1a	N1	M0	IIA	G3 Poorly differentiated	Yes	No	yes	Dead
							G1 Well				

Figure 3: Glimpse of Radiogenomics Clinical data from TCIA depicting the case based staging and grading information of NSCLC clinically endorsed by Dr. Sandy Napel and team (Stanford) [Source: https://www.cancerimagingarchive.net/]

Such clinical information may be very handy when appended as target labels in a tumor image dataset and used in a machine learning model to identify the stage or grade of a tumor.

1.4 Machine Learning Techniques

Artificial intelligence is nothing but the automation of various intellectual tasks ordinarily or extra-ordinarily performed by human beings, and machine learning is a typical subset of it [13]. Machine Learning is all about developing a model that transforms the data and the outcome is tested by a loss function. Subsequently, an algorithm may be used for adjusting the parameters of the model to reduce the loss. Machine learning has three pivotal categories: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the target variable regarding an input dataset is known beforehand (e.g., regression, sequence learning, etc.), whereas the target variable is unknown with unsupervised learning (e.g., clustering, dimensionality reduction, etc.). In reinforcement learning, an intelligent agent capable of interacting with the external environment is developed (e.g., robotics, etc.). Deep Learning is probably the most happening derivative of machine learning in today's context. To solve a difficult problem properly, one has to concentrate on the minute details of the same. After going through a lot of trial and error, one may come up with a better solution for the problem at hand. The same process goes on when someone uses machine learning to solve a problem. Here, human intelligence is mapped onto the machine. If the problem is complex and if a better solution is needed, a better technique should be used. *Deep Learning* is inevitably the digital replica of such a deep thought process. Deep Learning models have outperformed the traditional Artificial Neural Network (ANN) and other machine-learning techniques in computerized vision [14], [15]. A machine learning model par excellence can easily be developed by using Deep Neural Networks (DNN) as it uses deeper convolutional layers [16]. DNN also distributes the number of neurons along the spatial dimensions (Figure 4). Thus DNN becomes more conducive for image processing than the conventional Artificial Neural Network (ANN) methods which may suffer from problems like a huge number of dot products, inflexibility in accommodating large numbers of hidden layers, etc.



(a) (b)

Figure 4: (a) A regular 3-layer Neural Network (b) A CNN where the neurons are arranged in three dimensions [Source: https://cs231n.github.io/convolutional-networks/]

Structure of a typical DNN: INPUT layer injects the pixel values, CONVOLUTION layer moves across the pixels adjoining to input and decides regarding the output, Rectified Linear Units (RELU) acts as layer-wise activation function, POOLING layer performs a down-sampling operation and Fully Connected (FC) layer computes the class scores (Figure 5).

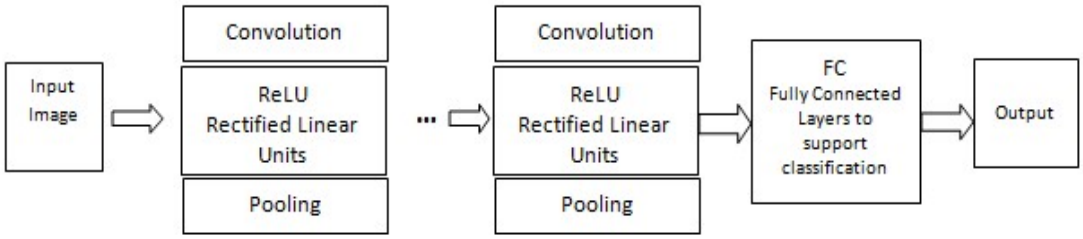


Figure 5: Conventional structure of a DNN model [Source: https://in.mathworks.com/]

1.5 Objective

The primary aim of the research work is to develop a machine learning model capable of detecting malignant tumors more accurately than the existing ones. Deep learning is the pivotal technique used in the research work. However, other leading machine learning techniques including the traditional Artificial Neural Network (ANN) cannot be left unturned. The research work has hence looked for the improved varieties of the orthodox machine learning algorithms, including various deep learning methods. The primary objective gives birth to other auxiliary objectives such as acquiring tumor image data, formulation of segmentation strategy, feature extraction, database preparation, data pre-processing, training & testing of CAD models, and evaluation of results got. In a nutshell, the problem at hand is to identify malignant tumors by using biomedical imaging techniques and by developing a useful machine learning model. The overall objectives of the research work are enumerated as:

a) To acquire tumor image data from a reliable repository and to preprocess them;

b) To plan an optimal image segmentation strategy by studying the existing methods;
c) To extract features from the segmented images and to prepare a dataset;
d) To pre-process the final dataset;
e) To carry out experiments with different machine learning method;
f) To develop a semi-automated deep learning model;
g) To further develop a fully automated deep learning model;
h) To test the results got and to compare the results;

1.6 The Workflow

It is a very challenging task to develop an effective automated system that may accurately classify malignant tumors. A holistic approach requires a series of distinct tasks to be performed carefully. The research work has been started with one particular type of tumor, and a model has been formulated for predicting various stages of such a tumor. Once the model gets established as a less costly yet effective way of predicting the concerned tumor-stage, it has been applied to classify other types of tumors. The workflow of the overall thesis is as follows (Figure 6):

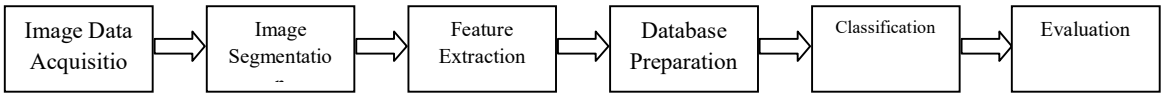


Figure 6: The workflow of the present study

1.6.1 Image Data Acquisition

The image dataset is retrieved from an open, authentic repository (TCIA) having accompanied by sound clinical information and endorsed by a team of eminent doctors and scientists. This has increased the reliability of the study and its outcome. After acquiring the dataset, several image processing operations have been applied to the dataset. This fine-tunes the image dataset and gets them ready for future steps.

1.6.2 Image Segmentation

The pre-processed image dataset has been used to fetch the specific Region of Interest (RoI). Thus, image segmentation operations have been carried out to extract the tumor part from the image.

1.6.3 Feature Extraction

The segmented image has been used for feature extraction. Different tumors have different image patterns, and those patterns have been recorded as features or attributes in a data file.

1.6.4 Database Preparation

The clinical information retrieved from the repository has been clubbed with the data file and one target variable is set for classification. As different cancer has different histological categories, a standard way of classification has been used in the study. In this way, different tumors could be included in the study. The data file has a cross-platform compatible extension i.e. *Comma Separated Value (CSV)*. Once the target variable is set, the data file is analyzed for finding out if any discrepancies are there. The dataset might have missing values or it might be unbalanced. Such discrepancies have been removed and important attributes are considered during classification.

1.6.5 Classification

The system has been bifurcated into a semi-automated and fully automated system. The present study explores both avenues, one after another. A semi-automated system undergoes many distinct steps to accomplish the classification task. Both the orthodox machine-learning methods and their improved varieties have been applied for classifying tumors. Deep learning has been the most widespread technology that has been used for biomedical image classification. A new model has been derived through experimentation and used in the study. In contrast, the fully automated approach does not need much prior processing. Images with proper tagging are fed directly into the model and training & test results are recorded.

1.6.6 Evaluation

The final mode of the system has been determined as per different candidates' respective accuracy levels of prediction and performance concerning other evaluation metrics. Results obtained have been evaluated in the light of different statistical metrics, and the outcome is compared with other models used in the study. This depicts the efficacy of the newly formed model. A tradeoff is set between the delay characteristic of the semi-automated method and the resource constraints of the fully automated method.

1.7 Outline of the thesis

First, an appropriate medical imaging modality has been selected. There are many options available, but the selected modality has to have an edge over others so that the candidate tumor may get identified perfectly. Sometimes it becomes tough to separate tumors from body organs due to physical and bio-chemical deceptiveness. The model developed in the present study helps in overcoming such problems. A set of tumor images are fed into the deep learning model which, upon sufficient training, yields validated test results. The dataset prepared is also compatible across different platforms. In this way several tools and techniques could be applied on the dataset as and when required. The thesis has been divided into the following ten chapters:

Chapter-1: Introduction. In this chapter, the importance of the present work has been discussed. It also contains brief descriptions regarding tumor categories, medical imaging techniques, medical imaging databases, and machine learning techniques. Here, the objective of the study and the workflow has been discussed. At last, the outline of the thesis has been depicted.

Chapter-2: Deep Learning: An Overview. In this chapter, a detailed mathematical description regarding various ANN and DNN techniques used in the study has been introduced. This includes techniques like a Convolutional Neural Network (CNN), a two-Dimensional Convolutional Neural Network (2-D CNN), and Recurrent Neural Network (RNN), etc.

Chapter-3: Review of Literature. This chapter surveys the existing literature connected to the automated prognosis of malignant tumors. It includes eight types of cancers such as Lung Cancer, Breast Cancer, Bladder Cancer, Renal Cancer, Liver Cancer, Thyroid Cancer, Uterine Cancer, and Head & Neck Cancer. Studies done with each type of cancer during the last ten years have been reviewed and the research gap has been identified accordingly.

Chapter-4: Image Acquisition. In this chapter, first, the imaging modalities have been selected as per their respective relevance and availability. From different databases, a series of DICOM images have been retrieved. Each database represents a particular genre of a tumor. Stored images are pre-processed to get them ready for feeding in the segmentation phase. The set of images thus collected is a diversified mix in terms of the

originating organ of the tumor, tumor types, treatment strategy, scanner modality, treatment outcome, and demography of patients. An image database thus prepared is having a mixed genotype and phenotype i.e. varied hereditary structure and observed properties.

Chapter-5: Image Segmentation. In this chapter, the existing literature has been surveyed to determine contemporary segmentation strategies. The study has also compared the performance of the existing image segmentation methods. Later on, an optimal image segmentation strategy has been planned. The segmented tumor regions are extracted from the concerned DICOM images as accurately as possible.

Chapter-6: Feature Extraction. In this chapter, scalable features suitable for machine learning have been derived from a pool of segmented images. Traditional feature extraction techniques, and their hybrids, have been applied to extract features.

Chapter-7: Database Preparation. In this chapter, extracted features are further pre-processed and a final dataset is prepared by clubbing them with the clinical information fetched from the TCIA repository. Experiments are performed in a stratified manner where each stratum embeds a different target variable into the dataset. Such target variables are the identifiers helpful in staging and grading of tumors.

Chapter-8: Development of a One-dimensional CNN. In this chapter, prolonged experiments have been carried out with the final dataset. Both standard machine learning algorithms and their improved varieties are triggered. The same procedure has been applied with varied deep learning methodologies. Experiments are conducted with different hyper-parameter options available with these algorithms. Thus, a semi-automated system has been developed. It can diagnose malignant tumors with an improved level of accuracy. The semi-automated system is lightweight in terms of consuming resources. Such an approach is very useful where Graphical Processing Units (GPUs), cloud services, etc. are scarce or not easily affordable.

Chapter-9: Development of a Two-dimensional CNN. In this chapter, a fully automated system has been developed. A fully automated system does not require image segmentation and needs only fewer pre-processing. A two-dimensional CNN along with bi-directional Long Short Term Memory (LSTM) has been used here. The model has

been further improved by introducing a new technique called a *deep recurrent ensemble of branched residual neural networks.*

Chapter-10: Results & Discussion. In this chapter, the results obtained from the experiments performed are evaluated by various standard statistical measures as applicable. The evaluation results derived from different models are compared with each other. Such a comparison creates the opportunity to decide on a new deep learning model, which acts as a remedy to the problem at hand. This exhibits not only the relevance of the new model but will also show how the work may be carried out in the days to come.

1.8 Conclusion

The introductory chapter depicts both the concept behind the research work and ways how the concept has been materialized. It shows the need for carrying out such work. In this chapter, the categorization of tumors has been described and different medical imaging techniques have also been discussed. Various relevant imagery databases and machine learning techniques have also been narrated successively. These discussions show in how many ways tumors may be classified and which medical imaging techniques may be incorporated in the study. The narration also demonstrates different modalities of imaging databases and the utility of machine learning algorithms in classifying tumors. The objective of the study explains the need for carrying out the present research work. The workflow enumerates numerous tasks that are being performed during the lifetime of the study, from data collection to evaluation of outcome. The outline of the thesis has also been included to demarcate different chapters and their short descriptions. In short, the chapter introduces the basic driving forces behind the present research work.

2. Deep Learning: An Overview

Being a branch of Artificial Intelligence, Data Mining has gained immense popularity in different domains of computer application [17]. This includes complex fields like computational biology and precision medicine [18]. Data Mining solves real-life problems by extracting inherent data patterns hidden inside the dataset [19]. Machine learning tools can predict the outcome from an input set. Artificial Neural Network (ANN) is one of the major machine learning techniques developed as the digital replica of biological neurons. McCulloch-Pitts first proposed the *Artificial Neuron* model [20]. ANN came into the forefront of the machine learning domain when Frank Rosenblatt coined the term *Perceptron* [21]. Deep learning became popular when Geoffrey Hinton did his groundbreaking research on *Deep Neural Network* [22].

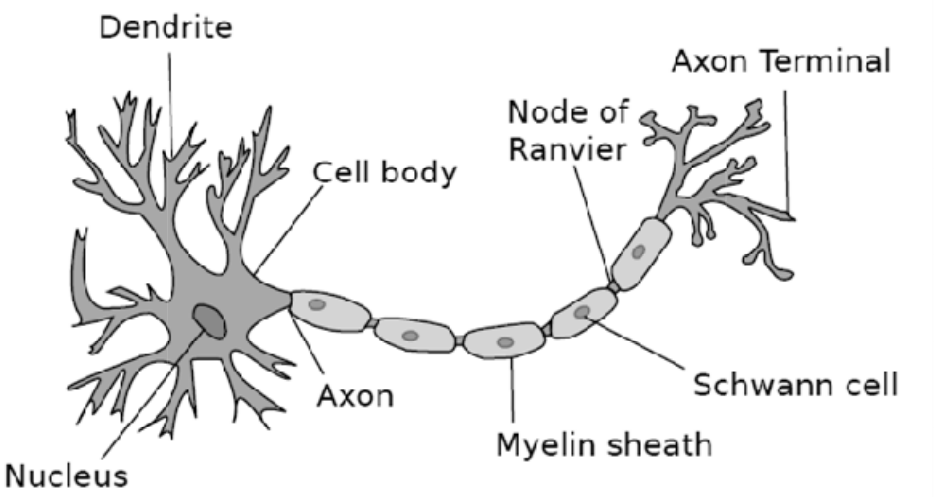


Figure 7: Typical structure of a biological neuron [23]

The human brain comprises billions of interconnected neurons (Figure 7) or nerve cells. Upon the occurrence of any external or internal event, neurons get neuro-chemical stimulation and propagate polarized electrical pulses through the interconnection [24]. This propagation ultimately results in the physiological response to the event that occurred. A biological neuron has four main parts: **Dendrites** are the fibers emanating from the cell body and receive activations from other neurons; **Soma** or the cell body processes the incoming activations and converts them to output activations; **Axons** are fibers which transmit the output activations to other neurons; **Synapses** are the junctions

that allow communication between dendrites and axons by diffusing chemicals. When strong input activations reach the nucleus inside the Soma, it sums up the activations and if the sum crosses a certain threshold, some output activations are generated. Otherwise, the input activations get decayed, and no output is generated.

2.1 Artificial Neural Network

The prime driving force of Artificial Neural Network (ANN) is called a perceptron which comprises an input, bias, weights, activation function, and output [25].

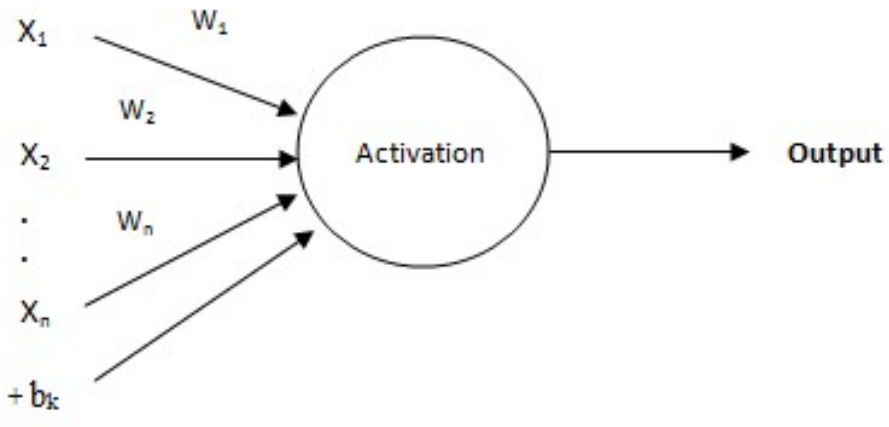


Figure 8: Structure of a typical Perceptron

In figure 8, $X_1, X_2 \ldots X_n$ is the input set; W_1, $W_2 \ldots W_n$ are respective weights; $+b_k$ is the bias. The output may be expressed as:

$$output = \begin{cases} 0, & w.x + b \leq 0 \\ 1, & w.x + b > 0 \end{cases} \quad \ldots \text{Equation 2.1.1}$$

Where, w and x are tensors representing weights and inputs, respectively. Here, w. x $\equiv \sum_j w_j x_j$ and the activation function may be expressed as $v_k = \sum_{j=1}^{n} w_{kj}\, x_j + b_k$. The bias b stands for negative threshold i.e. the quantity after adding which it is easier to get to the desired output. A single perceptron acts like a binary classifier and can solve linear problems. Not every problem is binary, and the independent variables may have non-linear relationships with the dependent variable. In those situations, a single perceptron may not be very useful. Therefore, the concept of Multilayer Perceptron (MLP) came into existence. In MLP, the input layer is succeeded by hidden layers comprising several perceptrons which are followed by the output layer (Figure 9). In this way, the non-linear

relationships of variables are preserved and multiclass problems may also be solved easily. The mathematical expression of MLP is:

$H_{l-1} = f(W_{l-1}X + b_{l-1})$

$H_l = f(W_l H_{l-1} + b_l)$ … Equation 2.1.2

$O = f(W_{l+1}H_l + b_{l+1})$

Where X is the input vector; W is the weight vector; b is the bias; *l* is the corresponding layer number; O is the output and f (…) is the non-linear activation function.

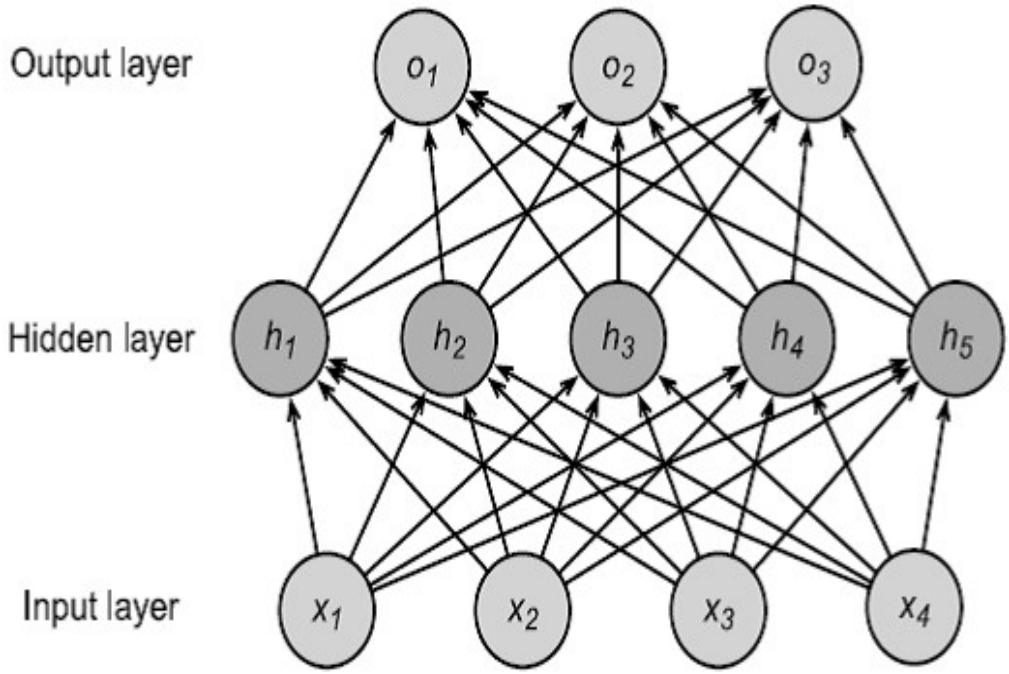


Figure 9: Structure of a typical Multilayer Perceptron with four inputs, a single hidden layer with five nodes and three output classes [26]

Thus, a typical artificial neural network comprises the input layer, followed by several hidden layers and the final output layer. The normal flow of ANN is called feed-forward pass where the input is injected and the output is derived by following the same direction. If the general form of ANN is considered as Y=WX+b, it may be understood that a small change in any weight or bias causes a small change in output.

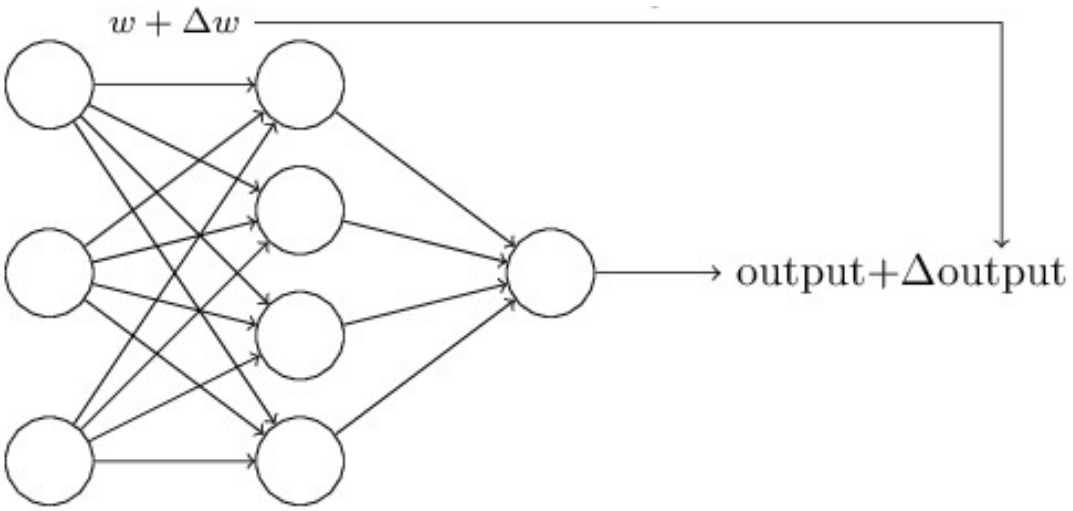


Figure 10: Backpropagation in Neural Network [27]

Thus, the output gets updated with the help of partial derivation and this technique is called backpropagation [28]. The adjusted output is calculated as:

$$\Delta\text{output} \approx \sum_j \frac{\partial output}{\partial w_j} \Delta w_j + \frac{\partial output}{\partial b} \Delta b \quad \text{… Equation 2.1.3}$$

Accordingly, the minimization of loss leads to generate optimized weight and bias:

$$w^*, b^* = \text{argmin}_{w,b}\, L(w, b) \quad \text{… Equation 2.1.4}$$

In this way, a more improved version of the neural network model can be developed.

2.2 Activation Functions

Different activation functions are used in different layers as per convention. Activation functions are responsible for non-linear transformation and final prediction.

2.2.1 Rectified Linear Unit (ReLU)

ReLU does element-wise activation and are mainly used in intermediate hidden layers and expressed as:

$$\text{ReLU}(x) = \max(0, x) \quad \text{… Equation 2.2.1.1}$$

If the input is negative, the output is zero and if the input is positive, the output is 1.

2.2.2 Sigmoid

The sigmoid function transforms an input having the domain (-inf, inf) to some values having the range (0, 1):

$$\text{Sigmoid}(x) = 1/(1+e^{-x}) \text{ where } x \in R \quad \text{… Equation 2.2.2.1}$$

2.2.3 Tanh

The tanh or hyperbolic tangent function transforms the input to some values in the range (-1, 1):

$\text{Tanh}(x) = (1-e^{-2x})/ (1+e^{2x})$ … Equation 2.2.3.1

2.2.4 Softmax

The most popular activation is the softmax method, which is used typically for detecting the most likely class from the final layer outcome. In other words, it measures the probability of the result got from the final layer. By using equation 2.1.2, the class may be predicted as:

$\hat{Y}=\text{softmax}(O)$ … Equation 2.2.4.1

In equation 2.2.4.1, O is the final layer output and $\hat{Y}$ is the probability score which may be further expressed as:

$\hat{Y}_i = (\exp(o_i)/\Sigma_j \exp(o_j))$ … Equation 2.2.4.2

Where, o_i is the relative levels of confidence for belongingness to each class i and $0< \hat{Y}_i <1$. The most likely class may be found by

$\hat{\imath}(o) = \text{argmax}_i\, o_i = \text{argmax}_i\, \hat{Y}_i$ … Equation 2.2.4.3

2.3 Loss Function

In machine learning, data is the raw fact a model learns from. A model shows how to transform the data into processed information. A loss function measures the inefficiency of the model. The pivotal objective is to update the model's parameters and to minimize the loss by invoking an algorithm. The most common losses are L1 loss or mean absolute loss between actual and predicted values:

$l(y,y')= \sum_i |y - y'|$ … Equation 2.3.1

and L2 loss or mean squared loss between actual and predicted values:

$l(y,y')= \sum_i (y - y')^2$ … Equation 2.3.2

The most popular loss function for the multiclass classification problem is the cross-entropy loss. This concept has come from entropy, which measures how many bits of information are contained in data. The entropy of a probability distribution p is calculated as:

$H[p] = \sum_j -p(j) \log\ p(j)$ … Equation 2.3.3

Similarly, the cross-entropy loss is measured as:

$L(Y, Y') = - \Sigma_j Y_j \log Y'_j$ … Equation 2.3.4

Where Y is the surrogate truth and Y′ is the predicted value. Thus the ultimate goal is to minimize the negative log-likelihood or to maximize the accuracy.

$L^*(Y, Y') = arg\min_{\hat{Y}_i} \sum_{i=1}^{n} H[p(Y_i), p(Y'_i)]$ … Equation 2.3.5

2.4 Convolutional Neural Network

ANNs and other machine learning techniques have been successfully applied in various fields [29], [30], [31], but there are a few major limitations of such techniques. ANNs and other leading machine learning methods must pass through a data pre-processing phase followed by a manual feature extraction curriculum. These make not only them sub-optimal but also make them highly resource-consuming methods, especially in case of complicated technical applications. For example, in computer vision data typically takes the form of matrices having arranged in more than one axis. In such cases, numbers of dot products also increase significantly while using ANN. On the other hand, a Convolutional Neural Network (CNN) reduces the number of dot products by bifurcating them along the spatial dimensions. There is also a limitation on the number of hidden layers that a typical ANN can accommodate [32]. With the help of advanced hardware setup such as cloud services, Graphical Processing Units (GPUs), etc. CNNs have surpassed such resource constraints. With deeper hidden layers, CNNs can achieve higher accuracy levels than other machine learning techniques. Unlike traditional machine learning methods, CNNs can automatically pre-process data and extract features. It can even reduce data dimensions automatically. CNNs are also equipped with transfer learning where pre-trained networks are employed for problem-solving. Transfer learning techniques such as LetNet or AlexNet can drastically reduce the long learning trail typically found in ANNs. All these characteristics have made deep learning an automatic choice for the data scientists.

2.4.1 Tensor

Data used in deep learning is called a *tensor*. The number of axes of a tensor is called its rank. Thus, a tensor having a single number such as a primary variable is called a scalar or a zero-dimensional tensor. A list of numbers is called a vector or a one-dimensional

tensor. An array of vectors is called matrix or two-dimensional tensor. An array of matrices is called a three-dimensional tensor and so on. An array of vectors or rank1 tensor has a shape (samples, features). Similarly, a rank2 tensor or 3D tensor takes a shape like (samples, timesteps, features) such as time-series data. Rank3 or 4D tensor's shape is (samples, height, width, numbers of color channels), e.g. images. Videos are 5D or rank4 tensors with shape (samples, frames, height, width, channel).

2.4.2 Image Processing

In image processing, an image takes the form of a matrix. A convolutional model having an input image array of dimension (h*w) with c channels and *l* layers may be defined as [23]:

$$H\ [i, j, l] = \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} \sum_{c}\ W[a, b, c, l].X[i-a, j-b, c] \quad \text{… Equation 2.4.2.1}$$

In equation 2.4.2.1, X [i, j] and H [i, j] represent pixel location (i, j) of an image and hidden state or activation, respectively; W stands for the weight tensor; a and b are convolutional offsets running all over the input images and belong to the range [-Δ, Δ]. The output shape $[n_h-k_h+1]*[n_w-k_w+1]$ is given by the difference (+1 is the bias) between the input shape $[n_h*n_w]$ and the convolutional kernel shape $[k_h*k_w]$. Then, by considering c_i as input channels and c_o as output channels, the output shape will be $c_i*c_o*[n_h-k_h+1]*[n_w-k_w+1]$ … Equation 2.4.2.2

In a convolution layer, an input array and a kernel/filter array are clubbed together to produce the output. An element-wise multiplication takes place here to yield a Sum of Product (SoP).

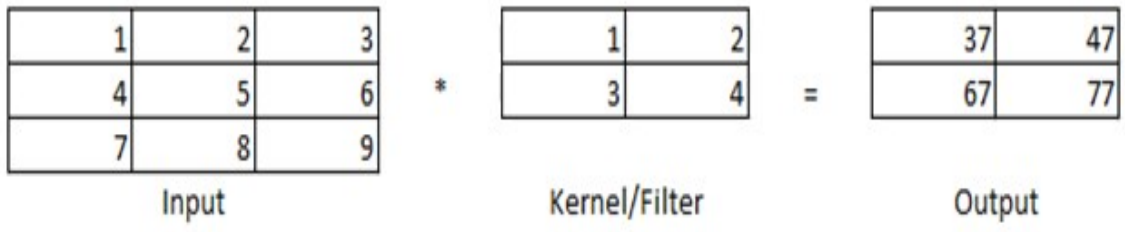


Figure 11: A 2-dimensional convolutional operation

In figure 11, the convolution begins with the top-left corner i.e. coordinate (0, 0) of the input array and it continues sliding from top to bottom and left to right. An element-wise multiplication takes place between the kernel array and the input sub-array of equal dimension. The output is summed up and thus the output array takes the equal shape as

that of the kernel. For example, the first element of the output matrix is calculated as = 1*1+2*2+4*3+5*4=37.

Many times the output array becomes much smaller after a series of convolutional operations. This may cause losing pixels on the boundary regions of input mages and this may also affect the classification task. *Padding* resolves this issue by adding extra pixels having 0 values around the boundary region of the input image. The output shape becomes $[n_h-k_h+p_h+1]*[n_w-k_w+p_w+1]$ where the padding dimension is (p_h, p_w).

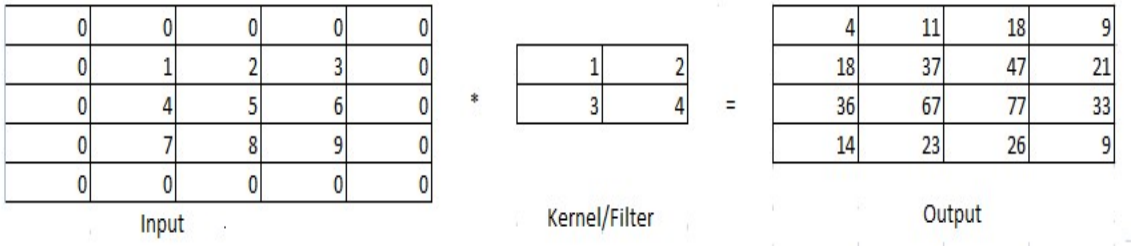


Figure 12: A 2-Dimensional convolutional operation with padding

In figure 12, the first element of the output matrix is determined by summing up the result of element-wise multiplication between the top-left sliding window and the filter array = 0*1+0*2+0*3+1*4=4. The same operation continues and the whole output matrix is formed accordingly.

Another problem arises when the input image is too heavy. It may not be very easy to continue working with such unwieldy input. In these cases, *strides* help to reduce the resolution of the input image. Unlike the normal sliding operation where all the array elements are considered while convoluting, strides skip a few pixels and down-sample the image.

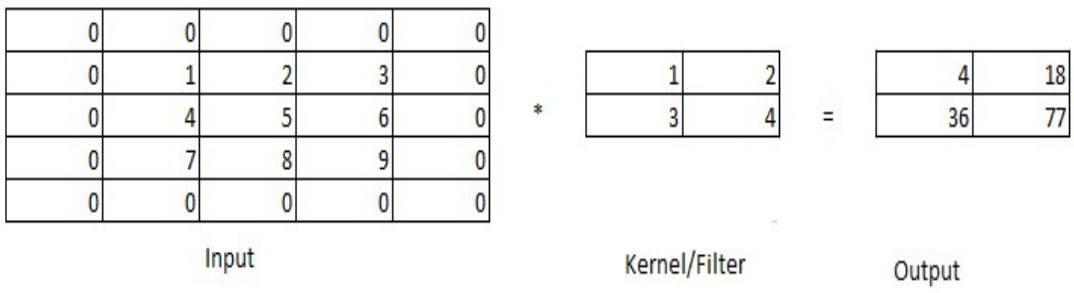


Figure 13: A 2-Dimensional convolutional operation with padding and strides

In figure 13, convolution has started from the top-left corner as usual. A stride of 2 has been used for both the width and the height. Thus, when the second element of the first column has to be decided, two columns are skipped (0*1+4*2+0*3+7*4=36). Similarly, when the second element of the first row has to be decided, two rows are skipped (0*1+0*2+2*3+3*4=18). The typical output shape becomes $[(n_h-k_h+p_h+s_h+1)/s_h]*[(n_w-k_w+p_w+s_w+1)/s_w]$ where s_h is the stride for height and s_w is the stride for width. For multiple input and output channels, the same Sum of Products (SoP) form is being maintained. The output shape with c_i as input and c_o as output channels is $c_i*c_o*[(n_h-k_h+p_h+s_h+1)/s_h]*[(n_w-k_w+p_w+s_w+1)/s_w]$.

When the input passes through convolutional hidden layers, it is down-sampled by using pooling layers. There are mainly two types of pooling layers: *Maxpooling* calculates the maximum and *Average Pooling* calculates an average value from the pooling window, respectively.

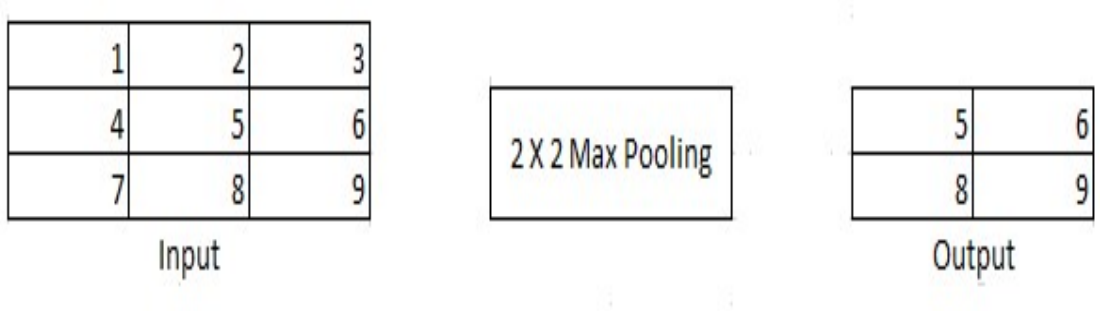


Figure 14: Maximum pooling with a 2X2 pooling window

In figure 14, the first output element calculated as max (1, 2, 4, 5) = 5. This is an important technique of deep neural network as it automatically reduces the dimensionality of the input and keeps important features for the next iterations.

2.5 Recurrent Neural Network

Unlike CNN which does not have a memory, a Recurrent Neural Network (RNN) has a feedback loop that can memorize the earlier incidents. Thus, a recurrent network can learn from its previous iterations and may be defined as:

$$H_t = \emptyset(X_t\, W_{xh} + H_{t-1}\, W_{hh} + b_h) \quad \text{… Equation 2.5.1}$$

In equation 2.5.1, H_t is the hidden state at time t; ø is the activation function; X_t, (t=1, …, T) is the mini-batch of instances with sample size n and d inputs at t^{th} iteration; W_{xh} is the

weight parameter where h is the number of hidden states; H_{t-1} is the hidden state from the previous time-step along with its weight parameter W_{hh}; b_h is the bias parameter.

In equation 2.5.2, O_t depicts the output (q is the number of outputs):

$$O_t = H_t W_{hq} + b_q \quad \text{... Equation 2.5.2}$$

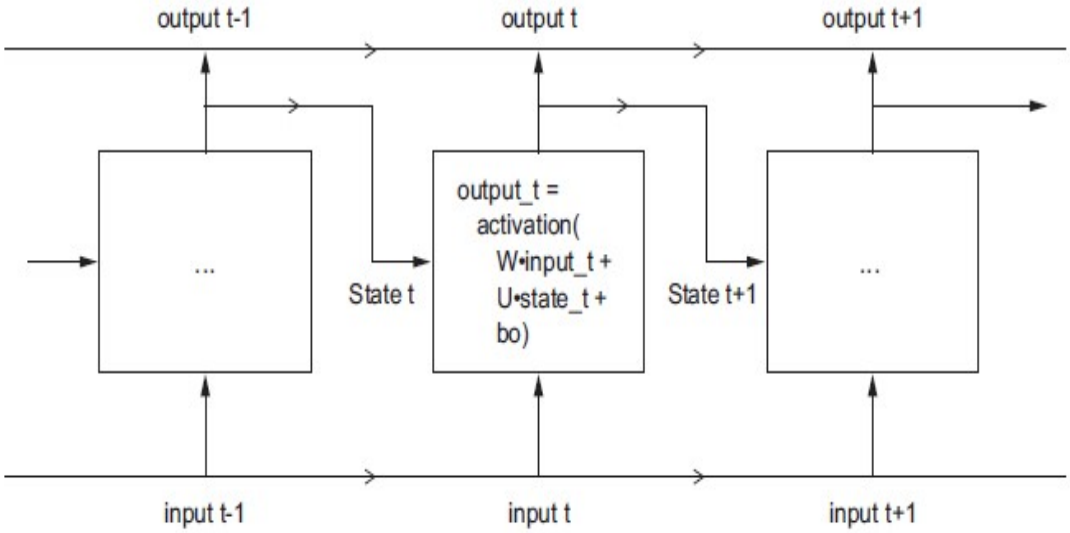


Figure 15: A simple Recurrent Neural Network [https://github.com/fchollet/deep-learning-with-python-notebooks]

In figure 15, each time-step is the output of the loop at time t. Each time-step t in the output tensor contains information about time-steps 0 to t in the input sequence.

2.6 Conclusion

In this chapter, various machine learning techniques have been discussed. Different ways of machine learning are also illustrated. A brief discussion on Artificial Neural Network (ANN) and Deep Learning Network (DNN) has been done. Different varieties of DNN have also been described. From the above discussion, it is evident that a deep neural network's benefit is manifold. The present research work has formulated a novel deep learning model for the classification of malignant tumors. The model has used CNN as the pre-processing technology followed by RNN which learns from the past activity. The present research work has moved one step ahead where the model learns from the previous as well as future incidents. The present research work used both semi-automated and fully-automated methods of classification. In the semi-automated mode, traditional machine learning techniques along with ANN have been used. The DNN model

introduced in this respect is a one-dimensional Convolutional Neural Network (1D CNN). This CNN model is appended with a recurrent network model and this has made the model unique. No other model in the literature has ever used such a hybrid model in tumor classification. In the fully-automated mode, a two-dimensional CNN (2D CNN) have been used. Both sequential and non-sequential options have been exploited. With the sequential CNN model, a bidirectional recurrent model has been used. With the non-sequential model, the same technique has been applied. These models are novel and have rarely been witnessed before. The outcome passes through the activation function and the most likely class has been predicted for every training and validation instances. Finally, the loss has been measured and updated accordingly to optimize it. Thus, the optimal model with minimum loss is developed and used for the prediction of malignant tumors. Once the prediction is done, the results have been evaluated by various metrics. The results have been compared with other models used in the study. The best possible semi-automated and fully-automated options are finalized according to their performances. Both models can challenge any contemporary traditional and advanced models as far as tumor classification is concerned.

3. Review of Literature[1]

Many commendable efforts have so far been made to classify tumors in the human body. The following review of literature covers an era of 2010 to 2019 and attempts to describe how significant studies have contributed in classifying tumors by using different machine learning methods adopted during the specified era. The review includes the nature of work, image modalities employed, classification techniques applied, the dataset used and the performance of the concerned study. A few major types of carcinoma have been included in the review: Lung, Breast, Bladder, Kidney, Liver, Thyroid, Uterus, and Head & Neck.

3.1 Lung Cancer

In 2010, Farag et al. [33] used Linear Discriminant Analysis (LDA) to classify lung nodules. They used the ELCAP database, SURF as the feature descriptor, and PCA as the feature selection tool. The highest accuracy achieved was around 82%. In 2012, Song et al. [34] trained the SVM classifier. They extracted SIFT features from the ELCAP dataset and achieved around 88% accuracy. In 2014, Zhang et al. [35] developed a method for classifying subtypes of a lung nodule. A feature set including SIFT, HOG was extracted from 379 nodules from the ELCAP dataset. SVM and probabilistic Latent Semantic Analysis (pLSA) analysis calculated probabilistic scores. The accuracy achieved was about 89%. In the same year, Kuruvilla & Gunavathi [36] did lung cancer classification by using an Artificial Neural Network. They used CT image and they extracted the Gray Level Co-occurrence Matrix (GLCM) features. The feature selection was accomplished by using Principal Component Analysis (PCA). Their model achieved an average accuracy of 94%. In 2015, Devinder Kumar et al. [37] did Lung Nodule Classification by using Deep Features. They used CT Images from LIDC (TCIA) dataset. The accuracy they achieved was near 75%. The system was developed to categorize a candidate tumor as either benign or malignant. During 2017, QingZeng Song et al. [38] studied Deep Learning for Classification of Lung Nodules. They used CT images from LIDC-IDRI (TCIA) database and the accuracy was around 84%. In the same year, Atsushi Teramoto et al. [39] classified lung cancer subtypes from cytological images using deep neural

[1] Based on author's publication no. 3 [Appendix C] and no. 8, 9 [Appendix D]

networks. The study achieved an accuracy of 71%. In 2018, Tafadzwa Lawrence Chaunzwa et al. [40] used deep-learning radiomics to predict lung cancer histology. LASSO method was applied to CT images and obtained an accuracy of around 75%. In the same year, Patrice Monkam et al. [41] studied CNN models discriminating between pulmonary micro-nodules and non-nodules. They used CT images from the LIDC-IDRI dataset. The system achieved around 88% of accuracy. Manu Sharma et al. [42] also used the LIDC-IDRI dataset for early detection of lung cancer during 2018. They used CT images and achieved an accuracy nearing 84%. In 2018 only, Mao et al. [43] used a deep auto-encoder for transforming local patches into local features and subsequently generated global features. The best accuracy attained was about 95%. In the same year, Liu et al. [44] trained CNNs for lung nodule classification. For training and testing, 1738 nodules and 1000 non-nodules were extracted from LIDC-IDRI. Furthermore, 421 samples were collected from the ELCAP dataset as test cases. The method had an overall accuracy rate of 92.3% on LIDC-IDRI and 90.3% on ELCAP. In 2018 only, Shaffie et al. [45] also conducted a study on deep learning-based classification of lung nodules. They used computed tomography scans from the Lung Image Database Consortium. In 2019, Bhatia et al. [46] did lung cancer detection by using Deep Learning Residual Approach (DLRA). The accuracy achieved was 84% on LIDC-IRDI dataset.

3.2 Breast Cancer

In 2010, Ayer T et al. [47] classified breast cancer by using an ANN model. They used mammographic and demographic data of 62219 patients with 10-fold cross-validation. The accuracy achieved was around 97%. In 2012, George et al. [48] applied Vector Quantization (LVQ) on microscopic images to classify breast tumors and achieved around 88% accuracy. In the same year, Swathi et al. [49] used Wisconsin Breast Cancer Data (WBCD) to train different neural network structures including Radial Basis Function (RBF), General Regression Neural Network (GRNN), Probabilistic Neural Network (PNN), Multilayer Perceptron (MLP) model and Back-Propagation Neural Network (BPNN). From the comparative study, it was found that the BPNN gave the best diagnostic performance of 99.28%. In 2013, Eshlaghy A et al. [50] used Support Vector Machines (SVM) with 547 clinical data and the accuracy achieved was 95%. In the same year, Azar & El-Said [51] applied three classification algorithms namely, Multi-Layer

Perceptron (MLP), Radial Basis Function (RBF), and Probabilistic Neural Networks (PNN) to classify breast cancer. They used WBCD and achieved 96% accuracy. In 2014, Seema Singh et al. [52] also used the WBCD database and investigated two variants of the back-propagation algorithm with three and four layers of neural networks. They achieved an accuracy of around 97%. During 2015, B. Singh et al [53] evaluated performances of some popular feature normalization techniques on breast tumor classification using ultrasound images. They used SVM and MLP with gradient descent or back-propagation and achieved an accuracy of about 84%. In 2015, Onan [54] employed Fuzzy-Rough Nearest Neighbor (FRNN) on mammograms and achieved 99% accuracy. During 2016, Huynh et al. [55] used transfer learning (CNN) to classify breast tumors as benign or malignant. They used 219 images, and the achieved AUC was 0.86. In 2017, Dhungel et al. [56] also employed CNN for mass detection. They used the public INbreast dataset containing 410 multiview images. The accuracy achieved was 90% and the sensitivity was 98%. In 2018, Xu et al. [57] used CNN and INbreast dataset to estimate breast density and the accuracy achieved was approximately 93%. Their classification endeavor was limited to benign and malignant tumor categorization.

3.3 Bladder Cancer

In 2015, Wang et al. [58] showed that Extreme Learning Machine (ELM) and Regularized ELM had more 0.8 sensitivity and specificity in predicting the mortality rate of bladder cancer patients after a radical cystectomy. In 2017, Xu et al. [59] did the tumor classification based on multiparametric MRI radiomic features for accurate differentiation between NMIBC and MIBC and got an AUC of 0.8610. In 2018, Ikeda et al. [60] trained CNN to distinguish between bladder tumors and healthy urothelium. They used a pre-trained network with 1.2 million images from the ImageNet data set. The outcome was 93.0% sensitivity and 83.7% specificity. In the same year, Eminaga et al. [61] used 18681 images from 479 cystoscopy videos. Their Xception-based model achieved an accuracy score of 99.52% while classifying the cystoscopic images. In another study of bladder cancer prediction during 2018, Cha et al. [62] investigated 123 subjects, and the AUC obtained was 0.8, whereas the assessment of doctors had an AUC of 0.74. In 2019, Shkolyar et al. [63] used deep CNNs (TUMNet) to detect bladder papillary tumors from cystoscopic videos of 100 patients. The work was reported with

90% sensitivity. In the same year, Zheng et al. [64] extracted radiomic features and used the radiomic-clinical nomogram to serve as an auxiliary tool for bladder cancer classification. Their experiments with the validation set showed an AUC of 0.876. In 2019 only, Lin et al. [65] integrated radiomics and transcriptomics to predict the progression-free interval (PFI) of bladder cancer patients. The radiomics risk model achieved an AUC of 0.956 and the transcriptomics risk model achieved an AUC of 0.948.

3.4 Renal Cancer

In 2017, Ing et al. [66] used a 2-step machine learning framework for quantitative imaging of tumor vasculature to get a prognostic gene signature. They also used the Cancer Genome Atlas (TCGA) data. Quantification of the vascular area helps in retrieving of 9 vascular attributes that predicted disease-free-survival. Two linear models with regularization bifurcated independent groups having 301 cases into good and poor disease-free survival groups. The AUC achieved was about 0.79. In 2018, Ali et al. [67] did a kidney cancer subtype classification using genome data. Features were extracted by Neighbourhood Component Analysis (NCA), and Long Short Term Memory (LSTM) was used to classify a given miRNA sample into kidney cancer subtypes. The Cancer Genome Atlas data repository (TCGA) was used, and the accuracy achieved was around 95%. In the same year, Bektas et al. [68] used machine learning-based CT texture analysis for prediction of Fuhrman Nuclear Grade of Clear Cell Renal Cell Carcinoma. The classifiers used were Support Vector Machines (SVM), Multi-Layer Perceptron (MLP), Naïve Bayes, k-Nearest Neighbors, and Random Forest. It was observed that SVM attained the highest predictive performance and correctly classified approximately 85% of nuclear grades with an AUC of 0.86. During 2019, many major developments took place in the field of automated classification of renal cancers. Han et al. [69] did a renal cancer subtype classification by using the deep learning method. A dataset was built from 169 renal cancer cases. The study showed around 85% of accuracy and 0.9 AUC. Zhou et al. [70] employed a deep learning model pre-trained by ImageNet dataset to classify renal tumors into benign and malignant. 192 cases (CT images) were used and the highest accuracy achieved was 93%. Tabibu et al. [71] used CNN to predict renal cancer histology subtypes and survival outcomes. The accuracy achieved was about 94%. Tian et al. [72] did the grading of renal carcinoma. The Cancer Genome Atlas (TCGA)

was used as the data repository. The Lasso model applied on 26 features, classified grade with 84.6% sensitivity, and 81.3% specificity in the test set. Kocak et al. [73] used machine learning-based CT texture analysis in predicting the mutation status of the gene encoding in renal cell carcinoma. The two machine learning models used were ANN and Random Forest. Random Forest achieved a better AUC of 0.98 than ANN, which achieved an AUC of 0.92. Rajendran et al. [74] did a computer-assisted radiologic assessment for the identification of tumor roughness as a method to predict renal cancer subtypes during 2019.

3.5 Liver Cancer

In 2011, Preis et al. [75] detected metastatic liver malignancy by using deep learning along with PET/CT images and ended up with an accuracy of 90.5%. In 2015, Kim et al. (76) detected liver cancer with an accuracy of 99.19 % by using a neural network and 98.19 % by using a fuzzy neural network. In 2017, Vivanti et al. [77] tried to detect new liver tumors by using deep learning and CT images. The study included 246 tumors of which 97 were new tumors and the accuracy achieved was 86%. In 2018, Sabut et al. [78] did liver cancer subtype classification by using 225 CT images. After tumor segmentation by Gaussian Mixture Model and the watershed model, various texture features were extracted and fed into a Deep Neural Network (DNN) classifier. The study claimed to achieve a classification accuracy of 99.38%, after 200 epochs of training. In the same year, Ben-Cohen et al. [79] proposed CNN for the detection of liver metastasis in CT examinations of the liver. The method involved a convolutional neural network based classification. A total of 20 subjects having a total of 68 lesions formed the training set. For validation purposes, 14 cases with an overall 55 lesions were included. The true positive rate was 94.6% with 2.9 false positive per case. In 2018 only, Bharti et al. [80] proposed a model to differentiate between chronic liver and cirrhosis and the presence of Hepato-Cellular Carcinoma (HCC). The system was based on higher-order features. The proposed CNN feature was able to differentiate four liver stages and the classification accuracy of the model was 96.6%. During 2018, Frid-Adar et al. [81] used medical images for improving the performance of CNN based medical image classification. They used limited data of CT images of 182 liver lesions for generating synthetic medical images by employing Generative Adversarial Networks (GANs). They compared

classification performance with synthetic data augmentation and classic data augmentation. The classification performance yielded 78.6% sensitivity and 88.4% specificity. In the same year, Sato et al. [82] predicted hepatocellular carcinoma using real-world data containing 539 and 1043 patients with and without HCC. The gradient boosting attained the highest predictive accuracy of 87.34% and an AUC of 0.94. In 2019 only, Romero et al. [83] adopted an end-to-end approach of deep learning incorporating feature extraction of the residual connections. Fully connected layers with pre-trained ImageNet weights were integrated as a classifier to provide a probabilistic output of liver lesion type. The accuracy achieved was 96%.

3.6 Thyroid Cancer

In 2014, Ding et al. [84] employed a Support Vector Machines (SVM) on the thyroid B-mode ultrasound images to classify the lesion. The accuracy achieved was about 97%. In 2016, Hongxun et al. [85] did a study to construct classifier models for differentiating malignant from benign thyroid nodules by including 970 patients reviewed by radiologists and nodules were graded according to a five-tier scoring system. Radiologists' diagnosis achieved an accuracy of 88.66%, and the radial basis function (RBF)–Neural Network (NN) achieved a sensitivity of 92.31%. In the same year, Razia & Rao [86] reviewed the performances of machine learning techniques for thyroid disease detection. They discussed the contribution of different neural network modeling techniques in identifying hypothyroid diseases over the past two decades. They observed that the Linear Discriminant Analysis (LDA) algorithm gave the highest accuracy of 99.62% with 6-fold cross-validation. In 2017, Torab-Miandoab et al. [87] applied image enhancement, image segmentation, and feature extraction to determine cold thyroid nodules automatically with 99% accuracy. In 2018, Farihah et al. [88] evaluated the consistency of the ultrasound classification system in predicting thyroid malignancy. A total of 91 patients were eligible and correlation of the classification with pathology results was assessed. The accuracy achieved was 93%. In 2019, Park et al. [89] developed an ultrasound-based deep learning model for the prognosis of thyroid nodules and compared its performance with those of a Support Vector Machines (SVM)-based model and radiologists' diagnosis. Images of 4919 thyroid nodules were used and the diagnostic performance was evaluated by using logistic regression. It showed overall comparable

diagnostic performance with radiologists and assessed thyroid nodules more effectively without loss of sensitivity. In the same year, Wang et al. [90] classified histological subtypes of thyroid tumors by using 11,715 fragmented images. Using the test set, VGG-19 yielded a better average diagnostic accuracy of 97%, whereas, the Inception-ResNet-v2 achieved an accuracy of 94% for all the malignant types. In 2019 only, Zhang et al. [91] used a random forest algorithm to diagnose thyroid nodules. The model performed better (AUC = 0.924) than radiologist diagnosis (AUC=0.834) on conventional ultrasound.

3.7 Uterine Cancer

Long back in 2002, Clark et al. [92] had reviewed 56 studies on endometrial cancer. Heterogeneity of diagnostic performance was found significant and was not explained by either study setting or study quality. In 2014, Tseng et al. [93] used SVM to predict the recurrence of cervical cancer. They used clinical data of 168 patients and achieved an accuracy of 68%. In the same year, Patil et al. [94] extracted calcification features and used ANN & SVM to diagnose uterine cancer. Although the study claimed to increase the accuracy level, it lacked proper utilization of evaluation metrics. In 2018, Malek et al. [95] proposed a computer-assisted method for distinguishing uterine sarcoma from leiomyomas by using perfusion-weighted magnetic resonance imaging. They used data of 42 patients and obtained an accuracy of around 92%. In 2019, Sun et al. [96] developed a CADx by using CNN and attention mechanisms (HIEnet). They predicted subtypes of the endometrial tumor with an accuracy of about 84%. In the same year, Akter et al. [97] performed machine learning analysis using 38 RNA-sequence and 80 enrichment-based DNA methylation datasets. Several biomarker genes were recognized by machine-learning experiments. They employed several machine learning techniques like SVM, Random Forest, etc. They recommended a generalized linear model for feature space reduction and classification performance maximization. In the same year, Santhi et al. [98] used a deep convolutional neural network for malignancy detection in uterine cancer. Contour extracted images showed an accuracy of 92.14% in a 5-class problem. In a review during 2019, Njoku et al. [99] observed that tissue specimens are limited by the invasiveness and unacceptability of current sampling techniques. They concluded that

studies developing valuable biomarkers for endometrial cancer detection should utilize the potential of endometrial fluids sampled using non-invasive methodologies.

3.8 Head & Neck Cancer

In 2017, Halicek et al. [100] developed a CNN classifier to detect head and neck cancer by using Hyperspectral imaging. The classifier was able to differentiate between normal and cancer affected tissue with 81% accuracy. In the same year, Ma et al. [101] proposed another CNN based model. The model could distinguish between normal and cancerous tissues with an average accuracy of 91%. They also used Hyperspectral imaging, but the test subjects were 12 tumor-bearing mice. In 2018, Gupta & Malhi [102] used 26019 CT scan images from TCIA and employed a deep learning framework to detect head & neck tumors. They used fuzzy c means image segmentation technique and extracted Gray Level Co-Occurrence Matrix (GLCM) features. Their claimed accuracy was 98.8%, although the method they followed was not fully automated. In the same year, Lo et al. [103] claimed to have 100% accuracy in predicting metastasis of a malignant tumor. They proposed a Support Vector Machines (SVM) based method called Pred-Meta. The data set used in the study had 75 samples from 70 patients of head and neck cancer provided by the Taipei Veterans General Hospital of Taiwan. Nevertheless, they did not demonstrate enough statistical evidence to support their claim. Li et al. [104] trained Artificial Neural Network, K Nearest Neighbor, and Support Vector Machine to detect recurrence patterns in nasopharyngeal carcinoma with 306 subjects. The best performing model was an artificial neural network with an AUC score of 0.812. In 2019, Halicek et al. [105] did a study for differentiating head & neck squamous cell carcinoma from normal tissues by using CNN. They considered 156 cases and achieved an AUC of 0.916. In the same year, Diamant et al. [106] used deep learning to predict clinical outcome of head & neck cancer. The treatment outcome was predicted by using CNN. A total of 300 cases from TCIA were considered and the AUC achieved was 0.92. Ma et al. [107] used adaptive deep learning for differentiating between benign and malignant head & neck tumors. An auto-encoder network was trained and the model had achieved a sensitivity of 92.32% and a specificity of 91.31%. However, the study carried out was an animal-based one.

3.9 Research Gap

From the above review it has been observed that, on a considerable number of occasions, in-vivo or non-invasive approaches have successfully challenged the in-vitro or invasive diagnosis of tumors. Machine learning, especially, deep learning has emerged as a seminal tool for CAD-based tumor prognosis. Researchers have so far concentrated on the classification of any particular type of tumor according to the originating organ [108] . This has indeed elevated the performance of domain-specific classification of tumors; the efforts related to the classification of heterogeneous tumors have been rarely witnessed. Most of the studies have ended up with a binary classification of tumors and tried to figure out whether the tumor was malignant or benign. These binary classification tasks have yielded higher accuracy; the methods are yet to be tested against a multiclass problem such as AJCC staging. The initiative to automate the TNM staging or histopathological grading has not been frequently seen. Although the histological subtypes of the tumor are being identified in several cases, those cases are also based on some particular organ. It has also been observed that studies done earlier were based on some single mode of imaging, in most of the cases; these were based on the CT scan. Studies based on multimodal imaging techniques are very less in number. In many cases, the data are not clinically endorsed and the sample size is also insignificant. Even in the case of deep learning, many researchers have used the transfer learning technique [109] which needs GPU or cloud computing support. Consequently, these studies gave birth to huge resource consuming architecture. On the other hand, a few worked with orthodox machine learning techniques [110]. In today's context, those standard methods may hardly find themselves useful in the concerned field of study. Thus, there is ample opportunity to carry out a study that can address all these issues. Such a study should be heterogeneous in terms of imaging modalities, tumor types should also be diversified concerning the originating organs. The data should be authentic and the algorithm to be used should be an improved one as per the performance which may be evaluated by various statistical metrics as applicable. The proposed architecture should also be lightweight than its contemporaries and should be less constrained by computing resources. The present study bridges the research gaps found in the related studies and

hopefully will come up as a new emerging scientific model for automated detection of malignant tumors of different origins.

3.10 Conclusion

In this chapter, an extensive review of literature has been done on existing works carried out so far with different types of tumors. As eight different types of tumors have been considered in the study, each of the tumor types has been taken for a tumor category. Brief description and outcome of every notable work for the last ten years have been discussed per tumor category. The review does not only narrate past work but also depicts their respective strength and limitations. This implies that the discussion regarding the related contemporary works acts loosely as a SWOT (Strength Weakness Opportunity Threat) analysis. The strength of the existing works is the success within their limited circumference. Weakness is the limitation where these works lack in pursuing experiments in a bigger periphery. Opportunity is the gap created as a scope for present and future studies. The threat is the inadequate advancement in the utilization of genomic biomarkers in tumor prognosis. From the review, the overall research gaps have been identified. These research gaps direct the present research work in a meaningful way by fine-tuning the objectives and points the target activities more precisely. This in turn has increased the overall efficacy of the present study.

4. Image Acquisition[2]

Medical image biomarkers of cancer have improved patient care through customized healthcare practices [163]. Biological imagery features are more advantageous than genomic biomarkers. A non-invasive procedure can characterize a heterogeneous tumor effectively [164]. On the other hand, in the invasive process, only limited tissue remains available for biopsy. The Cancer Genome Atlas (TCGA) data is mainly focused on mapping cancer phenotypes to genotypes. It provides clinical images matched to subjects from The Cancer Genome Atlas (TCGA). The Genomic Data Commons (GDC) data portal stores the clinical data while The Cancer Imaging Archive (TCIA) stores the radiological data [12]. In TCGA, case identifiers allow researchers to explore databases for finding correlated genotype, phenotype, and treatment outcomes. Tissues for TCGA are collected from different sites across the world. The heterogeneous image data sets vary in terms of scanner modalities, and acquisition protocols. In most cases, the images are acquired while carrying out routine care. It does not follow a controlled research study. This makes the dataset independent of any predetermined bias or correlation. A significant contribution from different institutions to the archive creates multi-institutional data sets and they become an open resource for researchers. As per the methodology adopted by TCGA, the comprehensive quantification of tumor phenotypes are done by applying a large number of quantitative image features. Features quantifying tumor image intensity, shape, and texture are extracted. A large number of radiomic features have been re-invented as having significant prognostic power in these independent data sets. Analysis has revealed that a predictive radiomic signature is linked with underlying gene-expression patterns. These suggest that radiomics identifies a general prognostic phenotype in various types of cancers. A significant scientific impact may be observed as imaging is normally used in clinical practice. This provides an unprecedented opportunity to improve the decision-support system which may help in cheaper cancer treatment.

In this research work, eight types of cancers have been included for classification: Urothelial Bladder Carcinoma, Cervical renal papillary cell carcinoma, Liver Hepatocellular Carcinoma, Non-Small Cell Lung Cancer, Uterine Corpus Endometrial

[2] Based on author's publication no. 9 [Appendix D]

Carcinoma, Thyroid Cancer, head and neck squamous cell carcinoma (HNSCC), and Breast Invasive Carcinoma. Thus, eight different datasets from The Cancer Imaging Archive (TCIA) have been considered for malignant tumor classification. Each of these datasets represents one particular type of tumor based on its originating organ. In this way, eight different originating organs have been covered: prostate, kidney, liver, lung, uterus, thyroid glands, head & neck, and breast.

4.1 TCGA-BLCA

TCGA-BLCA dataset represents Urothelial Bladder Carcinoma (BLCA) which is the most common type of bladder cancer and one of the leading causes of cancer-related death worldwide. The dataset was created by Ken Clark, last modified by Quasar Jarosz on Mar 30, 2020 [111]. The dataset comprises 111,781 images of 120 numbers of patients. Major imaging modalities are Computed Tomography (CT), Magnetic Resonance (MR), Computed Radiography (CR), Positron Emission Tomography (PET), and Digital Radiography (DX).

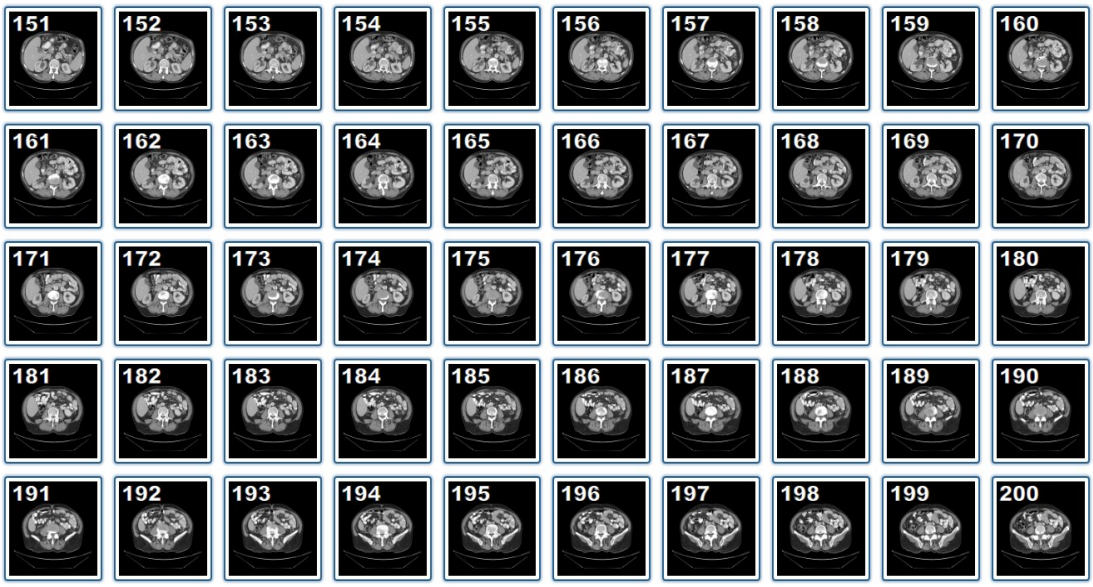


Figure 16: Glimpse of the image set from TCGA-BLCA (patient barcode: TCGA-2F-A9KO)

The following institutions have contributed towards the database development:

- University of North Carolina
- Barretos Cancer Hospital, Brazil
- University of Chicago

- University of Sheffield
- Memorial Sloan-Kettering Cancer Center, New York
- Lahey Hospital & Medical Center, Burlington
- University of Southern California

4.2 TCGA-KIRP

TCGA-KIRP depicts cervical renal papillary cell carcinoma which is one of the most common kidney-related cancers. It was created by Justin Kirby and was last modified by Quasar Jarosz on Mar 30, 2020 [112]. It has 33 cases consisting of 376 series and 26,667 images. Major imaging modalities are CT, MR, and PT.

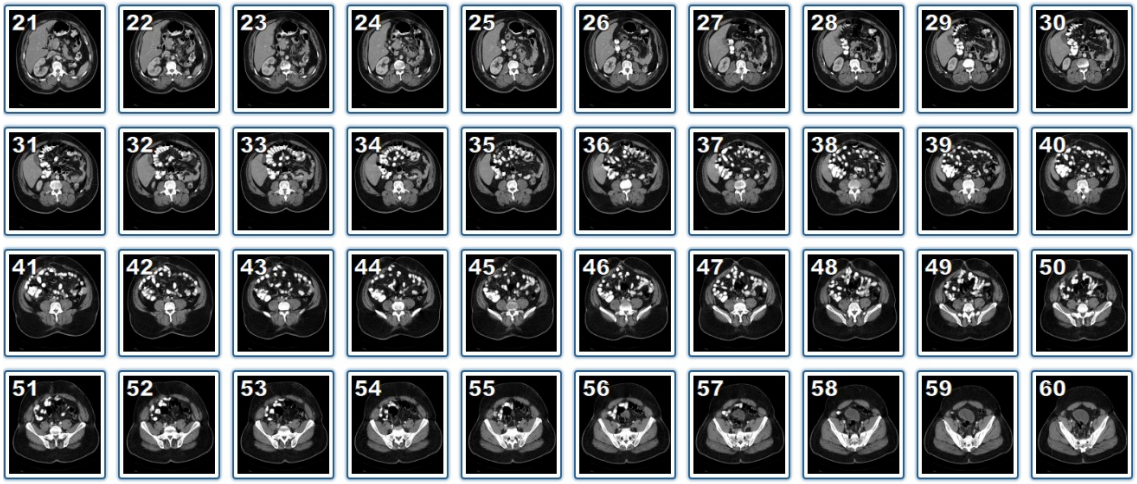


Figure 17: Glimpse of TCGA-KIRP image dataset (patient barcode: TCGA-B9-4113)

The following individuals have provided data for this collection:

- National Cancer Institute, Bethesda
- University of North Carolina
- Roswell Park Cancer Institute, New York
- Lahey Hospital & Medical Center, Burlington

4.3 TCGA-LIHC

TCGA-LIHC is the Liver Hepatocellular Carcinoma (LIHC) image dataset. It was created by Justin Kirby and was last modified by Tracy Nolan on Jan 08, 2020 [113]. 97 cases have been incorporated here. The total number of series is 1688 having a total number of 125,397 images. Major imaging modalities are CT, MR, and PT.

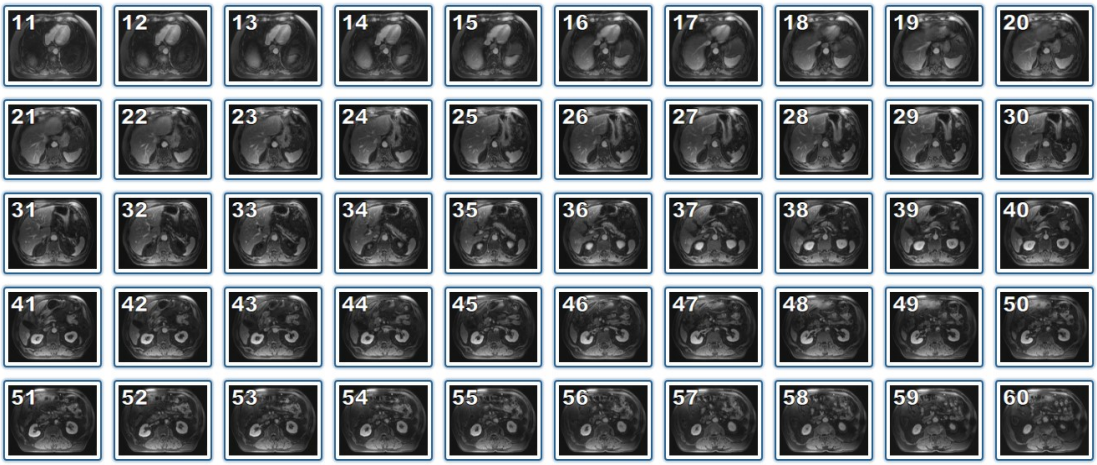


Figure 18: Glimpse of TCGA-LIHC Images (patient barcode: TCGA-BC-4073)

The following institutions have contributed data for this collection:

- Mayo Clinic, Rochester, MN
- The University of North Carolina,
- Alberta Health Services
- Lahey Hospital & Medical Center, Burlington

4.4 NSCLC Radiogenomics

It is a well-known radiogenomic dataset having 211 subjects of Non-Small Cell Lung Cancer (NSCLC) [114]. It was created by Kirk Smith and was last modified by Justin Kirby on Apr 01, 2020. The dataset comprises different imaging modalities like Computed Tomography (CT), Positron Emission Tomography (PET)/CT images. It also contains semantic observations and segmentation maps of tumors. Quantitative imaging data are mapped with clinical data, including pathological stage or grade of tumor, mean or maximum uptake values (in case of PET), treatment strategy, and survival outcomes [115]. It also depicts the demographic details of patients such as gender, age, smoking habits, location, etc. This dataset also helps in finding the inherent bond between genomic and medical image attributes. In this way, it facilitates the development of predictive biological features of medical images. Eminent doctors from Stanford University have constituted the database.

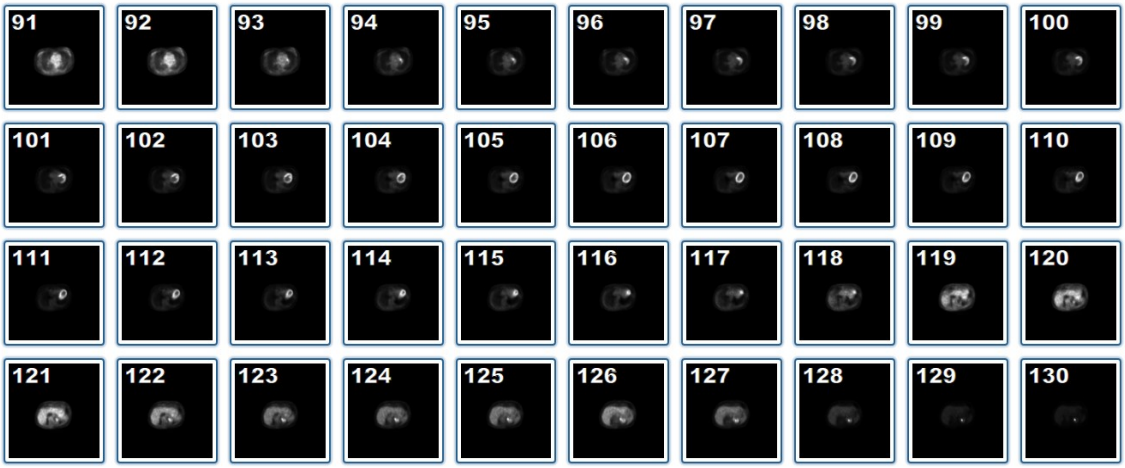


Figure 19: TCIA NSCLC Radiogenomics (Case ID R01-001)

4.5 TCGA-THCA

TCGA-THCA represents thyroid cancer which is a common sub-type of head and neck cancer. It was created by Ken Clark and was last modified by Quasar Jarosz on Mar 30, 2020. There are 6 cases in the image set with 28 series and 2,780 numbers of images [161]. Major imaging modalities are CT and PET.

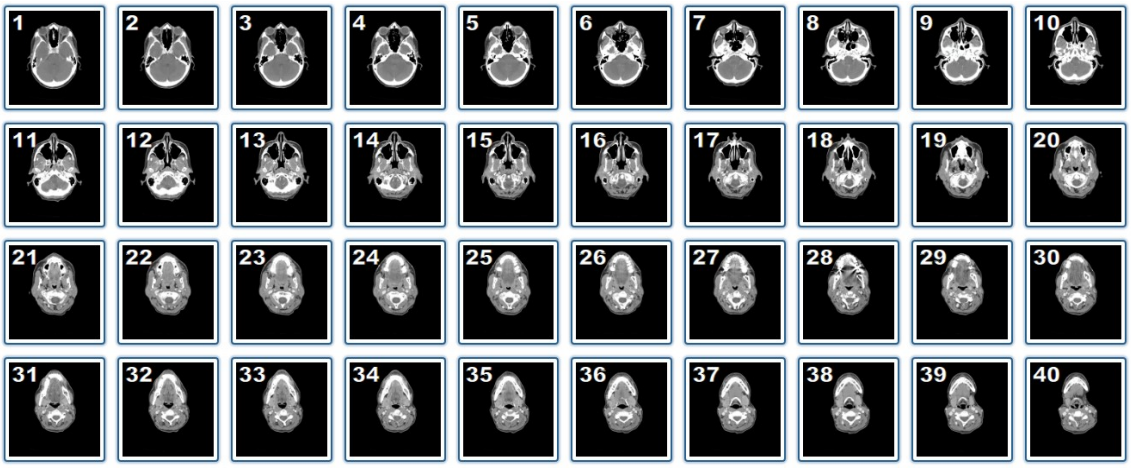


Figure 20: Glimpse of TCGA-THCA (patient barcode: TCGA-DE-A4MA)

The following institutions/individuals have provided data for this collection:

- The University of North Carolina,
- Roswell Park Cancer Institute, New York

4.6 TCGA-UCEC

TCGA-UCEC represents the Uterine Corpus Endometrial Carcinoma. It was created by Ken Clark and was last modified by Quasar Jarosz on Mar 30, 2020 [162]. There are 65 cases including 912 series having a total of 75,829 numbers of images. Major imaging modalities are CT, CR, MR, and PT.

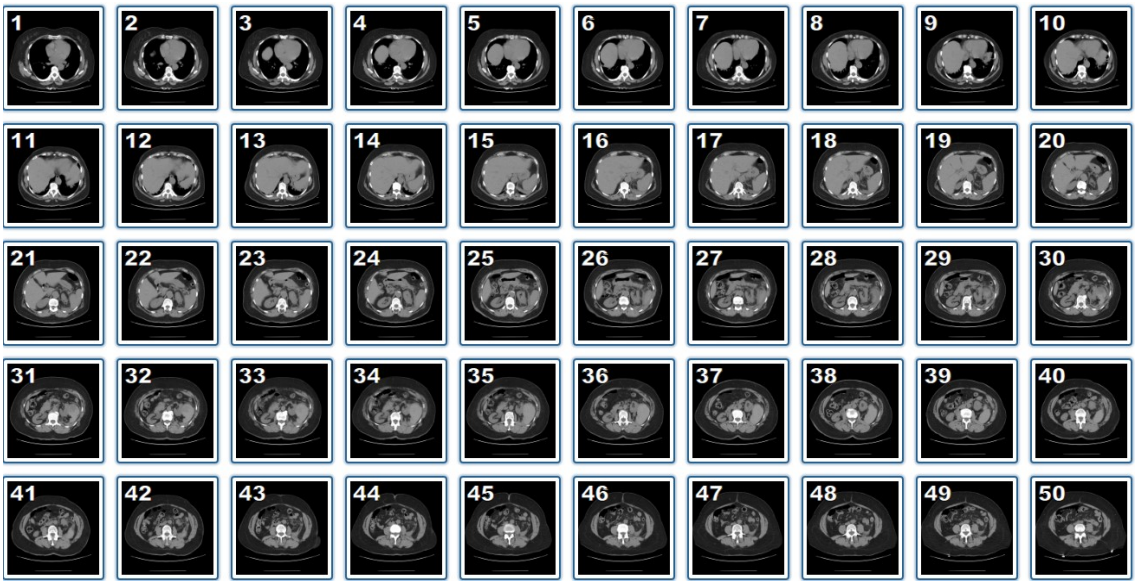


Figure 21: Glimpse of TCGA-UCEC (patient barcode: TCGA-D1-A0ZO)

The following institutions/individuals have provided data for this collection:

- Mayo Clinic, Rochester
- Washington University
- MD Anderson Cancer Center, Houston

4.7 Head-Neck-Radiomics HN1

Radiomics is the complete numerical representation of the tumor's observed characteristics by applying a large number of quantitative imagery attributes [116]. Clinical data of 137 head and neck squamous cell carcinoma (HNSCC) patients have been clubbed together to form the Head-Neck radiomics collection [117]. These patients were treated by radiotherapy. The main imaging modality is computed tomography (CT). It was created by Geri Blake and was last modified by Michelle Tacconelli on Apr 01, 2020. This dataset is provided as an open repository to support the research in radiomics.

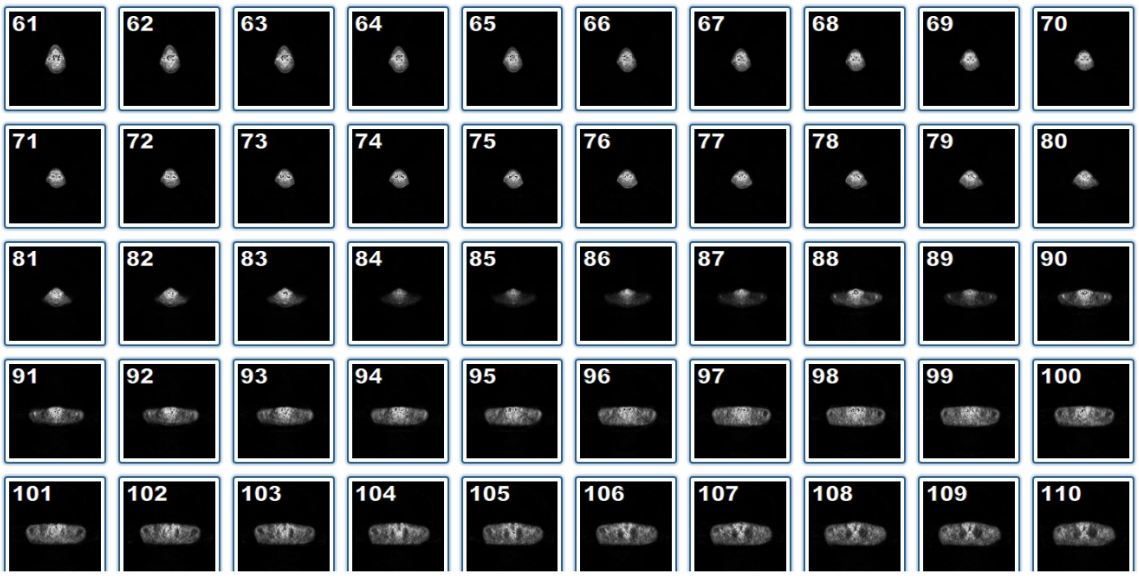


Figure 22: Glimpse of Head and Neck HN1 radiomics (Case ID: HN1004)

The following institutions have provided data for this collection:

- Maastricht University Medical Centre, the Netherlands
- Dana-Farber Cancer Institute, USA
- Harvard Medical School, USA.

4.8 TCGA-BRCA

TCGA-BRCA represents Breast Invasive Carcinoma. The dataset was created by Justin Kirby and was last modified by Quasar Jarosz on Mar 30, 2020 [118]. It has 164 cases with 1877 series containing a total of 230, 167 images. Imaging modalities are MR and mammography (MG).

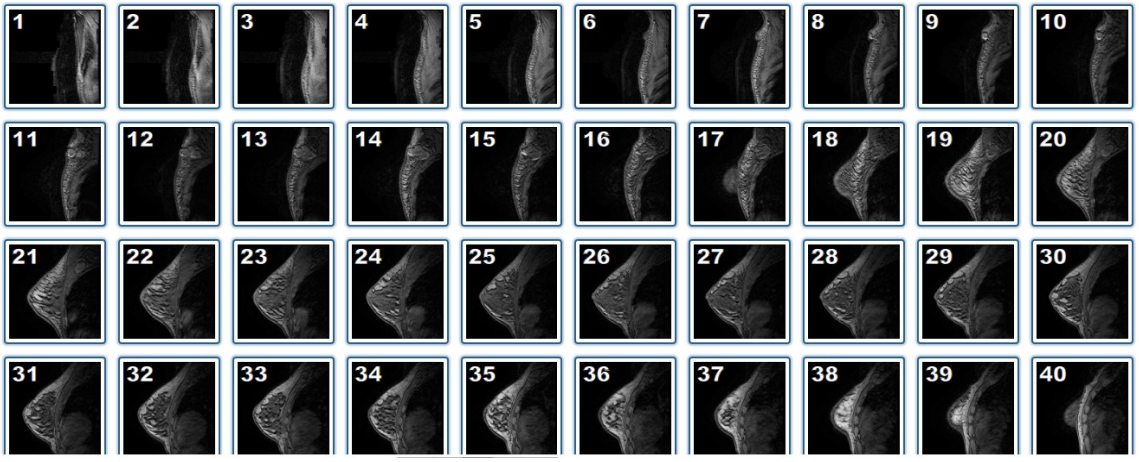


Figure 23: Glimpse of TCGA-BRCA (patient barcode: TCGA-EW-A1PC)

The following institutions have provided data for this collection:

- Mayo Clinic, Rochester
- University of Pittsburgh
- University of Miami
- Memorial Sloan-Kettering Cancer Center, New York
- University of North Carolina
- University of Chicago, Chicago
- Roswell Park Cancer Institute, New York

4.9. Final Image Collection

The final image acquisition has been carried out by retrieving images from all the aforementioned collections. Each subject has a large set of pre-surgical DICOM images stored in the respective database. Each case is identified with a unique Patient ID. Each patient's demographic, clinical, pathological, and/or genomic data are stored in TCGA. In the present research work, the most important features are the TNM staging, AJCC staging, and histological grading embedded in the clinical data against the respective patient. These features have been used as the class or target variable while carrying out the classification task. In some cases, the histological grades or the pathological stage are missing. Those cases are discarded during the final database preparation. Thirty best scans from each case having pathological data from all the eight collections have been taken to form the final image collection. In this way, 717 cases where the supportive clinical and pathological data are present have been considered and from each such case, thirty best scans were extracted. Thus, 21,510 radiological images have been collected to form the new image dataset. This newly prepared image collection is heterogeneous in terms of imaging modality, cancer type, cancer stage/grade, and patients' demographic characteristics.

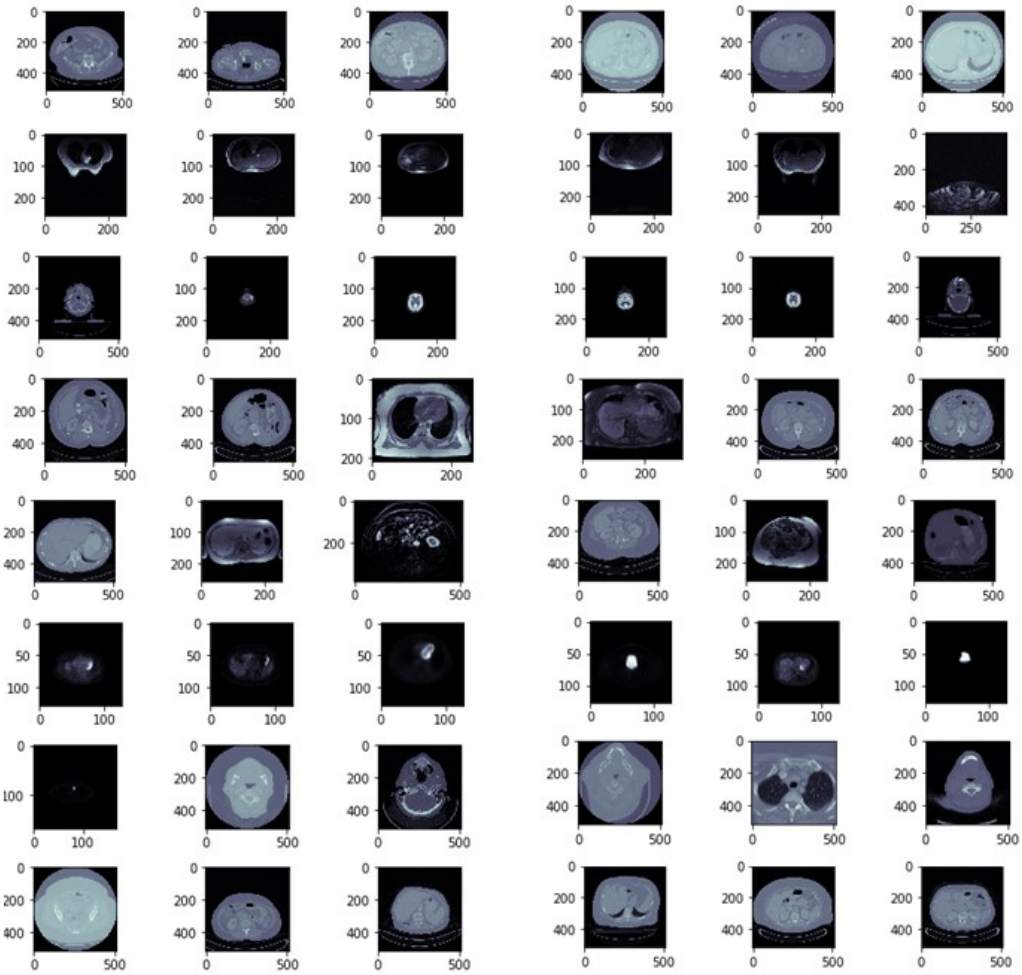


Figure 24: Glimpse of the final image collection (1st row represents TCGA-BRCA, 2nd row represents Head & Neck Radiomics, 3rd row represents TCGA-BLCA, 4th row represents TCGA-KIRP, 5th row represents TCGA-LIHC, 6th row represents NSCLC radiogenomics, 7th row represents TCGA-THCA and 8th row represents TCGA-UCEC)

4.10 Conclusion

In this chapter, the process of developing an image collection of the heterogeneous source has been described. Images from eight different open databases have been collected. Each database represents one particular type of tumor depending upon its originating organ. The cases where clinical information regarding tumor stage and grade are available to have been considered for the final image database preparation. Twenty best scans have been retrieved from each prospective case and clubbed together as a single image collection where each image has a case ID. This has made the new image repository a diversified mix of modalities containing details of patients of different demography. This has indeed brought novelty in the present research work.

5. Image Segmentation[3]

Image segmentation is the process of separating and grouping similar pixels of an image to extract the desired Region of Interest (RoI) [119]. Segmentation is a necessary step in the traditional image processing curriculum. If accurately done, segmentation does not only help in identifying tumor location and size but also helps in formulating treatment strategies [120]. In chemotherapy or radiation treatment, often the non-cancerous cells adjacent to cancerous cells got damaged due to improper manual demarcation of tumors. Proper segmentation decreases the chance of damaging non-cancerous cells by an exact delineation of the tumor. Thus, segmentation plays an important role both in automated tumor detection and in treatment strategy formulation. Such image segmentation comes in two flavors: implicit and explicit segmentation. In the orthodox machine learning process, explicit segmentation has to be done and manually crafted features have to be used for classification. The scenario has changed with the advent of advanced technologies such as deep learning, where implicit segmentation takes place followed by automatic feature generation.

5.1 Segmentation Methods

From the existing review of the literature [121], it has been observed that different existing segmentation techniques may be broadly categorized as thresholding based, region-based, boundary-based, stochastic & learning-based, and texture analysis based segmentation.

5.1.1 Thresholding Based Segmentation

Thresholding is a simple segmentation technique where all pixels of an image above a certain threshold value will be considered as foreground and the rest of the pixels are considered as background [122]. In this way, grayscale images become binary images and the Region of Interests are fetched. Thresholding comes in different varieties. The simplest form of thresholding is fixed thresholding. Here, all the values above a certain intensity level are added to a group. The rest of the values are considered background. The threshold value may be determined manually or as a result of earlier training and analysis. For example, a popular variety of PET is ^{18}F-FDG. Here radioactive glucose is

[3] Based on author's publication no. 6 and 7 [Appendix D]

injected in the patient's body and uptake values are measured through a PET scanner. For a malignant tumor, as the rate of cell division will be too high, the uptake values will also be high. In PET image segmentation often Standardized Uptake Value (SUV_{max}) is considered as the threshold value. Such manual threshold value may get affected by Partial Volume Effect (PVE), motion artifacts, or resolution related problems. To overcome these limitations, adaptive thresholding was proposed. In the case of adaptive thresholding, many image quality metrics are considered while selecting the threshold value. A few of the image quality parameters are Source-to-Background Ratio (SBR), Mean Intensity Value (MIV), etc. Unlike adaptive thresholding, iterative thresholding determines the optimal threshold value without any prior assumption of the volumetric metrics.

5.1.2 Region-based Segmentation

Region-based segmentation methods mainly use the homogeneity of the image intensities to determine the object boundaries. Two popular varieties of region-based segmentation are region grow and graph-based methods. In the region grow method, pixels connected to a user given seed region are either included or excluded based on the mean pixel intensities and standard deviations. In the graph-based method, both foreground and background based seeds are used to yield the segmentation outcome.

5.1.3 Boundary-based Segmentation

Boundary based methods do not use the statistics obtained from the image, instead, they identify the object boundaries. This is also known as deformable model-based segmentation [123]. These methods may be categorized in two varieties: active contour and gradient-based methods. While performing active contour, an initial contour is deformed around the desired Region of Interest (RoI) and gradually moves towards the edge of the intended object. In gradient or spectral based segmentation, gradient or the intensity difference is calculated between a voxel and its neighboring voxels. This process detects edges of the image as intensity difference will be high along the boundary of an image.

5.1.4 Stochastic and Learning-based Segmentation

Stochastic methods use the statistical difference between uptake regions and adjacent cells. Similarly, the learning-based method use pattern recognition techniques to classify pixel data. One of the leading stochastic and learning-based models is mixture models [124]. In the Gaussian mixture model, it is assumed that any image may be approximated by the summation of Gaussian densities. Expectation-Maximization is an optimization technique by using which the Gaussian densities may be separated. Another popular method is the Fuzzy Locally Adaptive Bayesian (FLAB) segmentation technique. It is an unsupervised method having two hard tissue classes and a finite class of fuzzy membership levels.

5.1.5 Texture Analysis-based Methods

One of the notable segmentation techniques is based on the texture analysis of images. It follows two different approaches: Intensity Volume Histogram (IVH) and morphological feature-based method. There are six metrics used in the IVH method: I10, I90, I10-90, V10, V90, and V10-90. Morphological features have two varieties: Geometrical Shape-Based (GSB) and Gray Level Co-occurrence Matrix (GLCM) or texture-based. There are four shape-based features: eccentricity, Euler number, solidity, and extent. Texture features are also four in number: contrast, local homogeneity, energy, and entropy.

5.2 Segmentation - a Comparative Study

From the existing review of the literature [125], it has been observed that selection methods of segmentation techniques have been often done arbitrarily in contemporary research papers. A few researchers have tried to compare different segmentation techniques, but those studies are tumor-specific. Thus, there is a need to carry out a study to evaluate various major segmentation techniques for different kinds of tumors.

5.2.1 Objective

The objective of the present study is to benchmark the performance of prime segmentation techniques by applying different kinds of tumors found in different parts of the human body.

5.2.2 Methodology

Six types of tumors, namely, Head & Neck, Kidney, Urinary Bladder, Lung, Thyroid, and Breast tumors are considered for the study. Data of 601 patients have been extracted from datasets Head-Neck Radiomics, TCGA-KIRP, TCGA-BLCA, TCGA-THCA, NSCLC Radiogenomics, and TCGA-BRCA, respectively. Twenty scans for each of the cases are used in the study. The image collection is non-uniform in terms of scanner modalities, subject demography, cancer stage, and treatment policy. The following segmentation techniques are used: Region Growing (region-based), Active Contour (deformable model-based), Watershed (gradient-based), Texture based (morphological) [126], and Fuzzy Clustering Means (learning-based) [127]. The accuracy of the segmentation result was measured by the Dice Similarity Coefficient (DSC):

$$DSC(V1, V2) = 2*(|V1 \& V2|) / (|V1| + |V2|) \quad \text{…Equation 5.2.2.1}$$

Where V1 is the segmented volume and V2 is the actual volume. DSC is the ratio of sensitivity (true positive rate) and specificity (true negative rate). Each segmentation algorithm is executed against all the data from each category of tumors and the average DSC value has been recorded subsequently. All the algorithms were implemented in Matlab R2015a [128].

5.2.3 Result and Discussion

All the algorithms have been codified and run with the selected imagery collection. Different segmentation techniques have revealed different results when applied to a particular type of tumor image set. A few algorithms have performed consistently, whereas a few algorithms have performed well in some particular cases. The performance was measured by considering the DSC score for the respective true and segmented volumes for all the images belonging to each type of tumor as generated by each segmentation method.

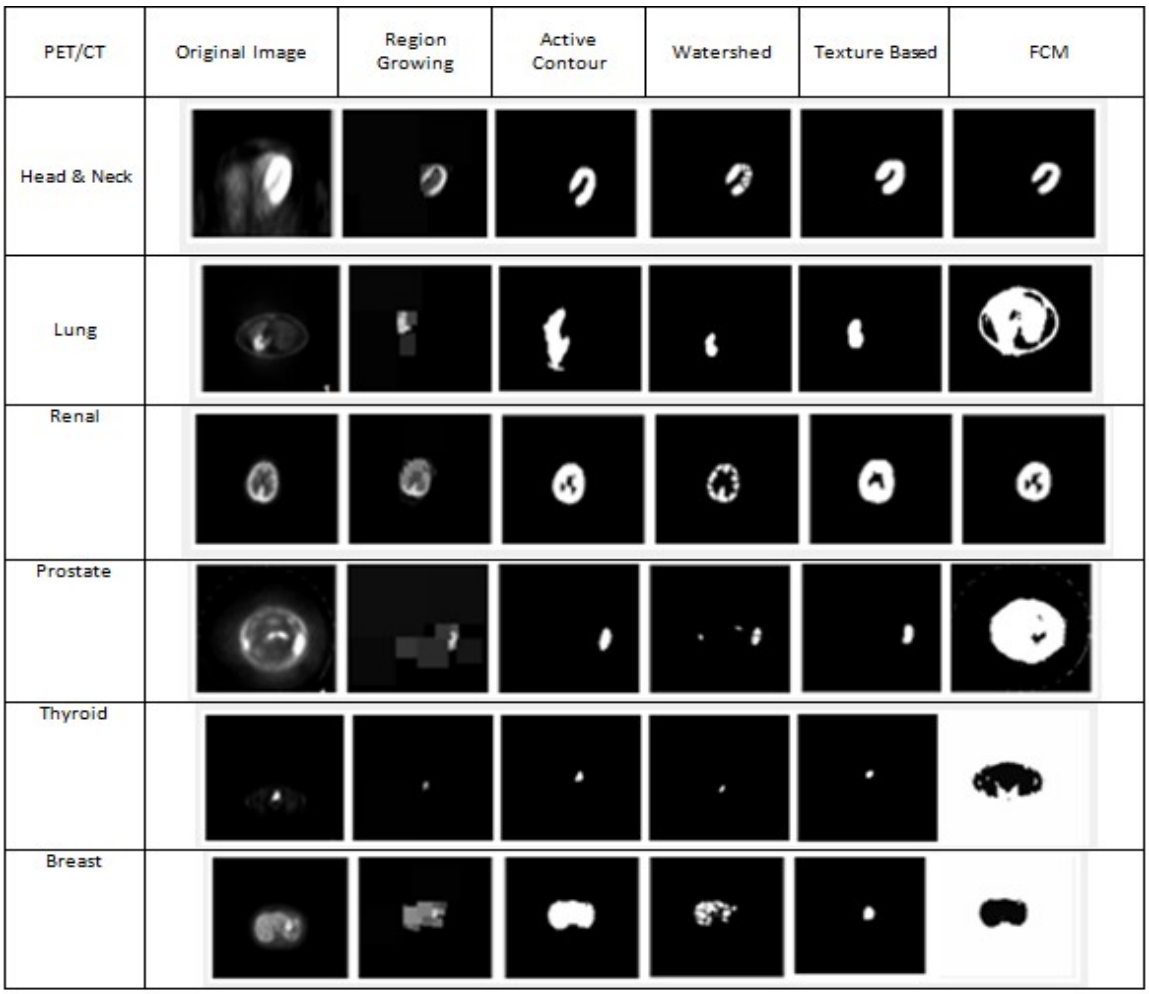


Figure 25: the result of the comparison of segmentation techniques

From figure 25 it may be observed that all the segmentation techniques except the Region Growing method have done a pretty good job in segmenting Head & Neck tumors. Active Contour, Texture, and FCM methods have given good results for renal tumors. Texture, Watershed, and Active Contour methods have performed commendably for Prostate, Thyroid, and Lung tumors. For Breast Tumors, the results of Watershed and Texture techniques are quite promising.

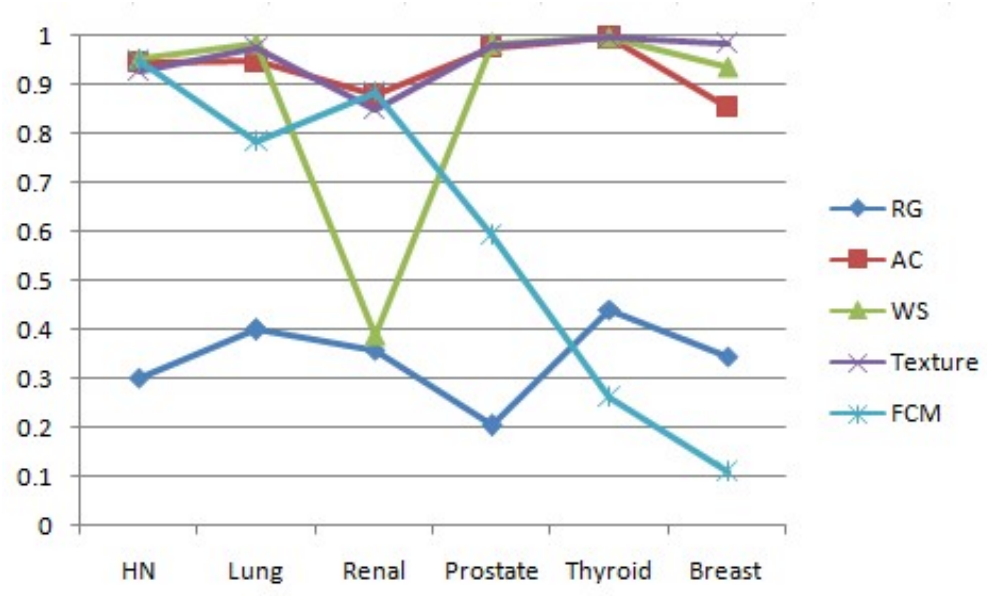


Figure 26: Graphical comparison of DSC values (RG-.> Region Growing, AC-> Active Contour , WS->Watershed, Texture->Texture Analysis, FCM-> Fuzzy Clustering Means

From Figure 26, it is evident that the texture-based segmentation technique has been a consistent performer for almost every type of tumor under consideration. It is also found that Active Contour and Watershed Segmentation techniques are useful in some cases. The performances of other techniques are not very consistent. Although in some cases a few techniques have done well, e.g. FCM is quite effective while segmenting Head & Neck or Renal tumors. In fine, it may be concluded that the texture genre of segmentation is quite handy as far as the automatic segmentation of tumors in the human body is concerned.

5.3 Segmentation Strategy

In the present research work, a holistic image segmentation strategy has been adopted. Often it is seen that the result of the segmentation is poor in resolution, or the alignment of the objects is not proper. Being a histogram oriented method, thresholding based segmentation often fumbles in providing the complete spatial significance of the candidate image. Region-based technique becomes inefficient when there is any anatomical defect, e.g., leakage in the boundary region of a tumor. Active contour often depends on the interactive demarcation of the tumor and the gradient-based approach may not work properly if the resolution of the image is not good. In such cases,

registration and reconstruction of the image may create more processing overheads. Based on the result of the comparative study, the current research work relies on the texture-based segmentation after preprocessing the target image set.

5.3.1 RGB Image Thresholding

The difference between grayscale and an RGB image lies in the number of colour channels present. A binary or a grayscale image is typically a single-channel image, whereas, an RGB image is a three-channel image. By turning off the second and third channel values, the red channel image is obtained. Subsequently, we may also obtain a grayscale image by using a red channel. This may turn out to be handy in image segmentation. The algorithm for RGB thresholding is as follows:

Step 1: Visit every channel of the concerned image;

Step 2: Store the channel value in a temporary variable;

Step 3: Set all the pixel values to zero whenever they are less than the threshold value;

Step 4: Store the value of the temporary variable in the original channel;

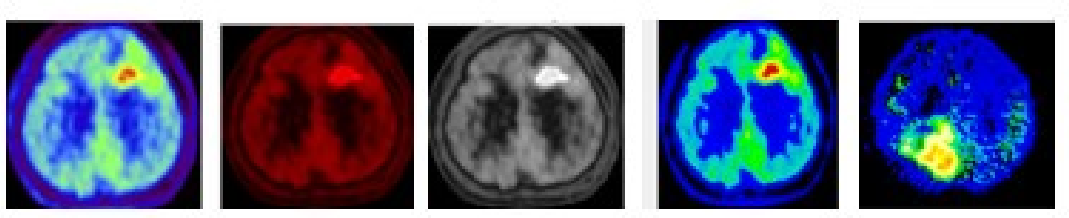


Figure 27: PET Image True Color Segmentation and Reconstruction (Original Image, Red Channel Image, Red Channel Grayscale Image, RGB Thresholded Image of the original images

5.3.2 Grayscale Image Thresholding

It is a common practice to convert the image into a grayscale image having a standard size that is suitable for processing and causes less memory and processing overhead. If any algorithm is true for a [n X n] matrix, then it is also true for an [n X n X k] matrix. The image is then sharpened by using the histogram equalization method. After that, the Gaussian blurring technique is used to remove the noise present. Then comes a very important step called *edge detection*. This edge detection phase helps in segmentation by detecting the boundary region of the image. At last, the flood-filling operation is carried

out to fill in the holes present in the image. The threshold value may be determined by using Otsu's method [129] or its improved version. The equation for thresholding may be written as:

$$O = (I>t) - (G) \quad \text{…Equation 5.3.2.1}$$

Where I is the input image, t is the threshold limit and G is the grayscale equivalent of I, and O is the output image. Here, I, t, and G are comprised of real values.

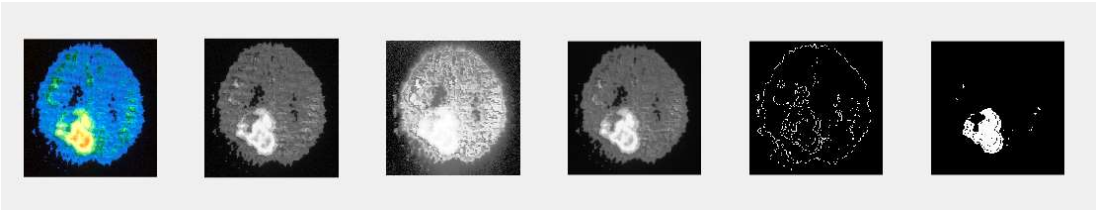


Figure 28: PET Image Grayscale Segmentation (Original Image, Grayscale Image, Enhanced Image, Filtered Image, Edge Detection, and Thresholded Image)

Fields	Area	MajorAxisLength	MinorAxisLength	Perimeter
17	71	19.4808	12.7999	90.8640
18	2	2.3094	1.1547	1.9600
19	8	3.6515	3.0551	7.2200
20	8	3.6515	3.0551	7.2200
21	3999	87.1589	74.9276	535.6300
22	2	2.3094	1.1547	1.9600
23	5	4.7441	2.6533	9.3620

Figure 29: Extracted Features from Segmented PET Image

The features extracted from the segmented images (Figure 29) are quite useful for the present research work [130]. The extracted features may help in the determination of exact tumor size and location and turn, it will also help to detect the stage and the growth rate of the tumor. Thus, such automated segmentation or reconstruction process may help to overcome the difficulties of manual segmentation. The job of radiologists will be much easier when such tools help them as a supportive decision-making system.

5.4 Segmentation Workflow

In the present research work, segmentation has been a manifold activity. First images are converted to grayscale and resized to [256X256] and then fine-tuned by following steps like blurring, sharpening, etc. as discussed in the preceding section. These pre-processing steps help in reducing the memory load-related overheads while conducting succeeding

experiments. While segmenting images, first the threshold value has been determined by following Otsu's histogram equalizing method (Figure 30).

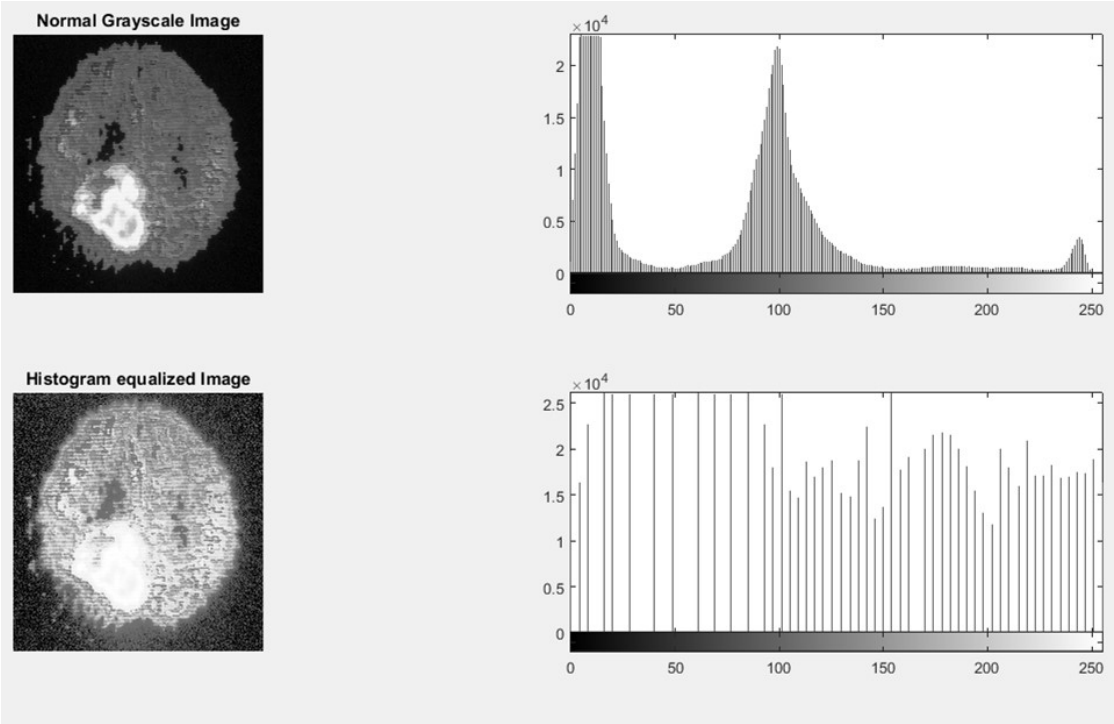


Figure 30: Normal grayscale image and histogram equalized image

It has been observed that segmentation done by using Otsu's method is much effective than segmentation done by using any arbitrary threshold value (Figure 31).

Figure 31: Original DICOM image of the tumor and segmented tumor image

As threshold-based segmentation faces problems when there is a deformity in the anatomical structure of the tumor, the present research work first determines the

threshold value by Otsu's method and then adopts a texture-based segmentation method. The algorithm is as follows:

Step 1: convert image to binary

Step 2: convert numeric to logical

Step 3: fetch region properties

Step 4: store region properties separately

Step 5: calculate and store area

Step 6: select density > threshold value

Step 7: store area having a maximum selected density

Step 8: store tumor class having an area equal to the area with max density;

Step 9: group tumor pixels where tumor label is equal to stored tumor class

Step 10: create a structuring element

Step 11: dilated to recover original dimensions

Step 12: show image of type array

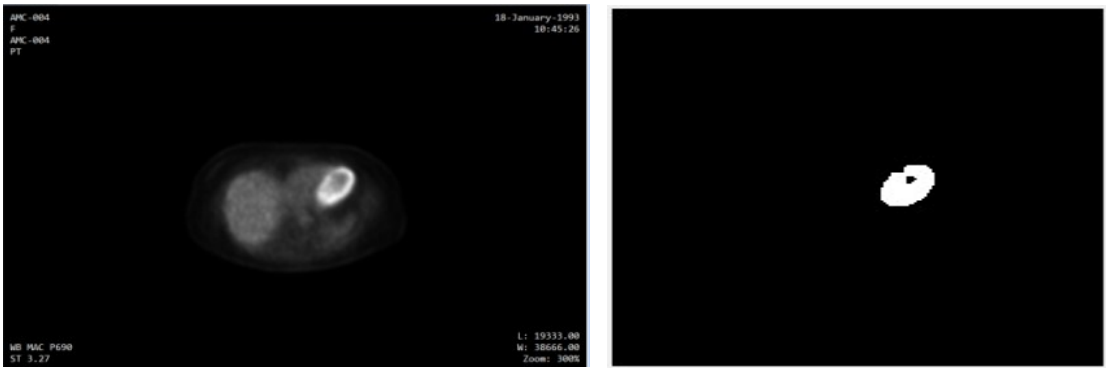


Figure 32: Sample base image and its texture-based segmentation (magnified version)

5.5 Conclusion

In this chapter, the process followed for image segmentation has been described in detail. Different categories of segmentation techniques have been described. A comparative study has also been conducted to find out the best possible technique for the problem at hand. An overall segmentation strategy has also been formulated. This includes different pre-processing steps, procedures for threshold value calculation, and final segmentation technique. Proper segmentation makes feature extraction easier by depicting a specific Region of Interest (RoI).

6. Feature Extraction[4]

The segmentation process is followed by the feature extraction phase. The present research work has primarily used a combination of two feature detectors: the Maximally Stable Extremal Regions (MSER) [131] which is used to detect blob features and then the Speeded Up Robust Features (SURF) [132] which is used to extract the scale-independent features suitable for classification.

6.1 Maximally Stable Extermal Regions (MSER)

Let $Q_1, \ldots, Q_i, \ldots\infty$ be a sequence of nested extremal regions ($Q_{i-1} \subset Q_i$). Extremal region Q_{i*} is maximally stable iff $q(i) = |Q_{i+\Delta} \backslash Q_{i-\Delta}|/|Q_i|$ has a local minima at i*. The extermal regions R_i are detected as follows [133]:

$\forall p \in R_i, \forall q \in \text{boundary}(R_i) \Rightarrow I_{in}(p) >= I_{in}(q)$ …Equation 6.1.1

Where I_{in} is the input image, ***iff*** means ***if and only if***. Here, $Q_1, \ldots, Q_i, \ldots\infty$ is an infinite sequence (ordered set) of nested extremal regions that belong to or forming the outer structure of any image or its parts.

In $|\mathbf{Q_{i+\Delta} \backslash Q_{i-\Delta}}|/|\mathbf{Q_i}|$, $Q_{i+\Delta} \backslash Q_{i-\Delta}$ => the relative complement of $Q_{i+\Delta}$ and $Q_{i-\Delta}$

$|Q_{i+\Delta} \backslash Q_{i-\Delta}|$ and $|Q_i|$ => cardinality of the expression within

$\mathbf{|Q_{i+\Delta} \backslash Q_{i-\Delta}|/|Q_i|}$ => ratio of two absolute values

6.2 Speeded Up Robust Features (SURF)

The sum of the original image within a rectangle can be evaluated quickly using the integral image, $S(x,y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(x,y)$ …Equation 6.2.1

A Hessian matrix H (p, σ) at point p and scale σ, is determined to obtain the point of interest:

$$H(p, \sigma) = \begin{bmatrix} Lxx(p,\sigma) & Lxy(p,\sigma) \\ Lyx(p,\sigma) & Lyy(p,\sigma) \end{bmatrix}$$ …Equation 6.2.2

The lowest level of the scale space is obtained as:

σ_{approx} = current filter size X (base filter scale/base filter size) … Equation 6.2.3

Then, the Gaussian second-order derivative with a box filter is approximated and later, the maxima of the determinant of the Hessian matrix is extracted as candidate interest

[4]Based on author's publication no. 1 and no. 5[Appendix D]

points by using equations 6.2.2 and 6.2.3:

$\text{Det}(H_{approx}) = D_{xx}D_{yy} - (\beta D_{xy})^2$ …Equation 6.2.4

While computing the determinant of the Hessian matrix, the term β compensates for the error caused by the approximation of the true Gaussian derivative masks. An analogous computational form, performed over the entries of a matrix $\sqrt{\sum_{i,j} a_{ij}^2}$ is called the Frobenius norm. The multiplicative factor indicated in the equation 6.2.4 defines the ratio of Frobenius norms between the box filter approximations and their true peers. It has been observed that β may practically be approximated using a static constant 0.9 even though β is not independent of the scale size σ.

6.3 MSER-SURF Methodology

The combined MSER-SURF algorithm may be written as [134]:

Step 1: De-noised image used as input;

Step 2: Extermal regions are detected using MSER (by using equation 6.1.1);

Step 3: Candidate interest points are determined using SURF (by using equation 6.2.4);

Step 4: Finally feature vectors are extracted and merged accordingly:

$\overrightarrow{F_{MSER-SURF}} = \{\overrightarrow{F_{MSER}}, \overrightarrow{F_{SURF}}\}$ … Equation 6.3.1

Equation 6.3.1 implies that SURF descriptors are extracted at locations identified by the MSER detector. Upon analyzing the algorithm, it may be observed that at first the MSER detector incrementally walked through the intensity range of the concerned image to detect stable regions (equation 6.1.1), and then the SURF points of interest were detected from those regions (equation 6.2.4). From each concerned image 12 MSER regions were detected (Figure 33) and a sample of the structure of such a region is as follows: [Count, Location, Orientation, PixelList] = (1, 44.5362, 12.7516,-1.4958, 69*2 int32). This is a typical MSER region structure culled out of an image. The structure consists of a *Count* (no. of stored regions; type integer), *Location* (locations of ellipses, stored as an *M*-by-2 array of [*x y*] coordinates), and *Orientation* (ellipse orientation, stored as a value in the range from -pi/2 to + pi/2 radians. This value represents the orientation of the ellipse as measured from the *X*-axis to the major axis of the ellipse) and *PixelList* (point coordinates

for detected MSER regions, specified as an *M*-by-1 cell array. Each cell is an array of (x, y) coordinates of dimension p-by-2, where p is the number of pixels in an MSER region detected; data type is a 32-bit integer). From these regions, SURF features are retrieved. A neighborhood of size 20s was taken around each of valid SURF points where s is the window size. Each region is divided into 4x4 sub-regions. For each sub-region, horizontal and vertical wavelet-based responses are recorded. This when represented as a vector gave SURF feature descriptor with a total of 64 dimensions. Features are given arbitrary names f_i, i=1 to 64.

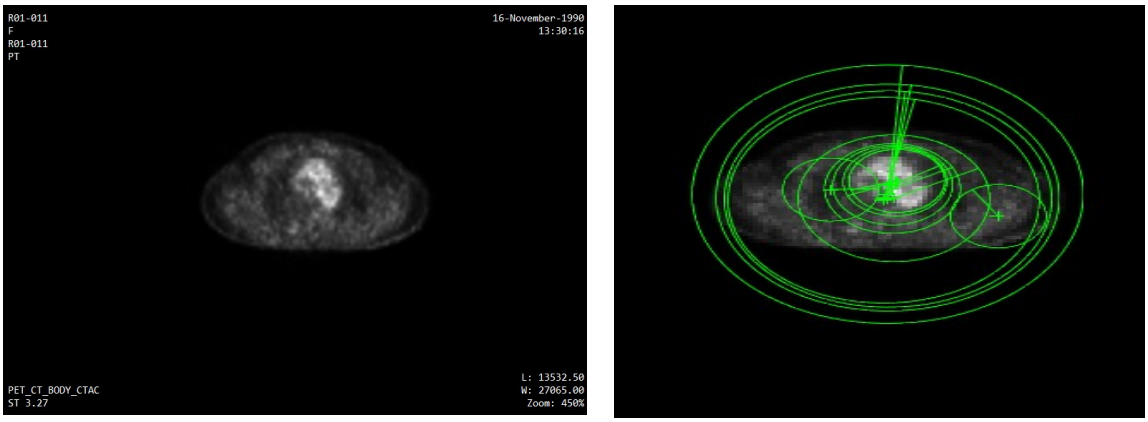


Figure 33: (a) base image (b) image with MSER regions and SURF points detected on it

f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	f17	f18	f19	f20
0.578774	0.341802	0.907862	0.946885	-0.61278	-0.1207	-0.07568	0.224436	0.875566	0.654432	0.407034	0.502841	2.105008	2.93241	1.869117	2.517305	-0.65664	1.032748	0.784053	0.897851
-0.10001	-0.63247	-0.05705	0.209487	1.134052	0.136475	0.704956	0.026082	-0.39125	0.148589	0.200012	0.027693	0.323856	-0.32468	-0.02525	-0.01429	0.019264	-0.08957	-0.21139	-0.1889
0.467441	-0.48853	0.03694	-0.1898	-0.25864	-0.19164	-0.43828	-0.29827	0.28435	0.880797	0.085575	0.721569	-0.22172	0.28327	-0.14555	-0.00686	-0.02017	-0.70893	-0.30386	-0.38381
-0.03201	0.065766	-0.37844	-0.37765	1.99261	2.399467	1.726054	2.221893	-0.7808	1.93831	1.300851	1.828556	-0.98552	0.581482	0.656336	0.269106	0.682647	0.949762	0.415279	0.052271
2.260805	1.390013	1.75832	0.90859	-0.52302	-0.40589	-0.08704	-0.31317	0.085178	-0.40092	-0.13928	-0.54452	-0.58586	0.140184	0.730167	-0.02735	-0.67807	-0.55947	0.611199	0.961426
-0.11817	-0.02085	-0.51321	-0.53583	-0.09623	1.392135	-0.22594	1.242198	-0.455	0.254093	-0.15339	0.05523	0.101633	-0.19568	-0.48503	-0.50075	1.341887	-0.29707	1.062489	0.361841
0.120809	0.351224	-0.29281	-0.15961	-0.63115	-0.14608	-0.6443	-0.25884	-0.00088	-0.45556	-0.59862	-0.68703	0.068654	-0.1915	-0.4462	-0.4636	0.157769	-0.34948	-0.13343	-0.47265
-0.10911	-0.05356	-0.50164	-0.56412	-0.72499	-0.69468	-0.86456	-0.79484	0.10987	-0.66651	-0.71928	-0.89464	0.121453	-0.21239	-0.48076	-0.49344	-0.45482	-0.30236	-0.72242	-0.68209
-0.12479	-0.18875	-0.46834	-0.48078	0.49783	-0.09496	0.144751	-0.19492	0.962377	2.110545	0.873837	2.009904	-1.02731	1.36939	0.70172	0.968683	-0.43797	-0.06619	-0.71423	-0.6924
-0.02093	0.384362	-0.39535	-0.11667	2.596569	1.111917	2.294274	0.967282	-1.65684	1.669529	1.06216	1.545554	-0.5212	0.181743	0.16886	-0.14682	0.003229	-0.23554	-0.28811	0.119497
-0.64425	-0.57761	1.301347	0.934338	1.523284	0.900813	1.291683	1.173476	-0.5252	0.028622	0.360416	0.229731	0.036153	-0.65659	0.638576	0.17524	-0.77184	1.147281	0.080386	0.562724
0.541997	1.932077	3.243887	2.385588	0.936639	0.496033	0.606426	0.506562	2.030926	-0.75438	3.399623	0.362257	1.437835	-3.09231	1.601881	2.379514	0.848948	-0.74682	2.038314	1.165616
0.010111	0.355443	-0.03487	0.140942	1.72442	0.77964	1.206458	0.720517	-0.33095	0.039493	0.309576	0.072036	-0.2481	-0.58393	-0.02289	0.202156	0.507489	-0.2757	0.585233	0.045416
-0.13257	-0.03658	-0.52649	-0.55173	2.421225	0.694337	3.114052	0.558719	0.957306	0.270918	1.02765	0.32021	-3.52191	0.600876	3.340439	1.016578	0.080707	-0.10457	-0.21126	-0.51541
0.073657	-0.47893	-0.2862	-0.17836	-0.80335	-0.75063	-0.72632	-0.69894	0.088762	-0.50723	-0.55737	-0.72914	0.409488	0.245729	-0.13694	-0.08669	-0.38801	-0.63127	-0.64886	-0.35421
0.216038	-0.05243	-0.13751	-0.31743	-0.67436	-0.80605	-0.69963	-0.81137	0.15365	-0.8451	-0.53729	-0.75241	0.104661	-0.24884	-0.32498	-0.34949	-0.18103	0.263383	-0.22822	-0.42601
0.543694	0.231538	0.999222	0.97935	1.48568	1.63953	1.446042	1.503461	-1.30061	1.167117	0.930873	1.180111	2.682846	-0.86369	2.297976	0.552708	1.947328	1.356899	1.673964	1.23045
1.209054	1.079489	0.787083	0.637701	0.697819	4.151942	0.666225	3.9302	-0.71766	2.652664	0.425412	2.580705	-0.82963	0.462092	0.49442	0.120551	6.324245	2.192857	6.097446	2.093466
-0.21031	-0.02177	-0.43858	-0.39612	0.580587	2.432248	0.811212	2.254612	-0.46708	1.826536	0.585048	1.710869	-0.30171	0.238924	-0.05892	-0.0913	0.287648	0.735748	0.001786	-0.10547
3.584811	0.742977	2.902291	1.483609	-0.79728	0.786023	1.17818	0.891787	-0.01397	0.226591	0.355006	0.212877	-0.84093	1.030641	0.518065	0.691103	-0.19625	-0.10162	-0.10387	0.609002
1.480837	0.85516	0.981762	0.952231	0.416319	1.117993	0.278139	0.974128	0.40856	1.062523	1.205364	1.516175	-1.357	1.114449	1.121816	0.747981	0.4657	-0.23268	1.017643	1.786934
-0.05191	-0.10832	-0.35944	-0.45259	2.080528	0.754759	1.477703	0.620517	0.235592	0.503106	0.889871	0.625762	-0.6358	-0.26752	0.33584	-0.04797	0.201447	0.426633	-0.00805	-0.24875
-0.2219	-0.13158	-0.50426	-0.50901	-0.46239	-0.57307	-0.68611	-0.66349	0.100355	-0.12027	-0.5474	-0.33894	0.093872	-0.15744	-0.47345	-0.4571	-0.35215	-0.03068	-0.64665	-0.66948
0.056657	0.107733	-0.34257	-0.33379	-0.60698	-0.21719	-0.66997	-0.31765	0.09762	-0.23288	-0.47457	-0.45344	0.065928	-0.08181	-0.31752	-0.36222	0.111657	0.46084	-0.14511	0.056005

Figure 34: MSER-SURF features extracted from the image collection

6.4 Other Features

The present study has also considered other features including first order and higher-order statistical derivatives. This includes texture-based features portrayed by the Gray Level Co-occurrence Matrix (GLCM) [135]. The present research work has also used other local feature extraction techniques to retrieve local patterns from the concerned images. Major features used in the study apart from MSER-SURF, are as follows:

6.4.1 Entropy

Entropy measures the lost image information during transmission. This is an important feature for loss calculation during image transformation.

$$\text{Entropy} = \sum_{i,j=0}^{N-1} -\ln(P_{ij})\, P_{ij} \quad \text{… Equation 6.4.1.1}$$

6.4.2 Energy

Energy or Angular Second Moment (ASM) is the measure of uniformity in pixels of an image.

$$\text{Energy} = \sum_{i,j=0}^{N-1} (P_{ij})^2 \quad \text{… Equation 6.4.2.1}$$

Where, P_{ij} = $(i,j)^{th}$ pixel of the concerned image and N = number of gray levels in the image

6.4.3 Homogeneity

Local homogeneity is measured by Inverse Different Moment (IDM) depicting if the gray level is uniform.

$$\text{Homogeneity} = \sum_{i,j=0}^{N-1} \frac{P_{ij}}{1+(i-j)^2} \quad \text{… Equation 6.4.3.1}$$

6.4.4 Correlation

Correlation shows in an image how much correlated a pixel location is with its neighboring pixels.

$$\text{Correlation} = \sum_{i,j=0}^{N-1} P_{ij} \frac{(i-\mu)(j-\mu)}{\sigma^2} \quad \text{… Equation 6.4.4.1}$$

6.4.5 Contrast

It calculates the intensity contrast between a pixel and its adjacent pixels across the input image.

$$\text{Contrast} = \sum_{i,j=0}^{N-1} P_{ij}(i-j)^2 \quad \text{…Equation 6.4.5.1}$$

6.4.6 Mean

The GLCM mean is the mean intensity of pixels, $\mu = \sum_{i,j=0}^{N-1} iP_{ij}$ …Equation 6.4.6.1

6.4.7 Variance

The calculation of the variance of the intensities of all candidate pixel locations in the set of relationships that contribute to the GLCM:

$$\sigma^2 = \sum_{i,j=0}^{N-1} P_{ij}(i-\mu)^2 \quad \text{…Equation 6.4.7.1}$$

mean	sd	entropy	centroidx	centroidy	area	eccent	solidity	contrastx	contrasty	homox	homoy	correlx	correly	energx	energy
0.006897	0.082764	0.059435	79	67.70796	113	0.818106	0.957627	0.002728	0.003472	0.998636	0.998264	0.803936	0.750464	0.983365	0.982625
0.010132	0.100149	0.081666	73.44578	49.63855	166	0.853121	0.892473	0.004216	0.004216	0.997892	0.997892	0.793051	0.793051	0.975428	0.975428
0.023682	0.15206	0.161641	65.54897	64.35567	388	0.511559	0.97	0.00496	0.005704	0.99752	0.997148	0.894366	0.878521	0.948107	0.947371
0.004578	0.067505	0.042163	68.76	73.88	75	0.685435	0.9375	0.002232	0.00248	0.998884	0.99876	0.758879	0.732087	0.988515	0.988269
0.018066	0.133196	0.130442	62.19257	57.47297	296	0.342556	0.951768	0.004712	0.004712	0.997644	0.997644	0.869221	0.869221	0.959277	0.959277
0.023193	0.150522	0.159013	68.28421	80.12105	380	0.654716	0.938272	0.0062	0.005332	0.9969	0.997334	0.865246	0.884112	0.947825	0.948683
0.00885	0.093661	0.07307	65.57931	64.36552	145	0.690578	0.917722	0.003472	0.003472	0.998264	0.998264	0.805145	0.805145	0.97872	0.97872
0.020996	0.143375	0.146997	78.29942	57.19477	344	0.715253	0.828916	0.008185	0.008929	0.995908	0.995536	0.803958	0.786136	0.950134	0.949402
0.016235	0.126383	0.119746	62.38722	74.08271	266	0.846886	0.93662	0.005704	0.00372	0.997148	0.99814	0.824168	0.885327	0.961886	0.963852
0.010498	0.101924	0.084077	73	56.53488	172	0.827223	0.934783	0.003472	0.004216	0.998264	0.997892	0.835454	0.800195	0.975438	0.9747
0.017212	0.130064	0.125486	82.93617	72.71986	282	0.727651	0.886792	0.005704	0.0062	0.997148	0.9969	0.833976	0.81954	0.959969	0.959479
0.003418	0.058365	0.032925	71.5	86.41071	56	0.581167	1	0.001984	0.001984	0.999008	0.999008	0.71329	0.71329	0.991099	0.991099
0.010254	0.100744	0.082472	74.42857	60.93452	168	0.590845	0.879581	0.00434	0.003968	0.99783	0.998016	0.789474	0.807519	0.975062	0.975431
0.007996	0.089063	0.067191	85.09924	55.9313	131	0.75633	0.942446	0.002852	0.00372	0.998574	0.99814	0.82299	0.769117	0.981043	0.980181
0.024597	0.154899	0.166527	62.58065	65.933	403	0.707192	0.948235	0.00496	0.0062	0.99752	0.9969	0.898201	0.872751	0.946338	0.945112
0.006531	0.080551	0.056795	66.30841	66.75701	107	0.770868	0.955357	0.00248	0.003224	0.99876	0.998388	0.811836	0.755386	0.984345	0.983605
0.008911	0.09398	0.073485	65.5411	67.24658	146	0.701759	0.960526	0.00372	0.002976	0.99814	0.998512	0.792643	0.834115	0.978352	0.979091
0.004639	0.067952	0.042636	65.69737	72.82895	76	0.625433	0.962025	0.002232	0.00248	0.998884	0.99876	0.762037	0.735596	0.988393	0.988146
0.002808	0.052914	0.027843	58.06522	59.93478	46	0.311526	1	0.001736	0.001736	0.999132	0.999132	0.694782	0.694782	0.992579	0.992579
0.007019	0.083488	0.060309	74.63478	61.56522	115	0.779683	0.966387	0.003472	0.00248	0.998264	0.99876	0.754773	0.824838	0.982381	0.983367
0.002794	0.052786	0.027728	114.8544	114.5146	103	0.486528	0.990385	0.001206	0.001096	0.999397	0.999452	0.785803	0.805275	0.993164	0.993274
0.009705	0.098036	0.078828	57.67925	61.18239	159	0.629378	0.97546	0.003472	0.003224	0.998264	0.998388	0.822146	0.83485	0.977017	0.977263
0.007751	0.087703	0.065488	72.11811	62.01575	127	0.608784	0.969466	0.003224	0.002728	0.998388	0.998636	0.793651	0.825397	0.981161	0.981654

Figure 35: GLCM and other primitive features extracted from the image dataset

6.4.8 FAST-HOG Features

To retrieve other local features like Binary Large Objects (BLOBs), corners, and edge pixels, etc., a combination of Features from Accelerated Segment Test (FAST) [136] and Histogram of Oriented Gradients (HOG) [137] methods have been used in the present research work. Local features refer to a distinct pattern found in an image. The pattern differs by texture, color, or intensity from its conjugated vicinity. FAST features were detected and the strongest corners are selected. As FAST features are not much suitable for classification, the HOG features are extracted from the selected corners along with valid points from the detected features (Figure 36).

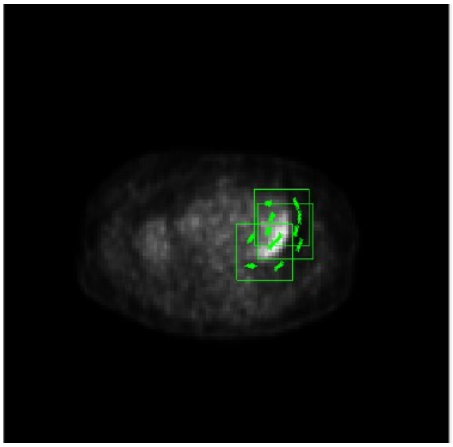


Figure 36: FAST-HOG features detected on NSCLC tumor

f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	f17	f18	f19	f20
0.578774	0.341802	0.907862	0.946885	-0.61278	-0.1207	-0.07568	0.224436	0.875566	0.654432	0.407034	0.502841	2.105008	2.93241	1.869117	2.517305	-0.65664	1.032748	0.784053	0.897851
-0.10001	-0.63247	-0.05705	0.209487	1.134052	0.136475	0.704956	0.026082	-0.39125	0.148589	0.200012	0.027693	0.323856	-0.32468	-0.02525	-0.01429	0.019264	-0.08957	-0.21139	-0.1889
0.467441	-0.48853	0.03694	-0.1898	-0.25864	-0.19164	-0.43828	-0.29827	0.28435	0.880797	0.085575	0.721569	-0.22172	0.28327	-0.14555	-0.00686	-0.02017	-0.70893	-0.30386	-0.38381
-0.03201	0.065766	-0.37844	-0.37765	1.99261	2.399467	1.726054	2.221893	-0.7808	1.93831	1.300851	1.828556	-0.98552	0.581482	0.656336	0.269106	0.682647	0.949762	0.415279	0.052271
2.260805	1.390013	1.75832	0.90859	-0.52302	-0.40589	-0.08704	-0.31317	0.085178	-0.40092	-0.13928	-0.54452	-0.58586	0.140184	0.730167	-0.02735	-0.67807	-0.55947	0.611199	0.961426
-0.11817	-0.02085	-0.51321	-0.53583	-0.09623	1.392135	-0.22594	1.242198	-0.455	0.254093	-0.15339	0.05523	0.101633	-0.19568	-0.48503	-0.50075	1.341887	-0.29707	1.062489	0.361841
0.120809	0.351224	-0.29281	-0.15961	-0.63115	-0.14608	-0.6443	-0.25884	-0.00088	-0.45556	-0.59862	-0.68703	0.068654	-0.1915	-0.4462	-0.4636	0.157769	-0.34948	-0.13343	-0.47265
-0.10911	-0.05356	-0.50164	-0.56412	-0.72499	-0.69468	-0.86456	-0.79484	0.10987	-0.66651	-0.71928	-0.89464	0.121453	-0.21239	-0.48076	-0.49344	-0.45482	-0.30236	-0.72242	-0.68209
-0.12479	-0.18875	-0.46834	-0.48078	0.49783	-0.09496	0.144751	-0.19492	0.962377	2.110545	0.873837	2.009904	-1.02731	1.36939	0.70172	0.968683	-0.43797	-0.06619	-0.71423	-0.6924
-0.02093	0.384362	-0.39535	-0.11667	2.596569	1.111917	2.294274	0.967282	-1.65684	1.669529	1.06216	1.545554	-0.5212	0.181743	0.16886	-0.14682	0.003229	-0.23554	-0.28811	0.119497
-0.64425	-0.57761	1.301347	0.934338	1.523284	0.900813	1.291683	1.173476	-0.5252	0.028622	0.360416	0.229731	0.036153	-0.65659	0.638576	0.17524	-0.77184	1.147281	0.080386	0.562724
0.541997	1.932077	3.243887	2.385588	0.936639	0.496033	0.606426	0.506562	2.030926	-0.75438	3.399623	0.362257	1.437835	-3.09231	1.601881	2.379514	0.848948	-0.74682	2.038314	1.165616
0.010111	0.355443	-0.03487	0.140942	1.72442	0.77964	1.206458	0.720517	-0.33095	0.039493	0.309576	0.072036	-0.2481	-0.58393	-0.02289	0.202156	0.507489	-0.2757	0.585233	0.045416
-0.13257	-0.03658	-0.52649	-0.55173	2.421225	0.694337	3.114052	0.558719	0.957306	0.270918	1.02765	0.32021	-3.52191	0.600876	3.340439	1.016578	0.080707	-0.10457	-0.21126	-0.51541
0.073657	-0.47893	-0.2862	-0.17836	-0.80335	-0.75063	-0.72632	-0.69894	0.088762	-0.50723	-0.55737	-0.72914	0.409488	0.245729	-0.13694	-0.08669	-0.38801	-0.63127	-0.64886	-0.35421
0.216038	-0.05243	-0.13751	-0.31743	-0.67436	-0.80605	-0.69963	-0.81137	0.15365	-0.8451	-0.53729	-0.75241	0.104661	-0.24884	-0.32498	-0.34949	-0.18103	0.263383	-0.22822	-0.42601
0.543694	0.231538	0.999222	0.97935	1.48568	1.63953	1.446042	1.503461	-1.30061	1.167117	0.930873	1.180111	2.682846	-0.86369	2.297976	0.552708	1.947328	1.356899	1.673964	1.23045
1.209054	1.079489	0.787083	0.637701	0.697819	4.151942	0.666225	3.9302	-0.71766	2.652664	0.425412	2.580705	-0.82963	0.462092	0.49442	0.120551	6.324245	2.192857	6.097446	2.093466
-0.21031	-0.02177	-0.43858	-0.39612	0.580587	2.432248	0.811212	2.254612	-0.46708	1.826536	0.585048	1.710869	-0.30171	0.238924	-0.05892	-0.0913	0.287648	0.735748	0.001786	-0.10547
3.584811	0.742977	2.902291	1.483609	-0.79728	0.786023	1.17818	0.891787	-0.01397	0.226591	0.355006	0.212877	-0.84093	1.030641	0.518065	0.691103	-0.19625	-0.10162	-0.10387	0.609002
1.480837	0.85516	0.981762	0.952231	0.416319	1.117993	0.278139	0.974128	0.40856	1.062523	1.205364	1.516175	-1.357	1.114449	1.121816	0.747981	0.4657	-0.23268	1.017643	1.786934
-0.05191	-0.10832	-0.35944	-0.45259	2.080528	0.754759	1.477703	0.620517	0.235592	0.503106	0.889871	0.625762	-0.6358	-0.26752	0.33584	-0.04797	0.201447	0.426633	-0.00805	-0.24875
-0.2219	-0.13158	-0.50426	-0.50901	-0.46239	-0.57307	-0.68611	-0.66349	0.100355	-0.12027	-0.5474	-0.33894	0.093872	-0.15744	-0.47345	-0.4571	-0.35215	-0.03068	-0.64665	-0.66948
0.056657	0.107733	-0.34257	-0.33379	-0.60698	-0.21719	-0.66997	-0.31765	0.09762	-0.23288	-0.47457	-0.45344	0.065928	-0.08181	-0.31752	-0.36222	0.111657	0.46084	-0.14511	0.056005

Figure 37: Glimpse of FAST-HOG features detected from the image collection

Figure 36 displays the original image with an overlay of HOG features around the strongest corners detected by FAST. HOG features are extracted from grayscale input images. HOG features are extracted around specified point locations. The function returns the locations from the input whose adjacent pixels are fully constrained by the input image. Scale information linked with the points is overlooked. HOG features having length N is returned as a 1-by-N vector, where. Local shape information is being encoded by the returned features (Figure 37). Its various input parameters are as follows:

a) Input image, specified M-by-N 2-D grayscale. The input image should be a real, non-sparse value. Images may lose detailed shape information that the HOG function can encode due to rigid cropping. The issue may be resolved by including padding of background pixels around the cropped area.

b) Center location point formed as an M-by-2 matrix having M number of (x, y) coordinates. Descriptors are being generated from the adjacent points that are fully confined by the image edge. The size of the neighborhood pixels is determined by the *BlockSize* parameter. Pixel locations that exist within the image boundary are often used to decide the valid output points. The function ignores the scale information associated with these points.

c) Cell size is expressed as a vector comprising pixel information. To capture large-scale spatial information, increase cell size. If the cell size is magnified, minute details regarding scale may get lost.

d) The number of cells in a block, specified as a 2-element vector. The ability to suppress local illumination changes may be reduced due to a large block-size value. As the number of pixels exists in a chunk, these changes may get lost with averaging. To capture the significant local pixels block-size may have to be reduced. Smaller block sizes can help suppress illumination changes of HOG features.

e) The number of overlapping cells between adjacent blocks, specified as a 2-element vector. Although large overlap values can confine ample information, they produce a larger feature vector size.

f) The number of orientation histogram bins, specified as a positive scalar. To encode finer orientation details, the number of bins should be increased. The increasing number of bins increases the size of the feature vector, which requires more processing overhead.

g) Selection of orientation values, specified as a logical scalar. If this property becomes true, orientation values get uniformly spaced within a span of bins between -180 and 180 degrees. If this property becomes false, it is evenly distributed from 0 through 180. Signed orientation differentiates between light-to-dark and dark-to-light transitions.

The output arguments are as follows:

a) Extracted features, returned as either a vector or a matrix. The local shape information may be encoded by features retrieved from regions or point locations within an image.

b) Valid points are associated with each feature descriptor vector output. This output can be returned as an M-by-2 matrix of (x,y) coordinates. The function culls out M number of attributes from candidate points in a region having a size equal to [CellSize.*BlockSize]. The extracted descriptors are the same type of object or matrix as that of the input.

6.5 Feature Selection

Feature selection or dimensionality reduction is a very important step in data analysis where the strongest spatial features are retained for the study from a pile of features. Normally the feature selection is done by running a Principal Components Analysis (PCA) [138] for dimensionality reduction. This is typically accomplished by selecting enough eigenvectors to account for 95% of the variance in the extracted feature set. In the present study, feature selection or dimensionality reduction has been done by Independent Component Analysis (ICA) [139] which is more advanced than PCA. PCA optimizes the second-order statistics and finds uncorrelated components, whereas ICA optimizes higher-order statistics and finds independent components [140]

6.5.1 Algorithm

If $X=[X_1, X_2 \ldots X_m]^T$ is a random observed vector having m elements which are a mix of m independent elements of another random vector $S= [S_1, S_2, \ldots, S_m]^T$, then X may be expressed as:

$$X=AS \qquad \ldots\text{Equation 6.5.1.1}$$

Where A is an m-by-m mixing matrix.

The goal of ICA is to estimate A and also estimate the source distribution S. Methods like PCA which is based on second-order moments, cannot recover A. ICA uses higher-order moments to recover A.

Now, pre-assumptions made during ICA are as follows:

a) sources are statistically independent;
b) mixing matrix is square;
c) there should be no external noise;
d) data have zero mean;
e) signals must not have a normal probability density

Statistical independence may be expressed as:

$E[g_1(x_i)\, g_2(x_j)] - E[g_1(x_i)]\, E[g_2(x_j)] = 0$ for $i \neq j$ …Equation 6.5.1.2

Where E[.] is the expectation for any function g(x). In specific, Gaussian uncorrelatedness is equivalent to independence. In ICA typically Minimization of mutual information or Maximization of non-Gaussianity is done. Independence may be achieved by forcing two random variables to be as far from the normal distribution as possible by measuring the non-gaussianity. Negentropy or the positive measure of gaussianity.are approximated to calculate the non-gaussianity. This algorithm is called FastICA:

1. *Center S by subtracting the mean*
2. *Whiten S*
3. *Choose a random initial value w for the de-mixing matrix*
4. *Calculate the new value for A*
5. *Normalize A*
6. *Check if the algorithm has converged and if it hasn't, resume from step 4*
7. *Take the dot product of A and S to get the independent source signals X (Equation 6.5.1.1)*

The eigenvalues are being decomposed w.r.t. its covariance matrix to whitening a signal. The potential correlations between the components of a signal are removed. This may be expressed as:

$$\tilde{x} = ED^{-1/2}E^{T}x$$ …Equation 6.5.1.3

Where D is a diagonal matrix of eigenvalues (lambda).

6.5.2 Methodology

In the present research work, transformation on numeric data has been done using the FastICA algorithm [141]. First, the data whitening (decoupling transform) has been done. Then, the FastICA main loop has been executed for 200 iterations with an error tolerance value 1.0E-4 (for solution convergence). Strongest 64 features are taken for final classification based on their respective ranks in ICA analysis.

f1	f2	f3	f4	f5	f6	f7	...	f58	f59	f60	f61	f62	f63	f64
0.439602	0.547821	0.00481	0.006076	0.145296	0.215474	0.04261	...	0.623217	0.314782	0.314895	0.359191	0.495266	0.116026	0.118451
0.451386	0.546407	0.037355	0.01083	0.130287	0.206117	0.049503	...	0.808204	0.029337	0.022213	0.435111	0.565142	0.011436	0.043668
0.425059	0.642745	0.220604	0.48215	0.200386	0.442791	0.204353	...	0.644277	0.351163	0.257528	0.268111	0.464487	0.263158	0.164063
0.451386	0.546407	0.037355	0.01083	0.130287	0.206117	0.049503	...	0.808204	0.029337	0.022213	0.435111	0.565142	0.011436	0.043668
0.464168	0.496577	0.051501	0.103768	0.103694	0.158244	0.071917	...	0.403364	0.265747	0.503787	0.391405	0.49696	0.064354	0.109445
0.441988	0.550917	0.009033	0.011411	0.400033	0.364758	0.266112	...	0.745275	0.141209	0.161274	0.628167	0	0.39599	1
0.440039	0.548542	0.005582	0.007051	0.127487	0.202639	0.03426	...	0.505227	0.198528	0.38514	0.333822	0.534269	0.160191	0.104016
0.44191	0.548837	0.011732	0.017664	0.125539	0.177816	0.025069	...	0.82816	0.003934	0.003029	0.429907	0.560158	0.002133	0.00317
0.434875	0.558347	0.050841	0.032015	0.134413	0.202589	0.051069	...	0.771489	0.114443	0.15947	0.394151	0.521096	0.067847	0.106775
0.439602	0.547821	0.00481	0.006076	0.145296	0.215474	0.04261	...	0.623217	0.314782	0.314895	0.359191	0.495266	0.116026	0.118451
0.434875	0.558347	0.050841	0.032015	0.134413	0.202589	0.051069	...	0.771489	0.114443	0.15947	0.394151	0.521096	0.067847	0.106775
0.440039	0.548542	0.005582	0.007051	0.127487	0.202639	0.03426	...	0.505227	0.198528	0.38514	0.333822	0.534269	0.160191	0.104016
0.441988	0.550917	0.009033	0.011411	0.400033	0.364758	0.266112	...	0.745275	0.141209	0.161274	0.628167	0	0.39599	1
0.438737	0.554547	0.003278	0.018076	0.133879	0.223408	0.038016	...	0.11798	0.087388	0.841907	0.304226	0.006421	0.204818	0.938166
0.430309	0.570249	0.019813	0.047477	0.126579	0.231136	0.036686	...	0.798131	0.012771	0.038544	0.425244	0.54657	0.009323	0.025348
0.455201	0.542289	0.037457	0.025837	0.133796	0.157754	0.062879	...	0.829217	0.028993	0.011555	0.435327	0.552661	0.015483	0.017582
0.434875	0.558347	0.050841	0.032015	0.134413	0.202589	0.051069	...	0.771489	0.114443	0.15947	0.394151	0.521096	0.067847	0.106775
0.430309	0.570249	0.019813	0.047477	0.126579	0.231136	0.036686	...	0.798131	0.012771	0.038544	0.425244	0.54657	0.009323	0.025348
0.440039	0.548542	0.005582	0.007051	0.127487	0.202639	0.03426	...	0.505227	0.198528	0.38514	0.333822	0.534269	0.160191	0.104016
0.438737	0.554547	0.003278	0.018076	0.133879	0.223408	0.038016	...	0.11798	0.087388	0.841907	0.304226	0.006421	0.204818	0.938166
0.441025	0.553973	0.007328	0.017022	0.118203	0.205648	0.026126	...	0.765321	0.032049	0.07289	0.449336	0.51297	0.051154	0.082131
0.455345	0.561656	0.034112	0.042714	0.115512	0.270605	0.049775	...	0.765962	0.092994	0.129534	0.308909	0.503025	0.197252	0.098937
0.440648	0.575651	0.037835	0.060644	0.180319	0.623649	0.361021	...	0.432169	0.177026	0.46923	0.370885	0.447306	0.102363	0.196295
0.470039	0.607865	0.067003	0.143087	0.354453	0.87161	0.584796	...	0.389561	0.141565	0.528585	0.414075	0.545626	0.038118	0.044827

Figure 38: Selected features as a result of ICA

6.6 Conclusion

Feature extraction is an unavoidable step in the orthodox machine learning curriculum. Although the manual feature extraction causes processing overhead, it is necessary to feed features in the traditional machine learning algorithms as they do not have an automatic way of extracting features. Such features may turn out very handy when limited hardware resources are there and images cannot be automatically processed. In the present study, different BLOB or Corner based feature extraction techniques have been described. MSER-SURF and FAST-HOG based techniques have been used. Other primitive and higher-order features have also been extracted by deriving first and second-order statistical derivatives. Extracted features have been further used for database preparation.

7. Database Preparation[5]

American Joint Committee on Cancer (AJCC) has propounded the popular Tumor-Node-Metastasis (TNM) system (Table 1) for cancer staging. A further abstraction on TNM staging is the AJCC staging which may be very helpful if used as the common class variable in a supervised machine learning method. For example, Table 2 depicts the AJCC staging derivation from the TNM staging of lung cancer. Different types of cancers have different propositions for TNM staging, but AJCC staging brings them under a common platform and if the AJCC staging is automated, the respective TNM staging may be easily determined irrespective of the cancer type.

Primary Tumor (T)	TX	Primary tumor cannot be assessed
	T0	No evidence of primary tumor
	T_{is}	Carcinoma in situ
	T1	Tumor size <=3cm
	T1a	Tumor size <=2cm
	T1b	Tumor size >2cm and <=3cm
	T2	Tumor size >3cm and <=7cm
	T2a	Tumor size >3cm and <=5cm
	T2b	Tumor size >5cm and <=7cm
	T3	Tumor size >7cm
	T4	Any size that invades other organs such as heart, nervous system, etc.
Regional Lymph Nodes (N)	NX	Regional Lymph Nodes cannot be assessed
	N0	No Regional Lymph Node metastases
	N1	Metastasis in ipsilateral peribronchial and/or ipsilateral hilar lymph nodes and intrapulmonary nodes, including involvement by direct extension
	N2	Metastasis in ipsilateral mediastinal and/or subcarinal lymph node(s)
	N3	Metastasis in contralateral mediastinal, contralateral hilar, ipsilateral or contralateral scalene, or supraclavicular lymph node(s)
Distant Metastasis (M)	M0	No Distant Metastasis
	M1	Distant Metastasis
	M1a	Separate tumor nodule(s) in a contralateral lobe
	M1b	Distant Metastasis (in extrathoracic organs)

Table1: Tumor-Node-Metastasis (TNM) Staging System

[5] Based on author's publication no. 1 and no. 9 [Appendix D]

Stage 0	T_{is}	N0	M0	Stage IIB	T2b	N1	M0	Stage IIIB	T1a	N3	M0
Stage IA	T1a	N0	M0		T3	N0	M0		T1b	N3	M0
	T1b	N0	M0	Stage IIIA	T1a	N2	M0		T2a	N3	M0
Stage IB	T2a	N0	M0		T1b	N2	M0		T2b	N3	M0
Stage IIA	T2b	N0	M0		T2a	N2	M0		T3	N3	M0
	T1a	N1	M0		T2b	N2	M0		T4	N2	M0
	T1b	N1	M0		T3	N1	M0		T4	N3	M0
	T2a	N1	M0		T3	N2	M0	Stage IV	Any T	Any N	M1a
					T4	N0	M0		Any T	Any N	M1b
					T4	N1	M0				

Table2: AJCC 7th Edition Staging System of Lung Cancer

7.1 Methodology

Extracted features from the image collection have been stored in a Comma Separated Value (CSV) file. In this way, the data file becomes platform-independent and can be accessed by any known data analysis or statistical packages. The clinical information from the original dataset has been extracted (Figure 39) and clubbed with the extracted features as per case-identification numbers. Four different datasets have been prepared: the T-staging information, M-staging information, N-staging information, and AJCC staging (Figure 40) information are appended in the feature set as the class labels, respectively. Another file has been prepared for inserting histopathological grading information as the class or target variable with the extracted features as per the concerned case identifier. Thus, five different data files have been prepared. All missing values for nominal and numeric attributes in the dataset are replaced by the modes (for nominal data) and means (for numeric data), respectively.

Case ID	Gender	Histology	Pathological T stage	Pathological N stage	Pathological M stage	AJCC Staging (Version 7)	Histopathological Grade	Chemotherapy
R01-021	Female	Adenocarcinoma	T1a	N0	M0	IA	G2 Moderately differentiated	No
R01-022	Male	Adenocarcinoma	Tis	N0	M0	IA	G2 Moderately differentiated	No
R01-023	Male	Adenocarcinoma	Tis	N0	M0	IA	G2 Moderately differentiated	No
R01-024	Male	Adenocarcinoma	T2a	N0	M0	IIA	G1 Well differentiated	No
R01-025	Male	Adenocarcinoma	T2a	N0	M0	IIA	G1 Well differentiated	No
R01-026	Male	Adenocarcinoma	T1b	N0	M0	IA	G2 Moderately differentiated	Yes
R01-027	Male	NSCLC NOS (not otherwise	T4	N0	M0	IIIA	G3 Poorly differentiated	No
R01-028	Male	Adenocarcinoma	T2a	N0	M0	IB	G3 Poorly differentiated	No
R01-029	Male	NSCLC NOS (not otherwise	T3	N2	M0	IIIA	G2 Moderately differentiated	Yes
R01-030	Male	Adenocarcinoma	T1a	N2	M0	IIIA	G1 Well differentiated	No
R01-031	Male	NSCLC NOS (not otherwise	T1b	N0	M0	IA	G3 Poorly differentiated	No
R01-032	Male	Adenocarcinoma	T2a	N0	M0	IB	Other, Type II: Moderately to poorly differentiate	No
R01-033	Male	Adenocarcinoma	T1b	N2	M0	IIIA	Other, Type I: Well to moderately differentiated	No
R01-034	Male	Adenocarcinoma	T2a	N1	M0	IIA	G2 Moderately differentiated	Yes
R01-035	Male	Adenocarcinoma	T2a	N2	M0	IIIA	G1 Well differentiated	Yes
R01-036	Male	Adenocarcinoma	T3	N2	M0	IIIA	G3 Poorly differentiated	Yes
R01-037	Male	Adenocarcinoma	T2a	N0	M0	IB	Other, Type II: Moderately to poorly differentiate	No
R01-038	Male	Squamous cell carcinoma	T2a	N0	M0	IB	G2 Moderately differentiated	No
R01-039	Male	Squamous cell carcinoma	T1b	N0	M0	IA	G1 Well differentiated	No
R01-040	Male	Squamous cell carcinoma	T2a	N0	M0	IB	G2 Moderately differentiated	No
R01-041	Female	Squamous cell carcinoma	T1a	N0	M0	IA	G2 Moderately differentiated	No
R01-042	Male	Squamous cell carcinoma	T2b	N0	M0	IIA	Other, Type I: Well to moderately differentiated	No
R01-043	Male	Adenocarcinoma	T1a	N0	M0	IA	G2 Moderately differentiated	No
R01-044	Male	Adenocarcinoma	T1a	N0	M0	IA	G2 Moderately differentiated	No

Figure 39: Glimpse of clinical information retrieved from NSCLC Radiogenomics

Case ID	f1	f2	f3	f4	f5	f6	f7	f8	...	f60	f61	f62	f63	f64	AJCC
R01-001	0.000876	0.000452	0.001856	0.001625	0.001185	0.015315	0.011849	0.026039	...	0.02235	-0.00078	-1.3E-05	0.002443	0.001339	IA
R01-002	8.78E-05	-0.00058	0.000641	0.00085	0.022209	0.021155	0.022783	0.021419	...	0.024818	0.005619	-0.00434	0.005647	0.00699	IA
R01-003	0.000747	-0.00043	0.000759	0.00043	0.005447	0.013704	0.006771	0.013865	...	0.015372	-0.00511	-0.00192	0.005113	0.001938	IIB
R01-004	0.000167	0.000159	0.000236	0.000233	0.032542	0.072547	0.037084	0.072559	...	0.054769	-9.4E-05	-0.00028	0.000155	0.000281	IIIA
R01-005	0.002829	0.001566	0.002927	0.001585	0.002265	0.008838	0.01169	0.013518	...	0.004772	-0.00035	0.000462	0.001113	0.000896	IB
R01-006	6.67E-05	6.67E-05	6.67E-05	6.67E-05	0.007402	0.049671	0.009745	0.049742	...	0.03673	-0.00023	-0.00012	0.000234	0.000116	IA
R01-007	0.000344	0.000462	0.000344	0.000462	0.000964	0.014738	0.003885	0.014783	...	0.039321	-0.00034	-0.00035	0.000376	0.000361	IIA
R01-008	7.72E-05	3.19E-05	8.12E-05	3.69E-05	-0.00017	0.00228	0.0008	0.0023	...	0.010192	-3.8E-05	-0.00018	5.94E-05	0.000177	O
R01-009	5.9E-05	-0.00011	0.000123	0.000125	0.014552	0.015899	0.014937	0.016272	...	0.046813	-0.0027	-0.00203	0.00577	0.004026	IA
R01-010	0.00018	0.000497	0.000215	0.000507	0.039812	0.043307	0.045043	0.043339	...	0.086461	-0.00102	-0.00147	0.001026	0.001467	IIB
R01-011	-0.00054	-0.00053	0.002351	0.001612	0.026894	0.038513	0.031001	0.048141	...	0.029896	0.000569	0.000924	0.001256	0.002247	IIIA
R01-012	0.000833	0.002142	0.004797	0.003137	0.019833	0.029321	0.021403	0.032609	...	0.055337	-0.00304	-0.00557	0.003036	0.005572	IB
R01-013	0.000216	0.000467	0.000669	0.000778	0.029315	0.035761	0.029807	0.037592	...	0.026484	-0.00264	-0.00242	0.002993	0.002909	IB
R01-014	4.99E-05	4.99E-05	4.99E-05	4.99E-05	0.037701	0.033824	0.056525	0.033824	...	0.03148	-5E-05	-5E-05	4.99E-05	4.99E-05	O

Figure 40: AJCC staging information clubbed as the class variable with the extracted feature set as per case ID

7.2 Standardization and Normalization

Feature standardization implies that each attribute in the data set will have zero-mean and a unit variance:

$$x' = \frac{x - \bar{x}}{\sigma} \quad \text{…Equation 7.2.1}$$

Where x is the original feature vector, $\bar{x}$ is the mean of that feature vector, σ is its standard deviation and x' is the standardized value. The min-max method of rescaling was adopted to confine the values of features within the range [0, 1]:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad \text{…Equation 7.2.2}$$

Where x is an input value of a feature and x' is the normalized value of the same. First, the dataset was separately standardized (Equation 7.2.1) (Figure 41) to the range [-1, 1] and then normalized (Equation 7.2.2) (Figure 42) to the range [0, 1].

Case ID	f1	f2	f3	f4	f5	f6	f7	f8	...	f60	f61	f62	f63	f64	AJCC
R01-001	0.578774	0.341802	0.907862	0.946885	-0.61278	-0.1207	-0.07568	0.224436	...	-0.00885	-0.15262	0.383916	0.540587	-0.03929	IA
R01-002	-0.10001	-0.63247	-0.05705	0.209487	1.134052	0.136475	0.704956	0.026082	...	0.104023	2.896317	-1.85726	2.003236	2.656619	IA
R01-003	0.467441	-0.48853	0.03694	-0.1898	-0.25864	-0.19164	-0.43828	-0.29827	...	-0.328	-2.21354	-0.60477	1.759495	0.24642	IIB
R01-004	-0.03201	0.065766	-0.37844	-0.37765	1.99261	2.399467	1.726054	2.221893	...	1.473962	0.176439	0.245499	-0.50404	-0.54412	IIIA
R01-005	2.260805	1.390013	1.75832	0.90859	-0.52302	-0.40589	-0.08704	-0.31317	...	-0.81281	0.053237	0.630399	-0.06653	-0.25068	IB
R01-006	-0.11817	-0.02085	-0.51321	-0.53583	-0.09623	1.392135	-0.22594	1.242198	...	0.648863	0.109633	0.330818	-0.46786	-0.62266	IA
R01-007	0.120809	0.351224	-0.29281	-0.15961	-0.63115	-0.14608	-0.6443	-0.25884	...	0.767368	0.058889	0.207516	-0.4031	-0.50577	IIA
R01-008	-0.10911	-0.05356	-0.50164	-0.56412	-0.72499	-0.69468	-0.86456	-0.79484	...	-0.56491	0.202851	0.299351	-0.54757	-0.5937	O
R01-009	-0.12479	-0.18875	-0.46834	-0.48078	0.49783	-0.09496	0.144751	-0.19492	...	1.110072	-1.06435	-0.66285	2.059356	1.242724	IA
R01-010	-0.02093	0.384362	-0.39535	-0.11667	2.596569	1.111917	2.294274	0.967282	...	2.923516	-0.26611	-0.36931	-0.1064	0.021816	IIB
R01-011	-0.64425	-0.57761	1.301347	0.934338	1.523284	0.900813	1.291683	1.173476	...	0.336312	0.491821	0.86973	-0.0012	0.393991	IIIA
R01-012	0.541997	1.932077	3.243887	2.385588	0.936639	0.496033	0.606426	0.506562	...	1.49994	-1.22466	-2.49695	0.811402	1.980349	IB
R01-013	0.010111	0.355443	-0.03487	0.140942	1.72442	0.77964	1.206458	0.720517	...	0.180229	-1.03382	-0.86504	0.791494	0.709644	IB
R01-014	-0.13257	-0.03658	-0.52649	-0.55173	2.421225	0.694337	3.114052	0.558719	...	0.408743	0.19728	0.365006	-0.55189	-0.65413	O

Figure 41: Standardized Feature set clubbed with AJCC staging

Case ID	f1	f2	f3	f4	f5	f6	f7	f8	...	f60	f61	f62	f63	f64	AJCC
R01-001	0.496845	0.582399	0.22489	0.248781	0.114322	0.286713	0.174441	0.228395	...	0.198125	0.400864	0.560865	0.151622	0.118513	IA
R01-002	0.442893	0.496058	0.077675	0.13013	0.472189	0.329647	0.338196	0.187094	...	0.220002	0.646873	0.334334	0.35047	0.618772	IA
R01-003	0.487996	0.508815	0.092014	0.065883	0.186874	0.274871	0.098377	0.119559	...	0.13627	0.234574	0.460931	0.317333	0.17153	IIB
R01-004	0.448298	0.557937	0.028641	0.035658	0.648078	0.707437	0.552394	0.644301	...	0.485515	0.427415	0.546874	0.009605	0.024835	IIIA
R01-005	0.63054	0.675293	0.354644	0.242619	0.132711	0.239103	0.172056	0.116455	...	0.042307	0.417474	0.585778	0.069085	0.079286	IB
R01-006	0.441449	0.55026	0.008078	0.010206	0.220147	0.53927	0.14292	0.440311	...	0.3256	0.422024	0.555498	0.014524	0.010261	IA
R01-007	0.460444	0.583234	0.041705	0.070741	0.110559	0.282475	0.05516	0.127768	...	0.348568	0.41793	0.543035	0.023327	0.031953	IIA
R01-008	0.44217	0.547362	0.009845	0.005653	0.091334	0.190891	0.008955	0.016163	...	0.090353	0.429546	0.552317	0.003687	0.015636	O
R01-009	0.440923	0.535381	0.014925	0.019063	0.341849	0.29101	0.220681	0.141078	...	0.414988	0.327299	0.455061	0.3581	0.356407	IA
R01-010	0.449179	0.586171	0.02606	0.07765	0.771808	0.49249	0.671591	0.383069	...	0.766459	0.391707	0.484731	0.063664	0.129852	IIB
R01-011	0.399634	0.50092	0.284924	0.246762	0.551929	0.457247	0.461275	0.426002	...	0.265023	0.452862	0.609969	0.077966	0.198914	IIIA
R01-012	0.493922	0.723331	0.581295	0.480274	0.431745	0.389672	0.317527	0.287138	...	0.49055	0.314364	0.269676	0.18844	0.493282	IB
R01-013	0.451646	0.583608	0.081059	0.119101	0.593135	0.437018	0.443397	0.331688	...	0.234772	0.329763	0.434624	0.185733	0.257487	IB
R01-014	0.440305	0.548867	0.006053	0.007647	0.735886	0.422778	0.843557	0.297999	...	0.279061	0.429096	0.558953	0.0031	0.004422	O

Figure 42: Normalized feature set clubbed with AJCC staging

7.3 Data Resampling

The dataset has been resampled to avoid the effect of sampling bias and class imbalance. Synthetic Minority Oversampling TEchnique (SMOTE) [142] was used as the data resampling technique in the study. If a dataset has s samples and f features in the feature space, to oversample, a sample is taken from the dataset, and its k nearest neighbors are considered. A vector between one of the k neighbors is taken. The vector is multiplied by a random number x which lies between 0, and 1. This is added to the current data point to create the new, synthetic data point. The SMOTE algorithm is as follows:

Input: no. of minority class samples (s); % of SMOTE (p); no. of nearest neighbors (k); no. of attributes (t)

Output: (p/100)*s synthetic minority class samples

Step 1: If p<100 then randomize the minority class samples p

Step 2: the amount of SMOTE is assumed to be a multiple of 100

Step 3: initialize an index to 0. This will count the no. of synthetic samples generated

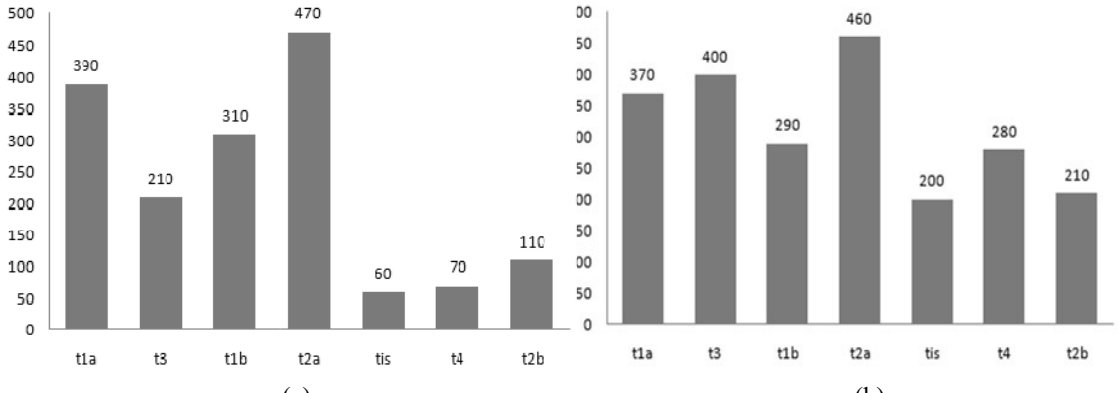
Step 4: initialize an array for storing synthetic samples

Step 5: for each minority class samples compute k, save the indices and populate the synthetic samples array for each t

At each resampling iteration, the index of the non-empty minority class value (to which SMOTE should be applied) has been selected along with 5 as the number of nearest neighbors and the average percentage of SMOTE instances to be created is 100 (with a random seed value of 1). In this way, the final dataset is prepared and is ready to be fed into any machine learning process. For example, T-stage information is clubbed with the feature set as the class variable (Figure 43). Such resampled data may be fed into any machine learning algorithm and the outcome obtained would be less biased. Thus, the result will be more robust and trustworthy, which may be compared with other machine learning models. In this way, the performance of the newly developed model becomes more reliable.

f1	f2	f3	f4	f5	...	f25	f26	f27	f28	f29	f30	f31	f32	f33	T-Stage
0.006897	0.082764	0.059435	79	67.70797	...	0.17429	0.068662	0.090976	0.149021	0.166608	0.237764	0.127928	0.122698	0.099106	T1a
0.010132	0.100149	0.081666	73.44578	49.63855	...	0.190752	0.29852	0.10056	0.062611	0.032788	0.014766	0.024693	0.120736	0.29852	T1a
0.023682	0.15206	0.161641	65.54897	64.35567	...	0.311401	0.161006	0.062116	0.050244	0.04441	0.028175	0.045779	0.124013	0.257344	T3
0.004578	0.067505	0.042163	68.76	73.88	...	0.308194	0.308194	0.188269	0.141006	0.065096	0.007018	0.002479	0.005268	0.015152	T1b
0.018066	0.133196	0.130442	62.19257	57.47297	...	0.246605	0.216608	0.141492	0.130275	0.200024	0.154105	0.074097	0.038471	0.012594	T2a
0.023193	0.150522	0.159013	68.28421	80.12105	...	0.083895	0.078524	0.042517	0.004371	0.032514	0.323828	0.323828	0.149492	0.059702	T1b
0.00885	0.093661	0.07307	65.57931	64.36552	...	0.207944	0.263077	0.21714	0.136991	0.00533	0.005793	0.024564	0.067887	0.105772	T1a
0.020996	0.143375	0.146997	78.29942	57.19477	...	0.276753	0.146699	0.071113	0.095483	0.039818	0.094928	0.050805	0.088894	0.152309	Tis
0.016235	0.126383	0.119746	62.38722	74.08271	...	0.006598	0.029807	0.018356	0.031114	0.033875	0.107816	0.1863	0.265212	0.265212	T1a
0.010498	0.101924	0.084077	73	56.53488	...	0.004886	0.117577	0.051444	0.054425	0.229287	0.295642	0.284459	0.224459	0.064874	T3
0.017212	0.130064	0.125486	82.93617	72.71986	...	0.134222	0.143584	0.192036	0.160101	0.212792	0.092397	0.09633	0.099099	0.125454	T1a
0.003418	0.058365	0.032925	71.5	86.41071	...	0.047549	0.216612	0.204857	0.20198	0.094418	0.107752	0.156852	0.25567	0.25567	T2a
0.010254	0.100744	0.082472	74.42857	60.93452	...	0.1631	0.183474	0.150857	0.141486	0.221363	0.130722	0.090742	0.092472	0.118165	T2a

Figure 43: A glimpse of the feature set used in the study with T-Staging as the class variable



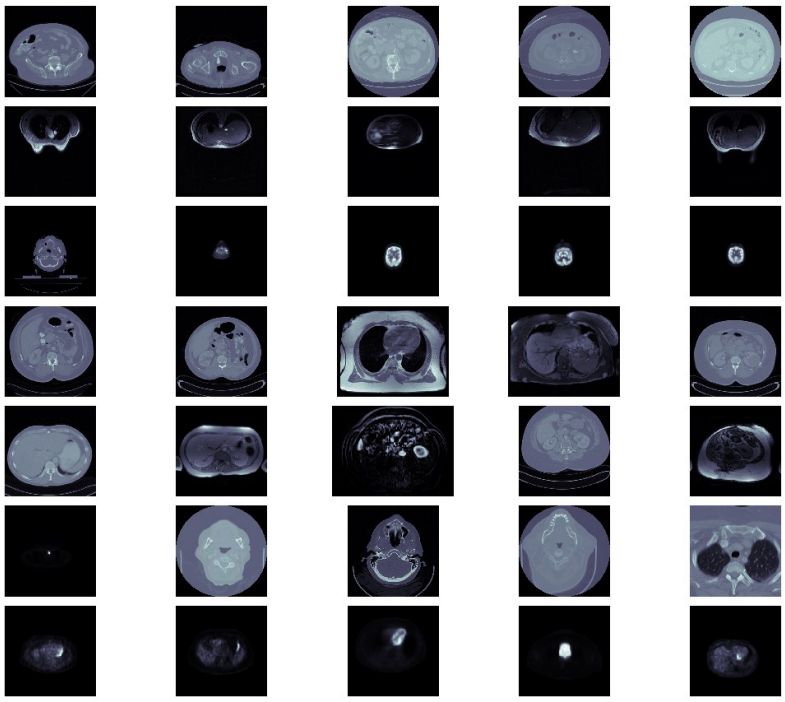
(a) (b)

Figure 44: Sample distribution under T-stage class (a) before re-sampling, (b) after re-sampling by SMOTE

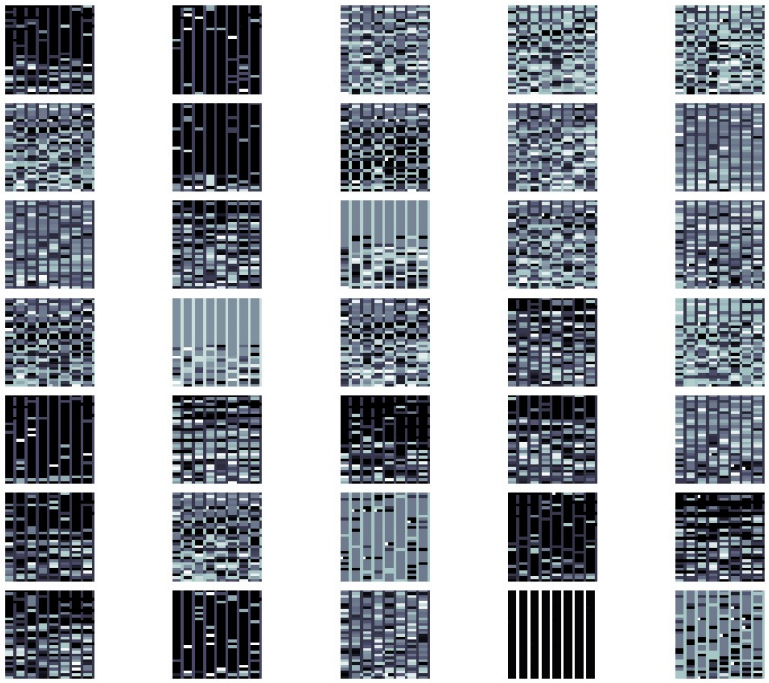
From Figure 44, it may be observed that after applying SMOTE, the class imbalance scenario has reduced to a larger extent. This has also reduced the chance of overfitting and underfitting while carrying out experiments with different machine learning algorithms.

7.4 Imagery Database Preparation

The main course of database preparation that has been discussed so far in the study is meant for use with traditional machine learning methods and a one-dimensional Convolutional Neural Network (1-D CNN). Such a database preparation follows manual feature extraction and gives birth to a semi-automated tumor classification system. This type of system is very useful where hardware resources are limited. On the other hand, if there is no hardware constraint, a fully automated system may be implemented by using a two-dimensional convolutional neural network (2-D CNN) or higher-level architectures as applicable. Database preparation for such a system is somewhat different and involves a specific flow of activities. Here, images are being used directly rather than extracting features into a repository. Images are labeled according to their CaseIDs and stored in a particular folder. Class or target variables of each of the images are stored in a CSV file as per the CaseID. Thus, there is a scope of mapping between the images and their respective class variables, e.g., AJCC staging. First images are read and converted into a pixel array. Later, these pixel arrays are downsampled (Figure 45), reshaped, and again transformed into image arrays. After that, image arrays are tagged with the corresponding class value and compressed for future use (Figure 46). Compression reduces the memory overhead while carrying out experiments.



(a)



(b)

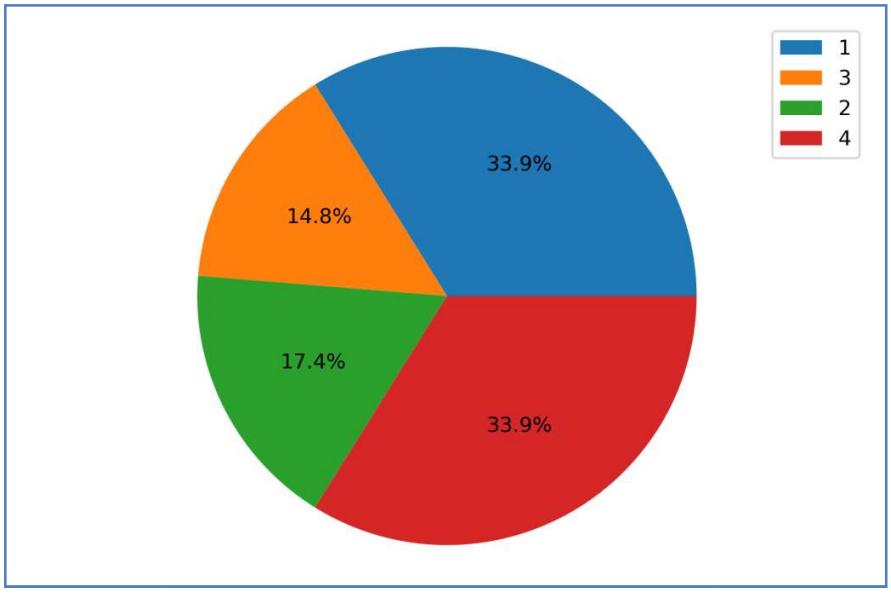
Figure 45: Image transformation during down-sampling phase at a glance - (a) original image collection (b) transformed image collection having distinct patterns for each class of tumors

In the case of image regeneration, images are labeled and directly fed into the machine learning model after resizing and re-sampling. As the outcome, regenerated images with calculated labels may be observed. Such a system works like an encoder-decoder system with an unsupervised learning mode. In the present research work, labels are tagged with the imagery array and fed into different machine learning models. Models work like encoders with supervised learning mode. The outcome may be decoded by using an inverse transformation to generate the stage and grade of a tumor.

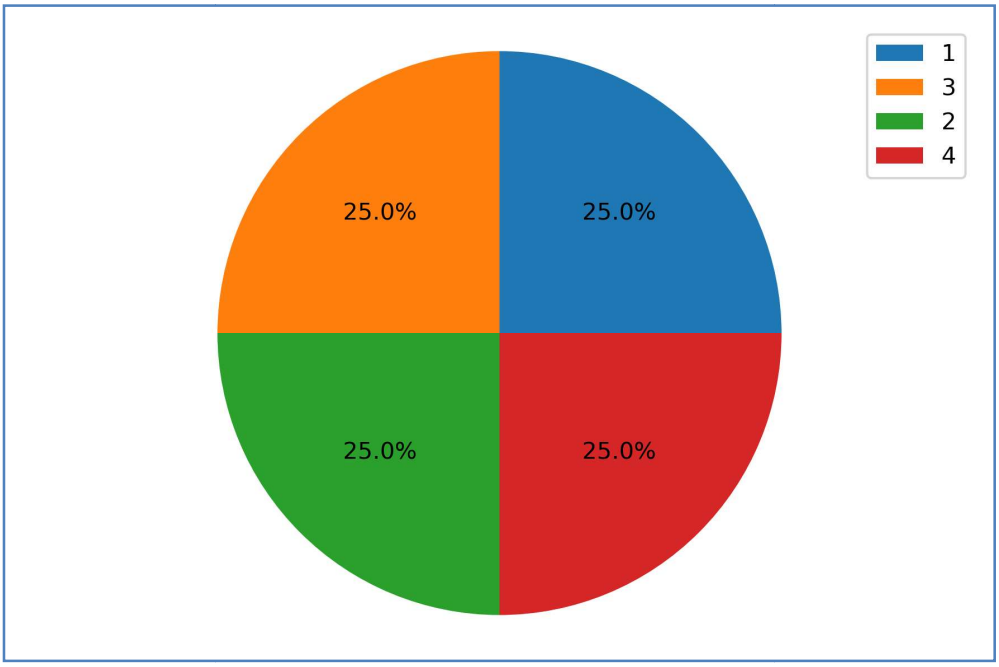
Image Array	[[[[0.0000e+00] [0.0000e+00] [0.0000e+00] ... [4.9176e+04] [4.9176e+04] [1.6184e+04]]	[[2.4000e+01] [2.4000e+01] [2.4000e+01] ... [2.3000e+01] [2.3000e+01] [1.6183e+04]]	[[1.6424e+04] [1.6407e+04] [1.6407e+04] ... [2.4000e+01] [2.4000e+01] [1.6184e+04]]	...	[[2.9725e+04] [2.9725e+04] [2.9725e+04] ... [2.2045e+04] [2.2045e+04] [1.6285e+04]]	[[2.3581e+04] [2.3581e+04] [2.3581e+04] ... [2.8701e+04] [2.8701e+04] [1.6285e+04]]	[[2.5629e+04] [2.5629e+04] [2.5629e+04] ... [1.4879e+04] [1.4879e+04] [-1.6465e+04]]]]
AJCC Staging	1	1	3		1	1	2

Figure 46: An instance of image arrays and their corresponding class labels (AJCC Staging)

Similar datasets may be developed by using the same image collection and by altering class variables as per need. At last, different varieties of Synthetic Minority Over-sampling Technique (SMOTE) [143] have been applied to balance class distribution.



(a)



(b)

Figure 47: Ratio of samples under different class (a) before and (b) after resampling [1=> AJCC Stage I; 2=> AJCC Stage II; 3=> AJCC Stage III; 4=> AJCC Stage IV]

First, the oversampling has been done with the minority classes and then resampling has been performed with borderline classes. At last, resampling has been done by using all the classes including both majority and minority classes. This has balanced the class distribution and helped in generating a consistent result.

7.5 Conclusion

This chapter describes the database preparation from features extracted in the earlier phase of the study. Vector or sequence processing is an area where deep learning is a bit underexplored than the traditional machine learning methods. The present research work has prepared a manual dataset of rank one tensor which may be fed into a one-dimensional CNN and other leading machine learning techniques. This has been done by fetching pathological staging or grading information from the clinical data and tagging those with the respective feature records. For fully automated DNN, the database has been prepared by mapping clinical information with the image array. Separate databases have been prepared for TNM staging, AJCC staging, histopathological grading, and histological subtypes.

8. Development of a One-dimensional CNN Model[6]

In the present research work, a one dimensional CNN model has been developed to determine the stage and grade of any given tumor image. To achieve the goal, different varieties of deep learning techniques have been used. Features have been extracted from segmented images and then the database has been developed by clubbing different staging and grading information with the extracted feature set. The dataset thus prepared is then standardized and normalized and also resampled. After that, the resampled data has been fed into the CNN model and its varied derivatives. Other leading machine learning algorithms are also trained and tested with the final dataset. Results have been recorded and evaluated accordingly. A comparative study has also been conducted in this regard to analyze the efficacy of the newly developed CNN model.

8.1 Methodology

Different machine learning techniques have been used in this part of the study. This includes One Dimensional Convolutional Neural Network (1D CNN), Gated Recurrent Unit (GRU), Support Vector Machine (SVM), Fuzzy Rough Nearest Neighbour (FRNN), MultiLayer Perceptron (MLP), Logistic Regression (LR), and Random Forest (RF).

8.1.1 One Dimensional Convolutional Neural Network (1D-CNN)

A convolution takes two vectors or matrices (all are tensors of different rank) as input – one of them is the original input and the other is the kernel or filter (24). On injecting the input in an activation function, a vector is produced as the output. The one-dimensional convolution may be defined as:

$$(f*g)(i)=\sum_{j=1}^{m} g(i).f(i-j+\frac{m}{2}) \quad \text{…Equation 8.1.1.1}$$

Where f is the input vector of length n and the filter g has length m,

The kernel navigates across the input tensor iteratively as per some predetermined offset. The product of the overlapping values of the kernel and input tensor are being summed up. This sum of element-wise multiplications will be the output. The one-dimensional convolutional layers may be defined as:

$$y^l = b^l + \sum_{i=1}^{n_{l-1}} (f^{l-1} * g^{l-1})(i) \quad \text{…Equation 8.1.1.2}$$

[6] Based on author's publication no. 3 and no. 4 [Appendix D]

Here, y^l is the l^{th} layer output, b^l is the l^{th} layer bias, f^{l-1} is the $(l-1)^{th}$ layer output and g^{l-1} is the $(l-1)^{th}$ layer filter. The dense layer output O may be expressed as:

$O = b + \sum_{i=1}^{n} w^{(i)} x^{(i)}$ …Equation 8.1.1.3

Where b is layer bias, w is weight vector and x is input component

The prediction $\hat{y}$ was done by the softmax activation function:

$\hat{y} = \text{Sotmax}(O)$ …Equation 8.1.1.4

Where *Softmax* = $(\exp(o_i)/\Sigma_j \exp(o_j))$ and o_i is the level of confidence for belongingness to category i. The cross-entropy loss is calculated as:

$l(y, \hat{y}) = -\sum_j y_j \log \hat{y}_j$ …Equation 8.1.1.5

Here, y is the actual and $\hat{y}$ is the predicted value.

8.1.2 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) [144] is a lightweight evolved version of simple Recurrent Neural network (RNN) and its performance is also stays on the higher side. GRU adds two more gates to the simple RNN architecture and they are called reset and update gates:

$R_t = ø\ (X_t W_{xr} + H_{t-1} W_{hr} + b_r)$

$U_t = ø\ (X_t W_{xu} + H_{t-1} W_{hu} + b_u)$ ...Equation 8.1.2.1

Here, ø is the activation function. Implementation of reset and update gates are as follows:

$H_t = ø\ (X_t W_{xh} + (R_t H_{t-1}) \Theta W_{hh} + b_h)$

$H_t = U_t \Theta H_{t-1} + (1-U_t)\ \Theta H_t$ …Equation 8.1.2.2

Where Θ is the element-wise multiplication. These reset and update gates act like memory cells which help a new hidden state in acquiring from an old hidden state.

8.1.3 Support Vector Machines (SVM)

A Support Vector Machine (SVM) is a linear discriminant that separates data into classes using a hyperplane having the maximum-margin [145]:

$f(y) = w^T y$ …Equation 8.1..3.1

Where, y results from applying a nonlinear transformation to the data; dot products are replaced by nonlinear kernel functions; the hyperplane w maximizes the margin between the transformed classes.

8.1.4 Fuzzy Rough Nearest Neighbour (FRNN)

The Fuzzy Rough Nearest Neighbour (FRNN) is a hybrid of Fuzzy Logic [146], Rough Set theory [147], and Nearest Neighbour algorithm [148]. The Fuzzy Logic states about the degree *k* ($0 \leq k \leq 1$) of belongingness of an element to a set. This contradicts the definite belongingness of an element to a particular set as proposed in the Cantor's classical set theory. The fuzzy membership function may be represented by the expression:

$$\mu_X(x) \in [0,1] \qquad \text{…Equation 8.1.4.1}$$

Where *X* is a set and $x \in X$. The Rough Set theory tells us about the indistinctness in decision making [149]:

R-lower approximation of X:

$$R_*(x) = \bigcup_{x \in U} \{R(x) : R(x) \subseteq X\}$$

R-upper approximation of X:

$$R^*(x) = \bigcup_{x \in U} \{R(x) : R(x) \cap X \neq \varnothing\}$$

R-boundary region of X:

$$RN_R(X) = R^*(x) - R_*(X) \qquad \text{…Equation 8.1.4.2}$$

Thus, the lower approximation contains the elements that belong to the concerned set and the upper approximation contains the elements that may belong to the concerned set. The boundary region is the difference between the two approximations. If the boundary region is empty, the set is a rough set or crisp, otherwise. The Rough Set theory is helpful in machine learning as it does not need any prior knowledge about data like possibility value in Fuzzy. Fuzzy and Rough Set theories often work in tandem and get more accurate results in data analysis. In such hybridizations, the rough approximation of fuzzy closeness is observed:

$$\mu_{\bar{A}(X)}(x) = \sup_{w \in X} \mu_A(x,w)$$
$$\mu_{\underline{A}(X)}(x) = \inf_{w \notin X} \mu_A(x,w) \qquad \text{…Equation 8.1.4.3}$$

Where A is a fuzzy indiscernibility relation and x ϵ U.

K-nearest neighbor (KNN) selects a near-optimal value of K based on cross-validation. Thus, the clubbing Fuzzy-Rough algorithm with KNN may boost up the accuracy significantly, especially, in case of any modular classification problem [150]. The typical FRNN membership function may be described by the following expression [151]:

$\tilde{\mu}_y(x) = 1/(1+k||x-y||^{2/(q-1)})$ …Equation 8.1.4.4

Where q describes the shape of the membership function, ||x-y|| is the Euclidean distance between y and x and k is the bandwidth of the membership.

8.1.5 MultiLayer Perceptron (MLP)

A perceptron may be described as follows:

$f_j(x,w) = \mathrm{sgn}(\sum_j w_{ji}x_i - \theta_j)$ …Equation 8.1.5.1

First, the product of input x_i and their corresponding weight w_i has been summed up. Subsequently, the difference between this sum and the threshold θ_j is calculated. This expression passes through an activation function and produces the output f_j. Several layers of such perceptrons are clubbed together to form the Multilayer Perceptron (MLP). Here, the output of one layer becomes the input to another and the final layer generates the class scores.

8.1.6 Logistics Regression (LR)

If k classes exist with n instances and m features, the parameter matrix B will have a dimension of m*(k-1) [152]. The probability for class j excluding the last class is:

$$P_j(X_i) = \frac{e^{X_i B_j}}{\sum_{j=1\ldots(k-1)} e^{(X_j B_j)+1}}$$ …Equation 8.1.6.1

The last class has got a probability:

$$1 - \sum_{j=1\ldots(k-1)} P_j(X_i) = {}^{1}\Big/_{\sum_{j=1\ldots(k-1)} e^{(X_i B_j)+1}}$$ …Equation 8.1.6.2

The (negative) multinomial log-likelihood is thus:

$$L = -\sum_{i=1\ldots n}\Big[\sum_{j=1\ldots(k-1)}(Y_{ij} * \ln(P_j(X_i)) + (1 - \sum_{j=1\ldots(k-1)} Y_{ij}) * \ln(1 - P_j(X_i))\Big] + ridge * B^2$$ …Equation 8.1.6.3

The optimized values of the m*(k-1) variables have been searched heuristically to detect matrix B with minimized L. Before using the optimization procedure, the matrix B is converted into an m*(k-1) vector.

8.1.7 Random Forest (RF)

It constructs a forest of random trees [153]. The algorithm is as follows:

1. *Pick random samples from a set of data.*
2. *Create a decision tree for each instance of a sample.*
3. *Retrieve the prediction result from every decision tree.*
4. *Voting is performed for every predicted result.*
5. *The most voted prediction will be considered as the final result.*

8.2 Experiment

A prolonged experiment has been carried out and the One Dimensional Convolutional Neural Network Model (1D CNN). In the initial CNN model (Figure 48), the input layer comprised a kernel equal to the input image resolution and in each subsequent layer, the kernel has been halved by the *maxpooling* layer. Default stride implies that all the data points have been considered during pooling. All the convolutional layers have the default padding i.e., the input is bounded by the padding of cells with zero values. L2 or Euclidean Norm regularizers have been used to prevent overfitting. After three hidden conv1d layers, the output has been flattened and fed into two subsequent fully connected dense layers. Here, the kernel of the first dense layer has been increased to match the image resolution and the kernel of the last dense layer is equal to the number of output, e.g., for grading it is equal to 4 as a 4-tier grading system is used in the experiment. Adam optimizer has been used with a learning rate of 1e-4. The batch size used is 128; the Rectified Linear Unit (RelU) is used as the activation function for all layers and Softmax is used as the activation function for the final output layer, respectively. All the hyper-parameters have been determined by performing repeated experiments.

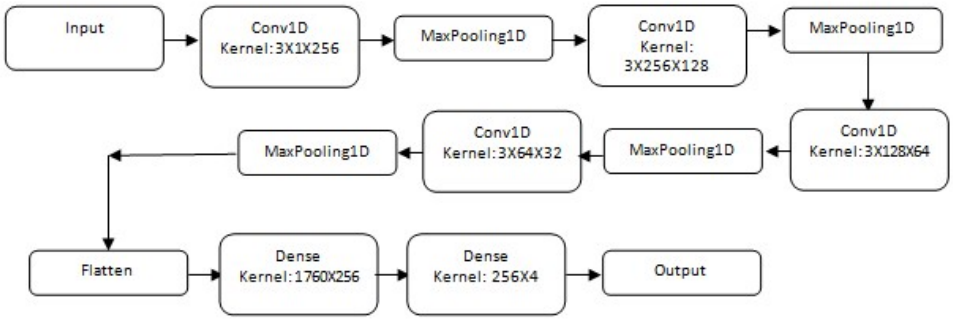


Figure 48: The initially developed 1D CNN model

1-D convolutions extract subsequences from sequences. Larger convolution windows can be used with 1-D CNN which is not possible with a traditional 2-D convolution layer. 1-D CNNs are also faster than 2-D CNN. A 1-D CNN can offer a computationally inexpensive alternative to a 2-D convent. While carrying out experiments it has been observed that the overall accuracy of the model gets improved whenever CNN acts as a preprocessor before an RNN. A Recurrent Neural Network (RNN) has a feedback path which helps in memorizing the earlier incidents, whereas a CNN does not have such a memory cell. This may boost up (Figure 49) the overall accuracy of the system. Thus, the new model becomes a memory enabled CNN which is a combination of the speed and lightness of CNN and the order-sensitivity of RNNs.

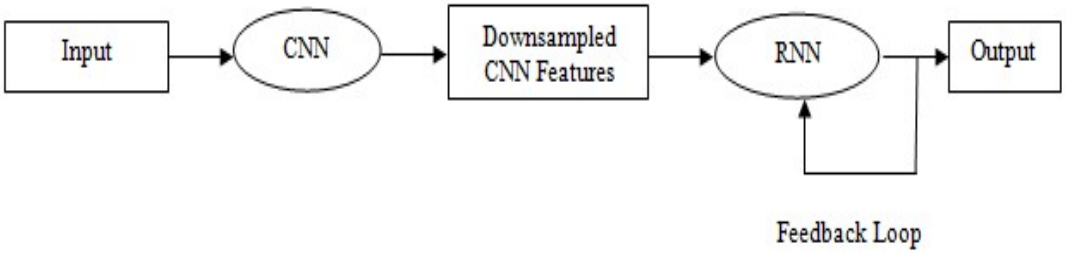


Figure 49: Structure of CNN-RNN Combined Model

The initial model is updated by adding GRU layers. The spatial information from the segmented tumor image has been retrieved, merged with the clinical staging and grading information, and then used with the CNN-RNN hybrid model to classify the information. The present research work has given its dataset a shape of a rank 1 tensor. A rank 1 tensor has a typical two-dimensional data structure, e.g. (row, column). A rank 2 tensor has a 3-D data structure, e.g. (row, column, color channels). Being a computationally cheap alternative to a recurrent network, a set of one-dimensional convolutional layers is used as preprocessing layers whose output was fed into an RNN. The CNN-RNN model comprises a one-dimensional convolutional input layer followed by three hidden one-dimensional convolutional layers and one GRU layer. Each convolutional layer is succeeded by a 1-D *MaxPooling* layer. The CNN output is fed to a lightweight GRU layer. The dense layer has a 2-D structure i.e. (output from the preceding layer, filter size). On the other hand, 1-D convolutional layers, or GRU layers have 3-D structure i.e. (output from the preceding layer, filter size, number of filters). As the dense layers and

their predecessors are incompatible, a flatten layer has been introduced in between so that, the output from the GRU layer could be fed into the dense layer. Thus, the GRU layer is succeeded by one flatten and one fully connected dense layer (Fig. 50). The following fully connected layer considers all the possible dot products. At last, there is an output layer with a sigmoid activation function for calculating the class scores.

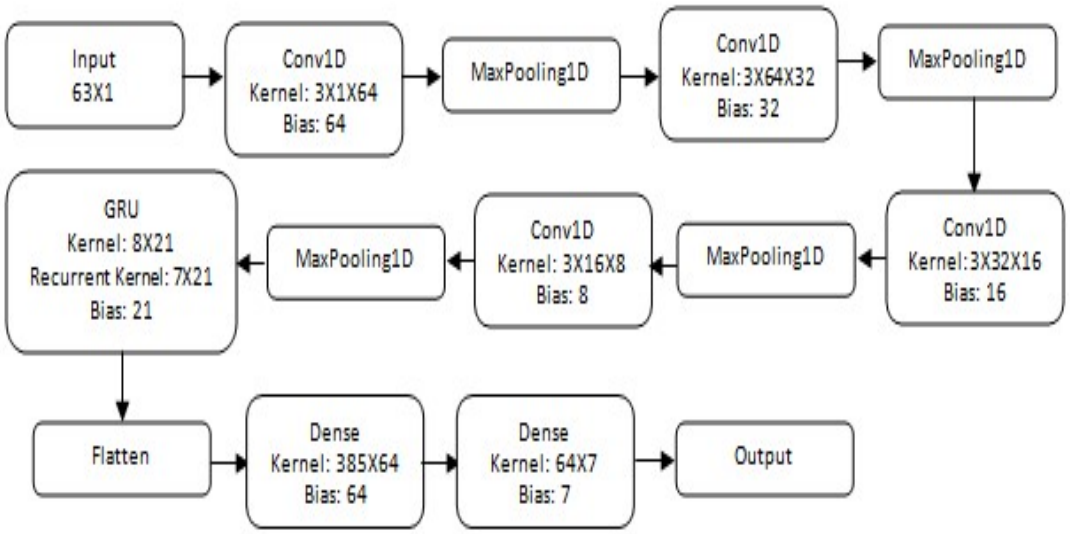


Figure 50: Finally developed CNN-RNN model

The training and test accuracy for each of the T stage, N stage, M stage, and grade have been separately measured and the results have also been compared with a few of the leading machine learning techniques: Support Vector Machines (SVM), Fuzzy Rough Nearest Neighbour (FRNN), Random Forest (RF), Logistic Regression (LR), and Multilayer Perceptron (MLP).

8.3 Evaluation

The base of the evaluation has been the confusion matrix which has four quadrants: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). This resembles the famous McNemar's statistical test that yields a contingency table as output by comparing before and after status while conducting the test.

		Predicted Value	
		True	False
Actual Value	True	True Positive	False Negative
	False	False Positive	True Negative

Figure 51: Structure of a typical Confusion Matrix

According to figure 51, if the actual value is *true* and the predicted value is *true* as well, the output is *true positive*; if the actual value is *true* and the predicted value is *false*, the output is *false negative*; if the actual value is *false* and the predicted value is *true*, the output is *false positive*; if the actual value is *false* and the predicted value is *false* as well, the output is a *true negative*. True positive shows the rate of acceptance; true negative shows the rate of rejection; false-positive shows the type-I error or false alarm; false-negative shows rate of miss or type-II error. Thus, the confusion matrix shows not only classification errors but also different types of classification errors. Such a confusion matrix may be applied to a binary as well as a multiclass problem. Many useful evaluation parameters may be derived after giving the class scores a shape of a confusion matrix. An important evaluation metric is the Receiver Operating Curve (ROC) which is ROC is a plot of the true positive rate against the false-positive rate. Some other major evaluation metrics are [154], [155]:

Accuracy = (TP+TN)/ (TP+TN+FP+FN) …Equation 8.3.1

Error Rate = 1 – Accuracy …Equation 8.3.2

True Positive Rate or, Recall = TP/ (TP+FN) …Equation 8.3.3

Precision = TP/ (TP+FP) …Equation 8.3.4

False Positive Rate = FP/ (FP+TN) …Equation 8.3.5

F-Measure = (2.Precision.Recall)/ (Precision+Recall) …Equation 8.3.6

RMSE = $\sqrt{\overline{(f-o)^2}}$ …Equation 8.3.7

Where, f = forecasts (expected values or unknown results) and o= observed values (known results)

Kappa = $(p_o - p_e)/(1 - p_e)$ …Equation 8.3.8

Where p_o is the relative pragmatic conformity and p_e is the assumed probability of conformity [156]. Kappa statistics closer to one, better is the classification result

A cost-sensitive evaluation has also been carried out simultaneously to prohibit the chances of getting a biased confusion matrix. The $(i, j)^{th}$ element in the cost matrix is the penalty for wrongly classifying an instance of class j as class i, e.g., for grading the penalty matrix would look like figure 52.

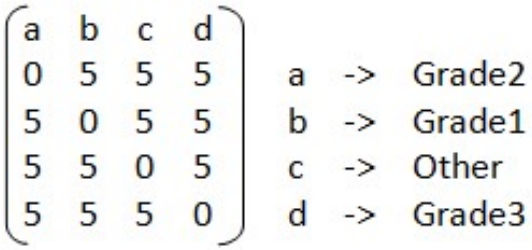


Figure 52: An example of the cost matrix used for grading

It has often been seen that accuracy alone may not stand as a trustworthy evaluation parameter as it assigns equal weight to both kinds of errors. There may be many impurities in the dataset including class biasness. So, other auxiliary parameters have been used to demonstrate the efficiency of the model developed affirmatively. High recall means less false negative or less type-II errors. This also means high true positives i.e. rate of hit is higher than the rate of miss. On the other hand, high precision means less false positive or less type-I error. This also means a high true positive rate. A type-I error means rejecting a true null hypothesis and Type-II error means accepting a false null hypothesis. Both the errors put an equal barrier before the correct conclusion. Precision and recall depict type-I and type-II errors respectively. Individually, they may be treated as good predictors of classification errors. However, they exist at one another's cost. This implies that both precision and recall cannot be high arbitrarily at the same time. On the contrary, F_β-score is a combination of both precision and recall. It can also measure classification errors correctly more than accuracy does. High recall, low precision implies that most of the true samples are properly acknowledged (low FN) but there is a chance of having significant false positives. On the other hand, low recall, high precision implies that many true samples have been overlooked (high FN) but the instances that have been classified as true are indeed true cases. This means the result has fewer False Positives.

F_β-Score makes a balance between precision and recall by using their harmonic mean. Thus it punishes the outliers and reaches at its best when $\beta=1$ that turns it into F_1-Score. Area Under Curve (AUC) fetched out of ROC provides a collective measure of performance against all possible classification threshold values. AUC may also be explained as the likelihood that the model ranks a random positive instance more highly than a random negative instance. AUC closer to 1 depicts better classification results. Thus, all the evaluation metrics used in the study are capable of determining the classification errors far more effectively and portrays a bias-free result which more trustworthy.

The simple one-dimensional CNN model is quite successful in identifying the TNM stage and histopathological grade of NSCLC tumor images (as described in the author's first full paper in Appendix E). However, during experimentation, it has been felt that features need to be re-incurred while moving ahead with CNN down-sampling. This triggers the embedding RNN after CNN pre-processing layers.

8.4 Conclusion

In this chapter, the process of developing a new one-dimensional deep learning model by combining CNN and GRU has been described. Techniques used in the study, details of experiments carried out, and methods of evaluation of outcome have also been described. Almost all the traditional machine learning methods efficiently do vector processing. Deep learning is capable of processing high order data but has not yet delved into the vector processing domain. The present research work has pioneered in developing such a model that can process vector or rank one tensor with equal efficiency as that of the leading machine learning models.

9. Development of a Two-dimensional CNN Model[7]

So far, the development process of a 1D CNN model has been discussed. Such a model is semi-automated as the features are being crafted manually. This model is useful when the hardware resources are scarce and users are dependent on CPUs only. When GPUs are available through advanced graphics cards or cloud services, users may switch to a fully automated system. The present research work has also developed such a system by introducing the same technique that has been used while developing the aforementioned 1D CNN model. The 1D CNN model is developed by using a combination of CNN and Gated Recurrent Unit (GRU) layers, whereas, in the fully automated model there is a subtle difference in implementation and technology used. In essence, both the semi-automated and the automated CNN models have been developed by using the same technology. The difference is only with the implementation and the technology found conducive for the respective mode of implementations. In the latter case, a two-dimensional Convolutional Neural Network technique has been used in conjunction with the Long Short Term Memory (LSTM) [157] – a more traditional form of Recurrent Neural Network (RNN). In particular, the model has been developed by combining CNN and bidirectional RNN (Figure 53). The bidirectional wrapper embedded in the model helps to fetch information both from forward and backward pass. In this way, the model developed becomes more accurate and robust in classifying tumor stage and grade.

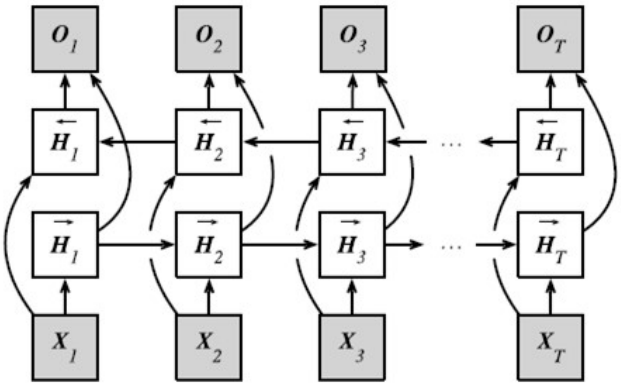


Figure 53: Structure of a typical bidirectional recurrent neural network with one hidden layer [Source: https://d2l.ai/]

[7] Based on author's publication no. 2 and no. 9 [Appendix D]

The second variety is a non-sequential recurrent ensemble of CNNs. Here, two important concepts have been merged and used together: inception or branched network [158] and residual [159] or reinjection of earlier layers. An inception model is a subgraph of convolutional layers with several parallel branches that learn spatial features separately and a residual connection is the reinjection of prior information downstream via feature-map addition. In the inception model, each initial convolutional layer is a point-wise convolution which computes features that mix information across channels of the input tensor, but never mix information across space. In the residual model, even if the output of a layer becomes smaller after activation or down-sampling, it may be regenerated by a reinjection from the original layer output.

9.1 Methodology

A time distributed convolutional model having (h*w) input image array having c channels and k layers may be expressed as:

h_t [i, j, k] = $\sum_{a=-\Delta}^{\Delta}\sum_{b=-\Delta}^{\Delta}\sum_{c} w[a,b,c,k].x[i+a,j+b,c]$ …Equation 9.1.1

In equation 9.1.1, x [i, j] and h [i, j] represent an image and hidden state respectively; w stands for the weight parameter; a & b are offset belong to the range [-Δ, Δ] for each pixel location (i, j); t is the time component across which the convolution takes place. The output shape $[n_h\text{-}k_h+1]*[n_w\text{-}k_w+1]$ is given by the difference (1 is the bias) between the input shape $[n_h*n_w]$ and the kernel shape $[k_h*k_w]$. By considering c_i as input channels and c_o as output channels, the output shape will be $ci*c_o*[n_h\text{-}k_h+1]*[n_w\text{-}k_w+1]$.

Equation 9.1.1 may be re-written as equation 9.1.2 for describing the inception model:

$O_l^{(i)} = f(W_l^{(i)}X_l + b_l^{(i)})$ … Equation 9.1.2

Here, *X* is the input vector; *W* is the weight vector; *b* is the bias; *l* is the corresponding layer number; *O* is the output; *i* is the inception branch number and *f* (.) is the non-linear activation function.

The residual layer may be enumerated as equation 9.1.3:

$X_{l+1} = O_l^{(i)} + X_l$ … Equation 9.1.3

The concatenation of inception and residual layers may be described by the equation 9.1.4:

$Y = g_c(\{F_j(W,X)\})$ … Equation 9.1.4

Where $\{F_j\ (.)\}$ is the collection of output tensors emanating from j (j=1, 2… n; n>0) inception branches and g_c is the concatenation operation w.r.t. axis c i.e. number of colour channels. This concatenated output has to pass through the *timedistributed flatten* layer which is necessary for initiating the *timestep* parameter required in the following recurrent layers.

The output of time distributed CNN can easily be fed into an RNN. A recurrent network can learn from its previous iterations and may be defined as:

$$H_t = \emptyset(X_t\ W_{xh} + H_{t-1}\ W_{hh} + b_h) \quad \text{…Equation 9.1.5}$$

In equation 9.1.5, H_t is the hidden state at time t; ø is the activation function; X_t is the mini-batch instance with sample size n and d inputs; W_{xh} is the weight parameter with h is the number of hidden states; H_{t-1} is the hidden state from the previous time-step. Its weight and bias parameters are W_{hh} and b_h, respectively.

In equation 9.1.6, O_t depicts the output (q is the number of outputs):

$$O_t = H_t W_{hq} + b_q \quad \text{…Equation 9.1.6}$$

Now, Long Short Term Memory (LSTM) [31] is a gated version of RNN that has been used along with a bidirectional wrapper. This will help the proposed model to learn from both sides, and the efficacy of the model will stay on the higher side. A typical LSTM may be described as:

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i) \quad \text{…Equation 9.1.7}$$

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f) \quad \text{…Equation 9.1.8}$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \quad \text{…Equation 9.1.9}$$

In equations 9.1.7, 9.1.8 and 9.1.9, I_t, F_t, O_t are the input gate, forget gate, and output gate, respectively. Inputs are processed by a fully connected layer with a sigmoid (σ) activation function and all the gates lie within a range of [0, 1].

Bidirectional RNN may be expressed as:

$$\vec{H}_t = \emptyset(X_t\ W_{xh}^{(f)} + \vec{H}_{t-1}\ W^{(f)}{}_{hh} + b^{(f)}{}_h) \quad \text{…Equation 9.1.10}$$

$$\overleftarrow{H}_t = \emptyset(X_t\ W_{xh}^{(b)} + \overleftarrow{H}_{t-1}\ W^{(b)}{}_{hh} + b^{(b)}{}_h) \quad \text{…Equation 9.1.11}$$

The recurrent nature of the model may be shown as:

$$\overrightarrow{H}_t^{(l)} = f(X_t, \overrightarrow{H}_{t-1}^{(l)}) \quad \text{…Equation 9.1.12}$$

$$\overleftarrow{H}_t^{(l)} = f(\overleftarrow{H}_{t-1}^{(l-1)}, \overleftarrow{H}_{t-1}^{(l)}) \quad \text{…Equation 9.1.13}$$

$O_t = g(\overleftrightarrow{H}_t^{(L)})$ …Equation 9.1.14

In equations 9.1.12, 9.1.13, and 9.1.14, X_t is the mini-batch input; t is the timestamp; l is the layer number; f is the layer activation function and g is the activation function for the output layer O_t. Here the forward and backward hidden states are merged to substantiate the hidden state $\overleftrightarrow{H}_t^{(L)}$ and passed on as input to the next bidirectional layer. The final layers are given by the equations 9.1.15, 9.1.16 and 9.1.17:

$h=w_hO_t+b_h$ …Equation 9.1.15

$o=w_oh+b_o$ …Equation 9.1.16

The prediction may be measured as $\hat{Y}$=softmax (o) …Equation 9.1.17

In equation 9.1.17, $\hat{Y}_i = (\exp(o_i)/\Sigma_j \exp(o_j))$ and o_i is the level of confidence for belongingness to category i. The loss is measured as the cross-entropy loss and is given by equation 9.1.18:

$l(Y, \hat{Y}) = - \Sigma_j Y_j \log \hat{Y}_j$ …Equation 9.1.18

Softmax is the activation function for the final layer.

9.2 Experiment

In the present study, two varieties of Deep Neural Network (DNN) have been tried. The first one is a typical sequential supervised Spatio-temporal method containing a hybrid of time distributed CNN and bidirectional RNN. The second one is a non-sequential recurrent ensemble of DNN. As bidirectional RNN needs data from both forward and backward pass, the bidirectional LSTM layers are preceded by hidden time distributed conv2D layers and succeeded by flattened dense layers. Many of the data pre-processing like feature selection, or segmentation are accomplished implicitly by the convolutional layers. Such as edge detection is done by using the *sobel* filter. Maxpooling layers down-sample the feature set to have fewer numbers of dot products along the spatial dimensions. The output of the convolutional layer is flattened and fed into the bidirectional LSTM layers and the output is injected in a dense layer which is succeeded by the fully connected layer. Softmax activation function measures the probability distribution, and the loss is measured by categorical cross-entropy. All the codes are

written and implemented by using Python 3.6.8 (IPython 7.5.0) on CPU cores of an Intel(R) Core(TM) i5-3230m CPU @ 2.60GHz processor (x64 based processor)[8].

The same line of an experiment as that of 1D CNN has been followed while developing the desired two dimensional CNN model. All the images are resized to 64*64 for better memory utilization. After getting converted to pixel array, the input dataset typically takes the shape of rank 4 tensor: (number of samples, image height, image width, number of color channels). From the available clinical data, the AJCC label corresponding to each tumor has been tied with the respective image array. The whole dataset has been compressed and loaded for the experiment. The class imbalance issue has been addressed by Synthetic Minority Over-Sampling (SMOTE).

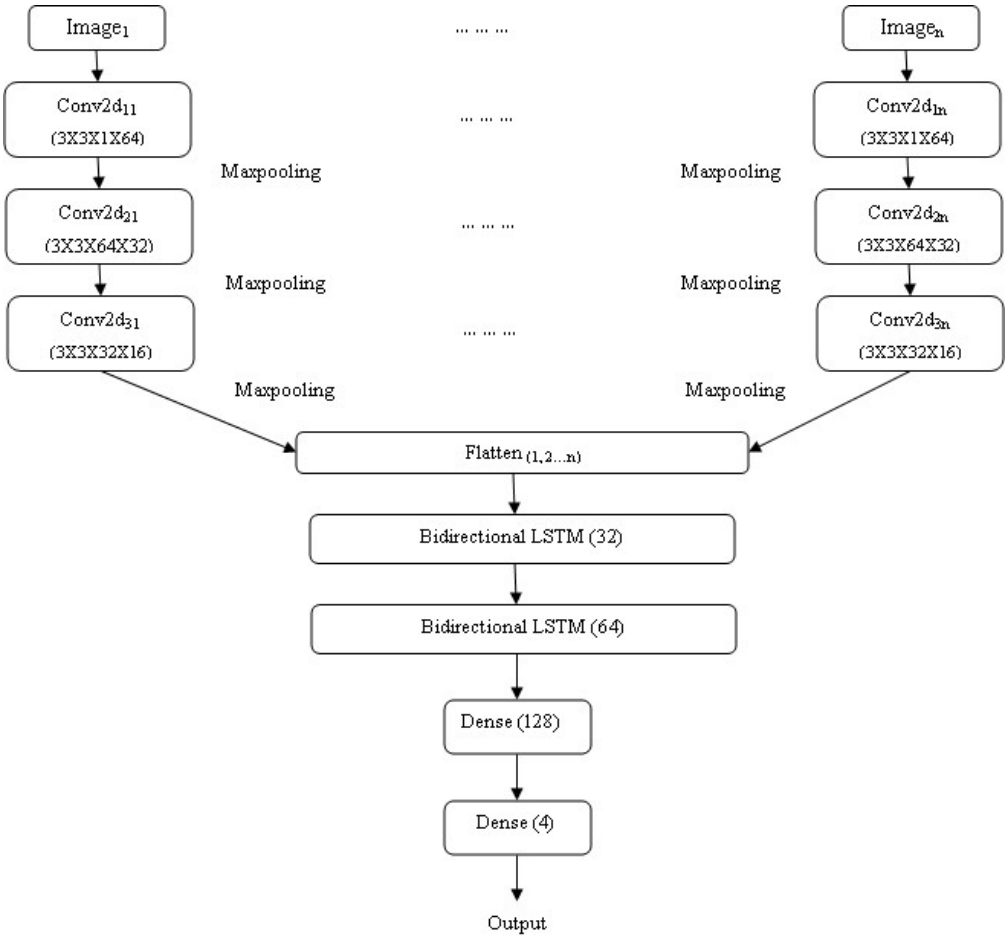


Figure 54: Structure of the CNN-BiRNN model ensemble with sequential image input to the time distributed convolutional layer for applying CNN layer to every temporal slice of the input

[8] Tools & platform have been selected based on author's publication no. 1 & 2 [Appendix C]

In the present study, three key sequential models are used. The first one is the single CNN+BiRNN model (Figure 55) which is the combination of Convolutional Neural Network (CNN) and the Bidirectional Recurrent Neural Network (BiRNN). During the training phase, best models are saved and later ten such best models are averaged to form the CNN+BiRNN ensemble (Figure 54). Another model used is a simple CNN model having similar architecture to the proposed CNN+BiRNN model (Figure 56).

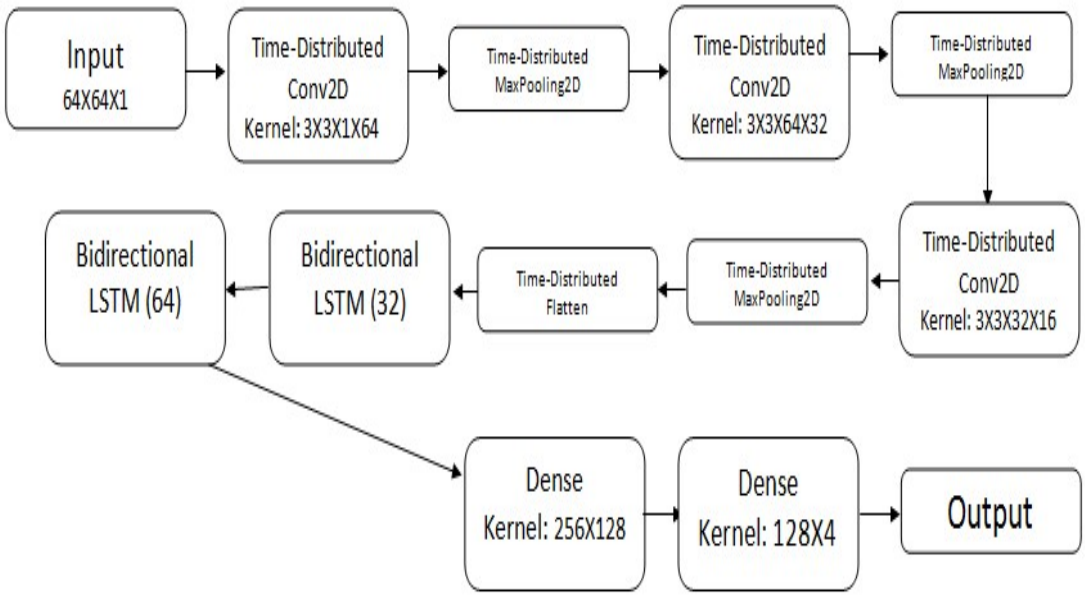


Figure 55: Structure **of a** single CNN+BiRNN model used in the study

The single CNN+BiRNN model comprises three time-distributed conv2d layers with filters, 64, 32, and 16 respectively. Conv2d layers have ReLU as a layer activation function with the default padding and strides. Each conv2d layer is followed by a time-distributed maxpooling layer and before inserting in bidirectional LSTM, the output is vectorised by a time-distributed flatten layer. There are two bidirectional LSTM layers with kernel 32 and 64, respectively. Recurrent layers are followed by a fully connected dense layer having filter 128. At last, there is an output layer. The CNN+BiRNN model ensemble has the same architecture like that of the single CNN+BiRNN model. The simple CNN model has three conv2d layers with kernels 64, 32, and 16, respectively. Unlike the CNN+BiRNN model, it does not have recurrent layers; instead, it has two

succeeding dense layers with kernel 32 and 64, respectively. Then there is a flatten layer followed by a fully connected dense layer having kernel 128. All the dense layers are followed by maxpooling layers. In the end, there is the output dense layer.

While experimenting, a 10-Fold Stratified Cross-validation is employed on the dataset. The cross-validator repeats itself 10 times with different randomization. This ensures that the train and test split has the minimum correlated data. The test class is converted into categorical and the dataset is normalized by rescaling. Hyper-parameters used are: batch size=128, learning rate=1e-1, optimizer=Adam. The experiment is carried out for 5000 epochs with early stopping callback with patience value =200. The best models are saved and later an average ensemble of ten such best models is formed. The ensemble model is executed with 10-Fold Repeated Stratified Cross-Validation. The simple CNN model is executed with 10-Fold Cross-Validation and other parameters remain the same as that of the CNN+BiRNN model.

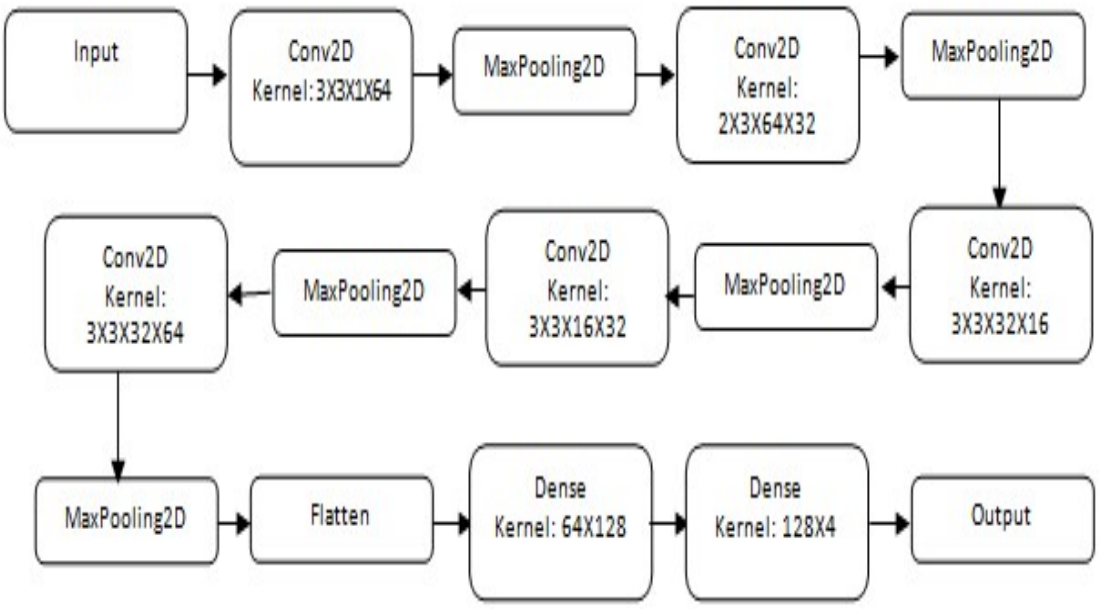


Figure 56: Structure of a CNN having architecture similar to the CNN-BiRNN model

The simple CNN model, the single CNN+BiRNN model, and the CNN+BiRNN model ensemble have also been employed to classify NSCLC histological subtypes (Author's second full paper as listed in Appendix E). Among them, the CNN+BiRNN model has

performed satisfactorily. Similar models have been applied here to figure out the stages and grades of the varied mixed of tumor images. As experiments continue, the need for preserving the features are cropped up. After thorough inculcation of the existing techniques, it has been felt that there has to be some way to retain strong features as the downsampling continues through iterations. This has paved the way for breeding a non-sequential counterpart of the CNN+BiRNN model ensemble.

In the case of the non-sequential model (Figure 57), the input tensor is fed simultaneously into three parallel inception branches of varying layers and kernels. The first branch starts with a (1X1) pointwise convolution layer with filter size 32 followed by a (3X3) conv2d layer with filter 16. These layers are succeeded by batchnormalisation and maxpooling layers. The second branch also starts with a (1X1) layer having filter 32 followed by a (3X3) conv2d layer with filter size 8. These are followed by batchnormalisation and maxppoling layers. The third branch starts with a maxpooling layer and it is followed by a (3X3) conv2d layer with filter size 8. These layers are succeded by batchnormalisation and maxpooling for layer normalisation and downsampling spatial features (Figure 58). The third branch is continued with another (5X5) conv2d layer with a filter 16 and is being reinjected by a residual layer of (1X1) having filter 16. The input to the residual layer is input from the third branch. This acts as the fourth branch of the model. All the branches are concatenated and flattened by a timedistributed *flatten* layer. The flatten layer output is passed through two bidirectional LSTM layers and one fully connected dense layer having kernels 16, 32, 64, respectively. At last, the output dense layer calculates the class score by using the *softmax* activation function. The optimizer used is *adam* with a learning rate $1e^{-4}$, and each conv2d layer is regularized by L2 or euclidean norm. The loss is being measured by using categorical cross-entropy. Default padding nad strides have been used in the model.

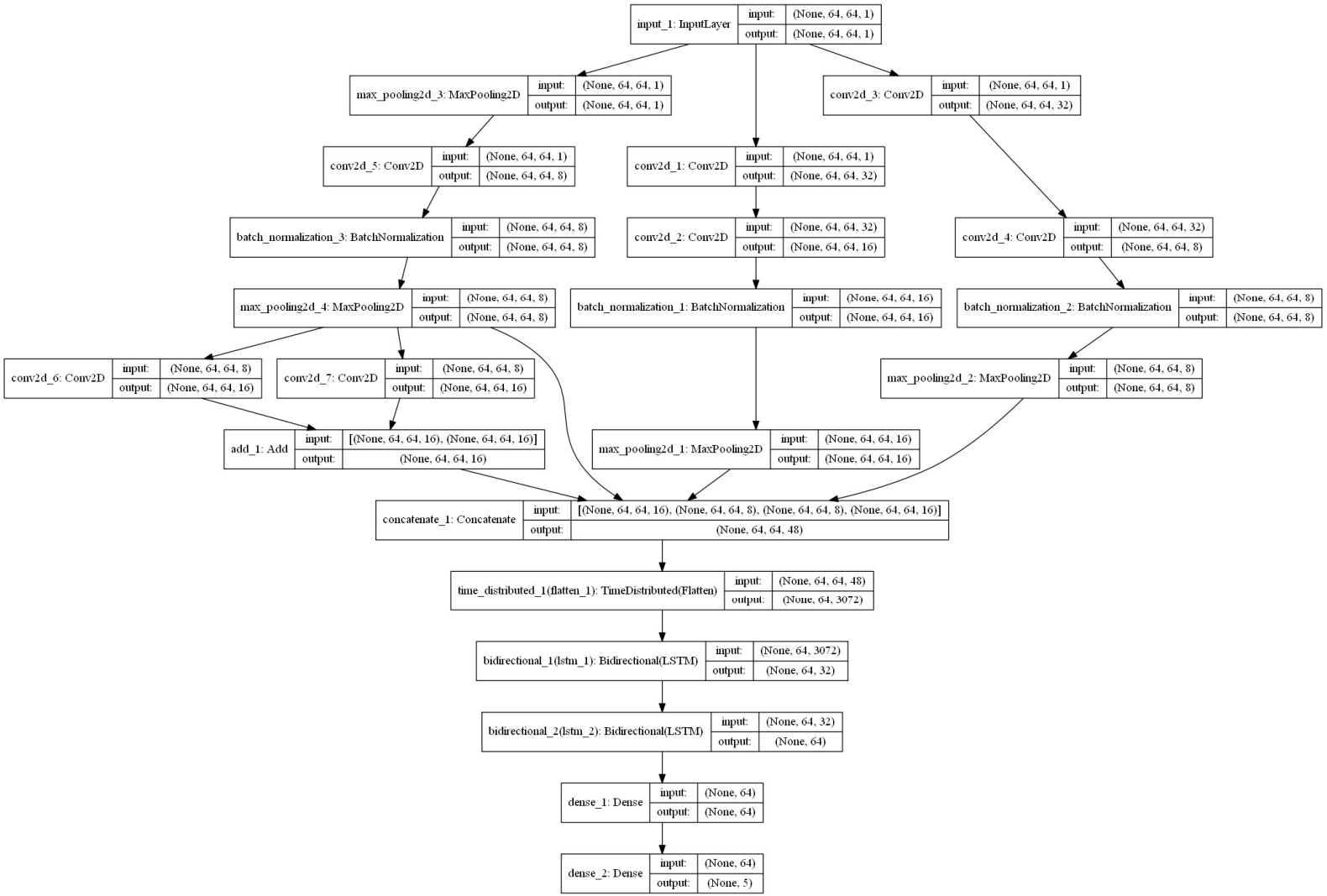


Figure 57: Non-sequential model as a combination of inception, residual, and recurrent techniques

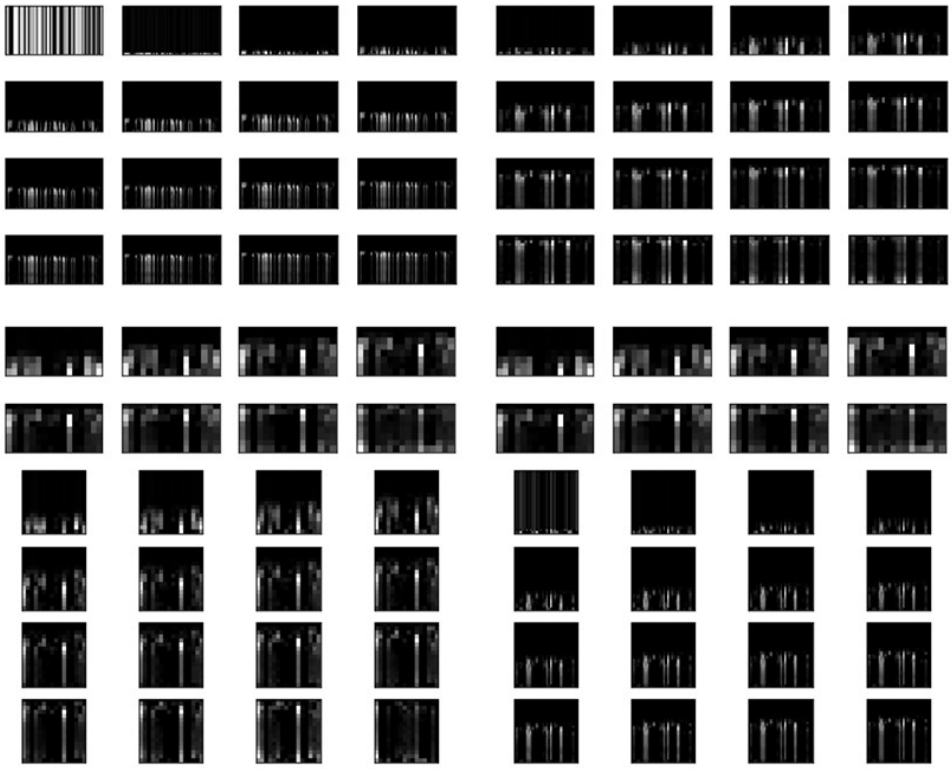


Figure 58: Sample of spatial features extracted by the non-sequential CNN model

9.3 Evaluation

Like the evaluation strategy followed for the 1D CNN, similar metrics have been used to examine the efficacy of the newly developed 2D CNN model. Parameters like F1 Score, kappa statistics, ROC-AUC score, etc. have been employed. While evaluating each model, the runtime information has been recorded and their mean performance indicators along with respective standard deviations have also been tabulated. Graphical evaluation of each model's performance during the training and validation phase has also been done. The pictorial comparison makes it easier to understand the performances of various models under consideration.

9.4 Conclusion

In this chapter, the development process of a 2D CNN has been described. The methodology followed, experiments carried out, and evaluation techniques adopted has been discussed. The experiments have started with a simple CNN model and ended up with a complex non-sequential recurrent CNN model. All the results retrieved are compared to select the final model for tumor classification.

10. Results & Discussion

Results of different experiments carried out in different phases have been carefully recorded for further evaluation. First of all, the class distribution in the newly formed dataset has been observed. It has been found that y-axis values are increasing as x-axis values are increasing (Figure 59) for most of the data points. This depicts the positive correlations in the feature set. This also justifies the use of the newly developed dataset in the present study.

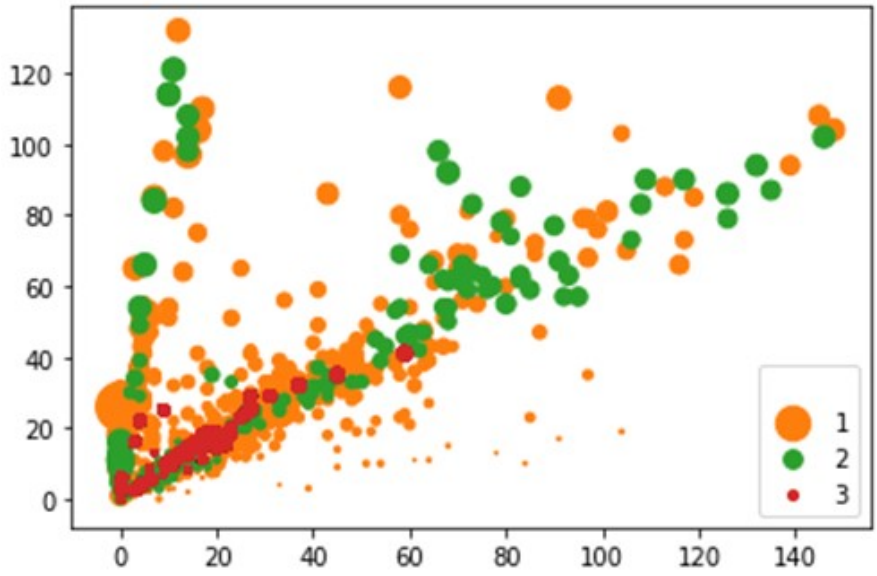


Figure 59: Class distribution observed in the dataset used in the study (1=> Malignant; 2=>Benign; 3=> others)

10.1 One Dimensional CNN

Figure 60 shows a sample of runtime structure of the newly developed model which is a combination of One Dimensional Convolutional Neural Network (1D CNN) and Recurrent Neural Network (RNN). Input parameters of a one-dimensional convolutional layer are (batch, steps, channels) and the output parameters are (batch, new_steps, filters). The first Conv1d layer has an input shape (None, 63, 1) and its output shape is (None, 61, 64). Here, *None* asserts that the batch length is not fixed; *63* is the present number of features (0… 63); *61* is the number of features to be used next, typically calculated as (no. of features – kernel size + filter bias) i.e. (63-3+1); *64* is the size of the filter/kernel used in the layer. The no. of parameters in Conv1d_37 is determined as (no. of

filters*filter size) +layer bias = (64*3+64) =256. As it is the initial layer, it does not have any output from previous layers.

Layer (type)	Output Shape	Param #	Calculation
conv1d_37 (Conv1D)	(None, 61, 64)	256	64*3+64=256
max_pooling1d_37	MaxPooling (None, 61, 64)	0	
conv1d_38 (Conv1D)	(None, 59, 32)	6176	64*3*32+32=6176
max_pooling1d_38	MaxPooling (None, 59, 32)	0	
conv1d_39 (Conv1D)	(None, 57, 16)	1552	32*3*16+16=1552
max_pooling1d_39	MaxPooling (None, 57, 16)	0	
conv1d_40 (Conv1D)	(None, 55, 8)	392	16*3*8+8=392
max_pooling1d_40	MaxPooling (None, 55, 8)	0	
gru_10 (GRU)	(None, 55, 7)	336	21*7+21*8+21=336
flatten_10 (Flatten)	(None, 385)	0	
dense_19 (Dense)	(None, 64)	24704	385*64+64=24704
dense_20 (Dense)	(None, 4)	260	64*4+4=260

Total params: 33,676
Trainable params: 33,676
Non-trainable params: 0

Figure 60: Runtime structure of the model comprising 1D CNN and RNN for AJCC Staging

For other convolutional layers, numbers of parameters are measured as (no. of filters*filter size*output filter from previous layers) +layer bias. For example, number of parameters in conv1d_38 = (64*3*32) +32=6176 and so on. Maxpooling or flatten layers don't have any trainable parameters. For gru_10 layer, number of parameters are calculated as (no. of kernel+ no. of recurrent kernel+ bias) = (21*7+21*8+21) = 336. For dense_20 layer, the no. of parameters= (input from dense_19*no. of filters) +bias= (64*4) +4=256+4=260.

10.1.1 Outcome of Experiment

Algorithm	Accuracy	F-Measure	ROC Area	RMSE	Kappa	Average Cost
CNN	0.91±0.04	0.90±0.02	0.96±0.05	0.25±0.06	0.87±0.07	0.24±0.21
CNN-RNN	0.97±0.02	0.95±0.01	1±0.01	0.1±0.01	0.92±0.04	0.09±0.1
FRNN	0.87±0.03	0.85±0.03	0.9±0.08	0.34±0.05	0.8±0.11	0.34±0.39
KNN	0.83±0.03	0.80±0.07	0.89±0.06	0.29±0.08	0.75±0.09	0.1±0.34
LR	0.72±0.03	0.71±0.08	0.79±0.07	0.19±0.08	0.65±0.11	0.97±0.39
MLP	0.76±0.03	0.74±0.08	0.83±0.05	0.38±0.07	0.69±0.11	1.4±0.41
RF	0.85±0.03	0.84±0.07	0.96±0.06	0.27±0.07	0.76±0.1	0.37±0.32
SVM	0.72±0.06	0.73±0.08	0.8±0.07	0.18±0.08	0.64±0.12	0.95±0.4

Table3: evaluation of grading by various algorithms concerning different metrics (CNN=>Convolutional Neural Network, CNN-RNN=> Convolutional Neural Network combined with Recurrent Neural Network, FRNN=>Fuzzy Rough Nearest Neighbours, KNN=>K-Nearest Neighbours, LR=>Logistic Regression, MLP=>Multi-Layer Perceptron, RF=> Random Forest, SVM=> Support Vector Machines)

Algorithm	Accuracy	F-Measure	ROC Area	RMSE	Kappa	Average Cost
CNN	0.85±0.02	0.85±0.03	0.91±0.06	0.27±0.04	0.78±0.05	0.8±0.3
CNN-RNN	0.96±0.01	0.96±0.01	0.99±0.01	0.07±0.02	0.93±0.03	0.11±0.2
FRNN	0.72±0.08	0.74±0.08	0.81±0.05	0.19±0.07	0.67±0.1	0.98±0.38
KNN	0.68±0.08	0.66±0.09	0.78±0.07	0.38±0.08	0.59±0.12	1.15±0.39
LR	0.6±0.05	0.58±0.12	0.6±0.08	0.24±0.09	0.39±0.19	1.23±0.5
MLP	0.67±0.09	0.65±0.09	0.78±0.05	0.32±0.07	0.59±0.13	1.51±0.61
RF	0.67±0.06	0.65±0.07	0.78±0.04	0.41±0.06	0.61±0.09	1.12±0.09
SVM	0.5±0.09	0.51±0.11	0.58±0.12	0.88±0.1	0.44±0.14	1.54±0.55

Table4: evaluation of T-Staging by various algorithms concerning different metrics (CNN=>Convolutional Neural Network, CNN-RNN=> Convolutional Neural Network combined with Recurrent Neural Network, FRNN=>Fuzzy Rough Nearest Neighbours, KNN=>K-Nearest Neighbours, LR=>Logistic Regression, MLP=>Multi-Layer Perceptron, RF=> Random Forest, SVM=> Support Vector Machines)

Algorithm	Accuracy	F-Measure	ROC Area	RMSE	Kappa	Average Cost
CNN	0.93±0.02	0.95±0.03	0.97±0.05	0.21±0.05	0.89±0.06	0.39±0.2
CNN-RNN	0.98±0.01	0.98±0.01	1±0.0	0.09±0.02	0.94±0.07	0.09±0.1
FRNN	0.92±0.05	0.91±0.04	0.99±0.03	0.2±0.01	0.87±0.05	0.45±0.4
KNN	0.82±0.06	0.78±0.07	0.88±0.05	0.3±0.02	0.7±0.11	0.41±0.4
LR	0.9±0.06	0.87±0.04	0.98±0.02	0.09±0.01	0.82±0.07	0.59±0.38
MLP	0.82±0.05	0.80±0.03	0.87±0.06	0.28±0.06	0.75±0.12	0.74±0.2
RF	0.83±0.07	0.84±0.05	0.9±0.03	0.21±0.04	0.75±0.09	0.38±0.1
SVM	0.8±0.09	0.78±0.06	0.87±0.04	0.18±0.02	0.71±0.07	0.7±0.9

Table5: evaluation of N-Staging by various algorithms concerning different metrics (CNN=>Convolutional Neural Network, CNN-RNN=> Convolutional Neural Network combined with Recurrent Neural Network, FRNN=>Fuzzy Rough Nearest Neighbours, KNN=>K-Nearest Neighbours, LR=>Logistic Regression, MLP=>Multi-Layer Perceptron, RF=> Random Forest, SVM=> Support Vector Machines)

Algorithm	Accuracy	F-Measure	ROC Area	RMSE	Kappa	Average Cost
CNN	0.95±0.02	0.92±0.04	0.98±0.03	0.23±0.06	0.9±0.04	0.29±0.1
CNN-RNN	0.99±0.01	0.98±0.02	1±0.01	0.09±0.02	0.94±0.01	0.08±0.09
FRNN	0.93±0.03	0.91±0.05	0.98±0.02	0.19±0.3	0.88±0.02	0.43±0.09
KNN	0.88±0.02	0.87±0.06	0.92±0.06	0.27±0.01	0.84±0.07	0.9±0.7
LR	0.92±0.04	0.86±0.06	0.98±0.05	0.08±0.7	0.85±0.07	0.41±0.8
MLP	0.88±0.02	0.89±0.08	0.92±0.05	0.24±0.01	0.84±0.06	0.51±0.9
RF	0.87±0.02	0.86±0.07	0.95±0.04	0.23±0.04	0.78±0.05	0.35±0.09
SVM	0.88±0.02	0.85±0.07	0.96±0.03	0.11±0.1	0.84±0.03	0.53±0.8

Table6: evaluation of M-Staging by various algorithms concerning different metrics (CNN=>Convolutional Neural Network, CNN-RNN=> Convolutional Neural Network combined with Recurrent Neural Network, FRNN=>Fuzzy Rough Nearest Neighbours, KNN=>K-Nearest Neighbours, LR=>Logistic Regression, MLP=>Multi-Layer Perceptron, RF=> Random Forest, SVM=> Support Vector Machines)

Algorithm	Accuracy	F-Measure	ROC Area	RMSE	Kappa	Average Cost
CNN	0.86±0.07	0.87±0.03	0.94±0.06	0.31±0.03	0.78±0.05	0.35±0.1
CNN-RNN	0.97±0.02	0.96±0.02	1±0.01	0.23±0.001	0.93±0.02	0.32±0.08
FRNN	0.71±0.08	0.71±0.05	0.82±0.09	0.27±0.02	0.66±0.09	0.89±0.2
KNN	0.72±0.09	0.72±0.04	0.81±0.04	0.35±0.3	0.69±0.1	1.18±0.42

LR	0.59±0.2	0.61±0.04	0.67±0.08	0.23±0.45	0.49±0.19	1.33±0.38
MLP	0.66±0.1	0.64±0.08	0.75±0.07	0.34±0.07	0.58±0.08	1.52±0.39
RF	0.68±0.1	0.67±0.05	0.79±0.06	0.39±0.09	0.59±0.08	1.18±0.27
SVM	0.52±0.02	0.52±0.05	0.62±0.06	0.86±0.08	0.42±0.1	1.53±0.35

Table7: evaluation of AJCC Staging by various algorithms concerning different metrics (CNN=>Convolutional Neural Network, CNN-RNN=> Convolutional Neural Network combined with Recurrent Neural Network, FRNN=>Fuzzy Rough Nearest Neighbours, KNN=>K-Nearest Neighbours, LR=>Logistic Regression, MLP=>Multi-Layer Perceptron, RF=> Random Forest, SVM=> Support Vector Machines)

Table3 shows the performances of different machine learning algorithms, including the newly developed CNN and CNN+RNN models while classifying histopathological grades. Similarly, table4 shows the performances during T-staging, table5 shows the same for N-staging, table6 shows it for M-staging, and table7 records it for AJCC staging. All of these tables have tabulated mean values of accuracy, F-measure, ROC area, RMSE, Kappa, and average cost along with their respective standard deviations for each method. From the result, it has been observed that the CNN+RNN model has performed more satisfactorily than other methods. Not only the accuracy of the CNN-RNN model is higher, but also the F-score is consistently on the higher side. This shows higher true positive values and lesser false positive and false negative values, i.e., lesser type-I and type-II errors respectively. This does not only show the relevance in the positive case detection but also shows the high rate of correctly identified negative cases. The High ROC area shows a greater true positive rate than the false positive rate. This implies that most of the positive cases have been identified correctly and the High Kappa value affirms the accuracy by showing perfect agreement between the true value and the predicted value. Low RMSE shows data concentration is intense along the line of best fit. The average cost is also lower for the CNN+RNN model than others. Thus, the performance of the CNN+RNN model concerning all the evaluation metrics is better than other machine learning methods. All these observations cast a decision in favor of the newly developed CNN+RNN model as far as the classification of malignant tumors is concerned.

```
acc: 1.0000 - val_loss: 0.0261 - val_acc: 0.9894
54/54 [==============================] - 0s 3ms/step
[0 0 3 1 0 2 0 1 0 0 6 1 0 2 1 1 3 1 1 3 0 3 1 2 0 2 3 3 1 6 3 0 2 1 4 4
0
 1 0 0 3 3 3 0 0 0 1 3 2 6 0 0 0 2]
['IA' 'IA' 'IIB' 'IB' 'IA' 'IIA' 'IA' 'IB' 'IA' 'IA' 'O' 'IB' 'IA' 'IIA'
 'IB' 'IB' 'IIB' 'IB' 'IB' 'IIB' 'IA' 'IIB' 'IB' 'IIA' 'IA' 'IIA' 'IIB'
 'IIB' 'IB' 'O' 'IIB' 'IA' 'IIA' 'IB' 'IIIA' 'IIIA' 'IA' 'IB' 'IA' 'IA'
 'IIB' 'IIB' 'IIB' 'IA' 'IA' 'IA' 'IB' 'IIB' 'IIA' 'O' 'IA' 'IA' 'IA'
 'IIA']
```

Figure 61: Inverse transformation of the AJCC stage prediction by the 1D CNN encoder model

From figure 61, the accuracy of the decoded prediction of AJCC staging done by the 1D CNN encoder model may be observed. The validation accuracy is around 98% which is quite encouraging and speaks for the efficacy of the model itself.

10.1.2 Analysis of Result

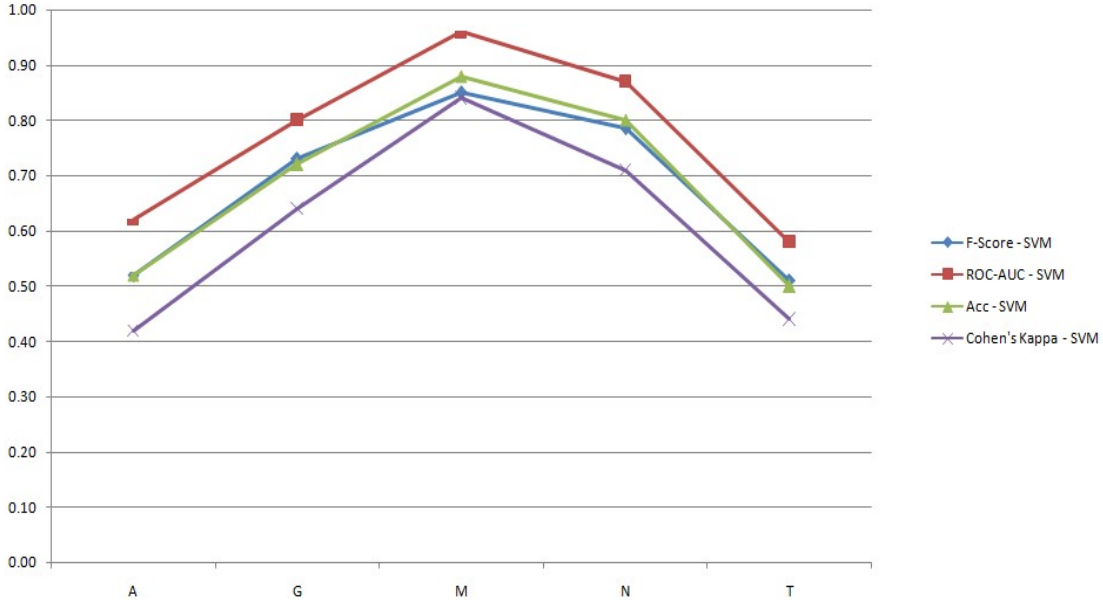


Figure 62: Performance of Support Vector Machines (SVM) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Charasteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

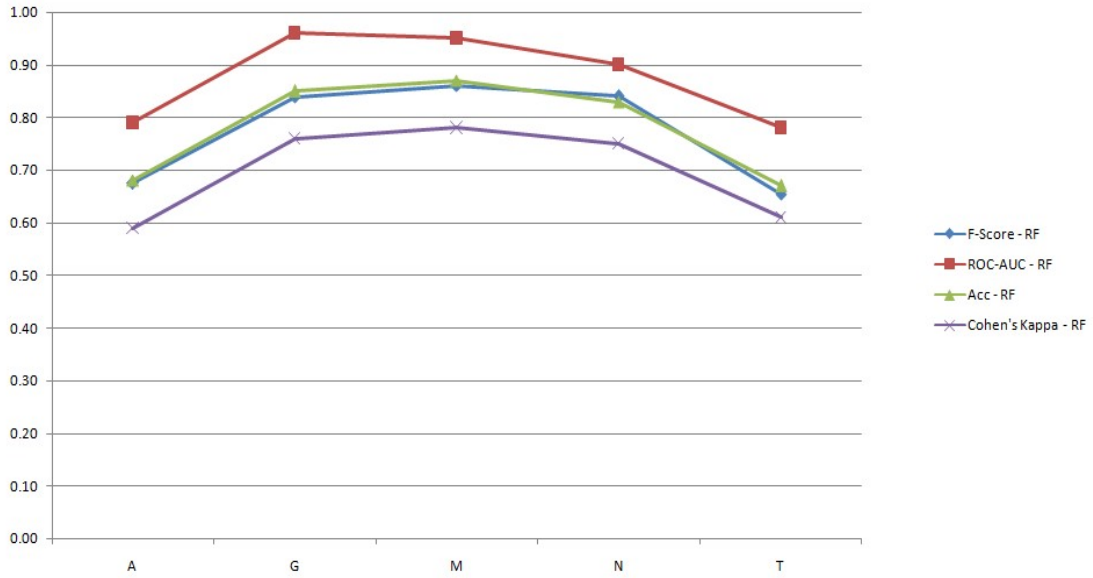


Figure 63: Performance of Random Forest (RF) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

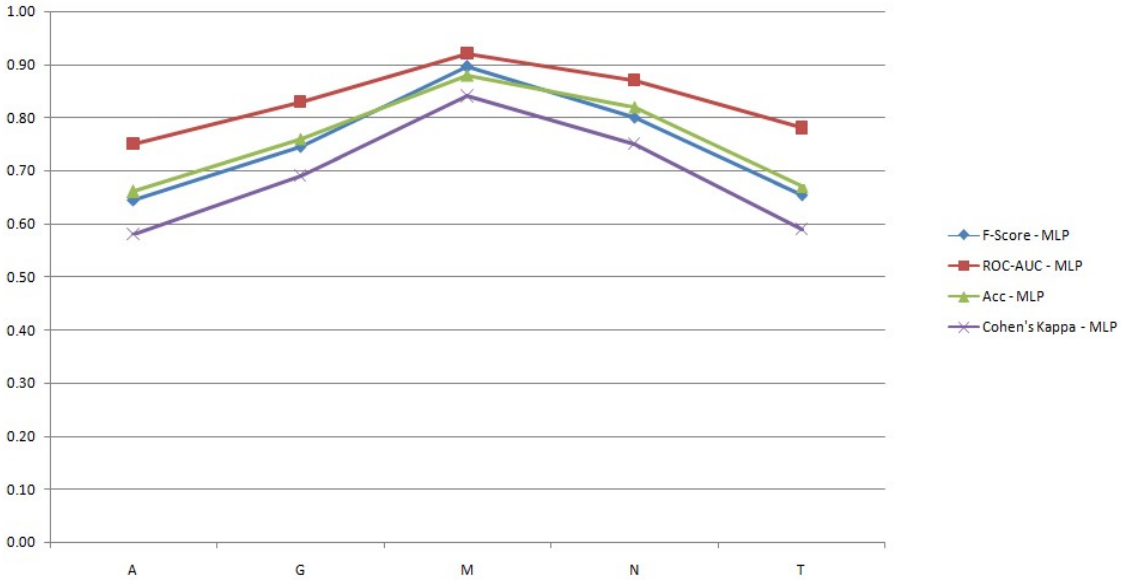


Figure 64: Performance of Multi-Layer Perceptron (MLP) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

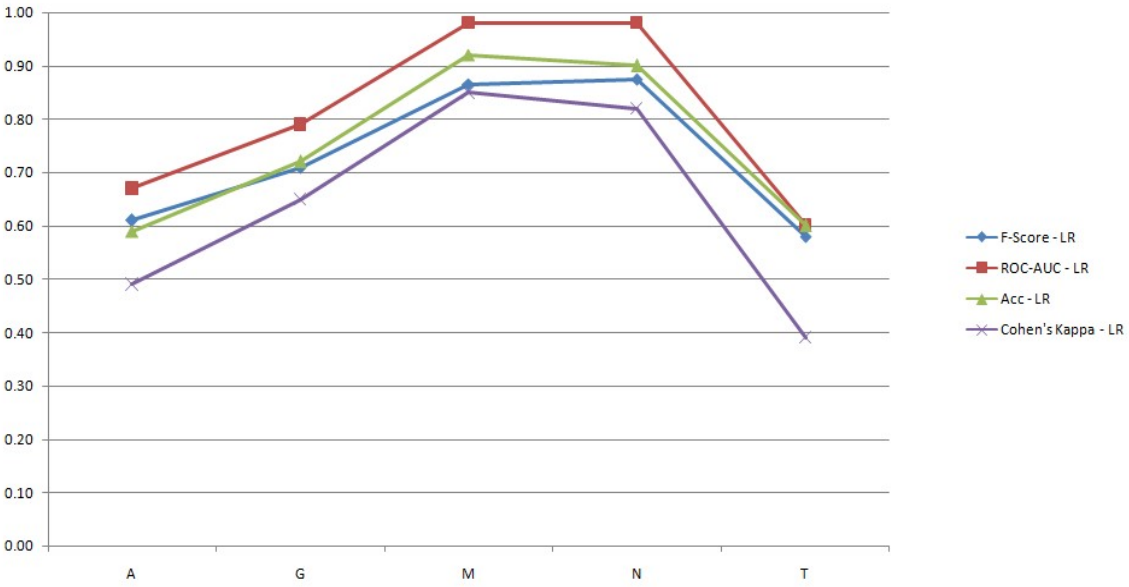


Figure 65: Performance of Logistic Regression (LR) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

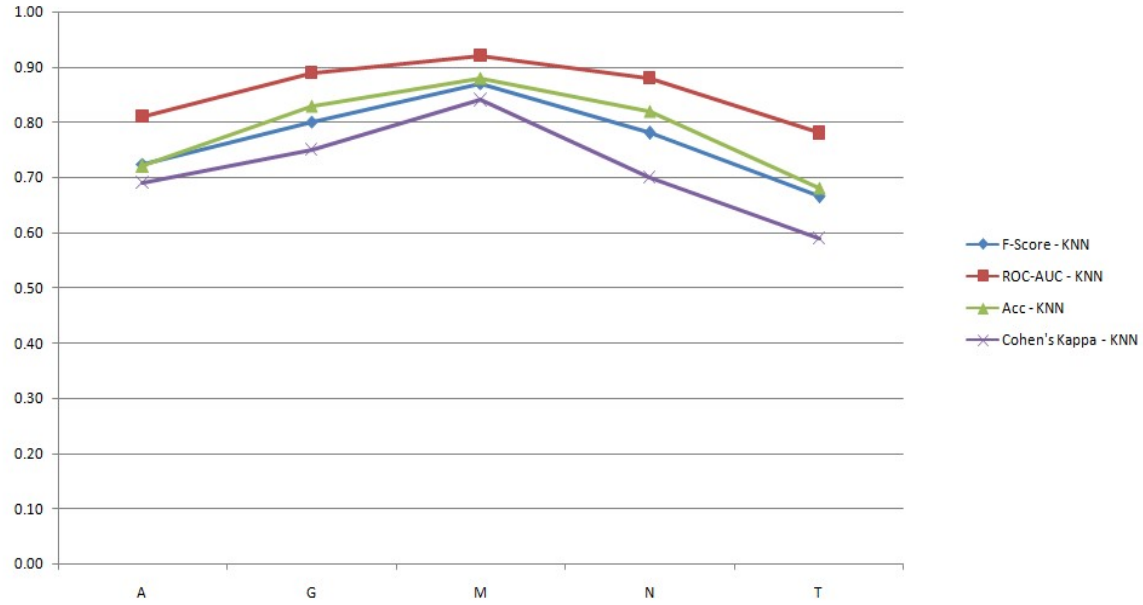


Figure 66: Performance of K-Nearest Neighbours (KNN) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

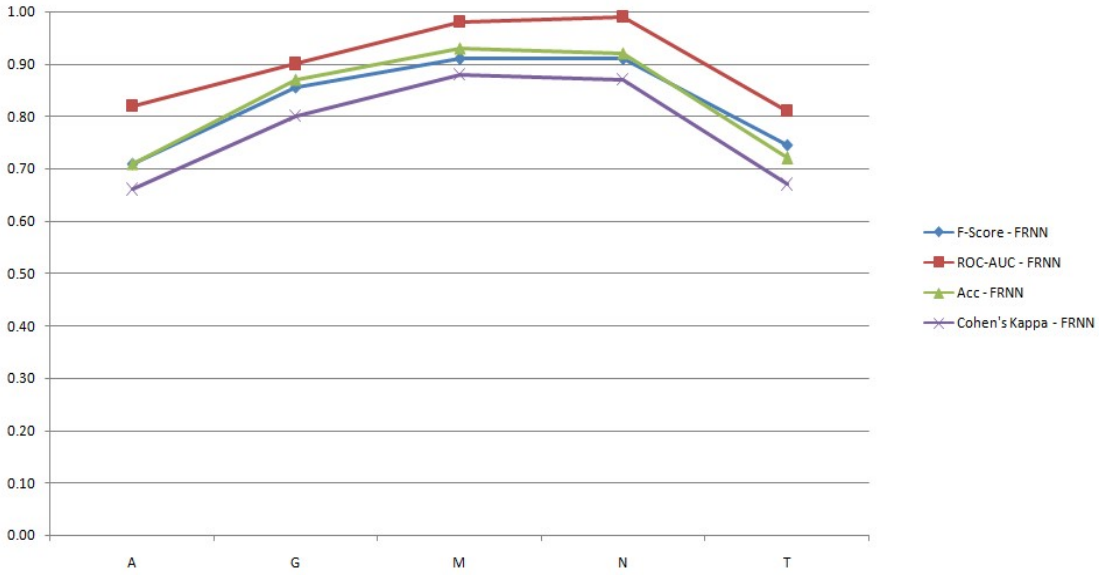


Figure 67: Performance of Fuzzy Rough Nearest Neighbours (FRNN) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Charasteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

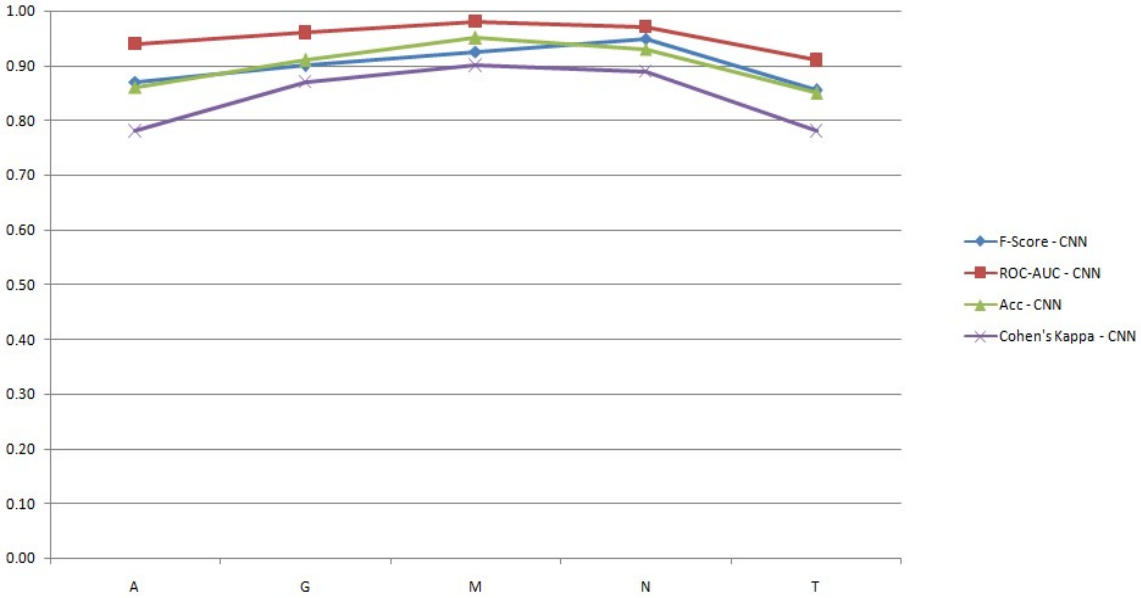


Figure 68: Performance of Convolutional Neural Network (CNN) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

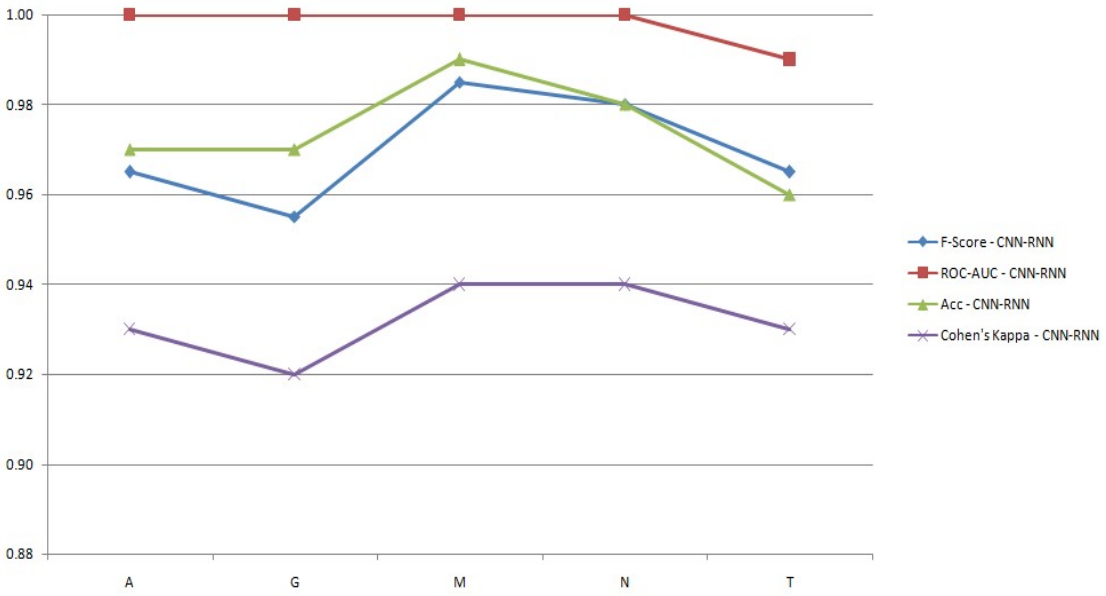
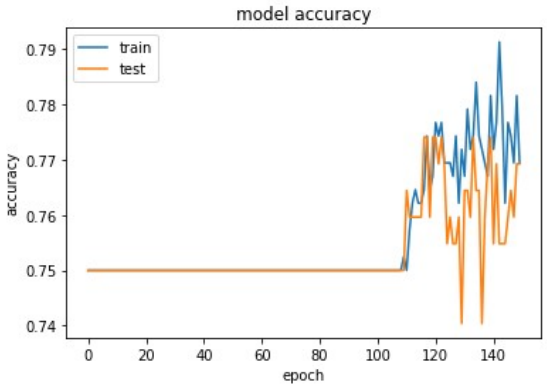


Figure 69: Performance of Convolutional Neural Network combined with-Recurrent Neural Network (CNN-RNN) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

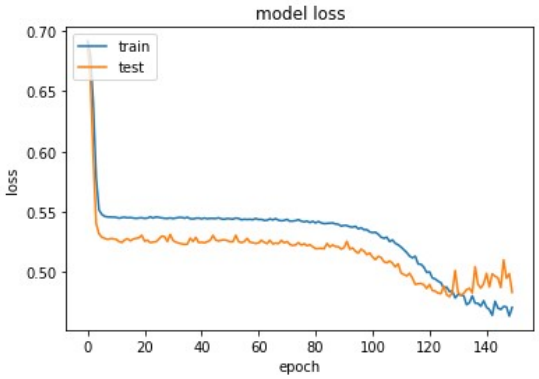
Figures 62, 63, 64, 65, 66, 67, 68, and 69 show the graphical comparison of performances of each machine learning method used in the study while conducting different staging and grading operations. The analysis reveals that all the machine learning methods have performed well when the number of class variables is less in number. Thus, in the case of N-staging or M-staging, performances of different algorithms are satisfactory. In such cases, the classification problems are less complicated with less number of target variables almost like a binary classification task. But the situation changes when target variables are more in numbers. In such cases, the algorithms face true multivariable-multiclass problems with real numbers having higher precisions as data values which have very tiny differences among each other. Orthodox machine learning methods find it very difficult to perform steadily in such cases. As a result, the performance curve for these algorithms concerning different parameters falls while classifying in terms of grade, AJCC stage, or T-stage. The newly developed CNN+RNN and CNN models perform commendably even in such adverse situations. Their performance curves always stay higher than the other methods employed in the study. These observations depict the superiority of convolutional models over other traditional models. CNN models even

performed better than the hybrid FRNN or MLP models. CNN+RNN model has a higher consistent curve than the simple CNN model. Although the CNN model has deeper convolutional layers, the CNN+RNN model has an edge over the former one. It has got recurrent layers which can memorize the past results. Thus while down-sampling features; the earlier important ones do not get faded away. This characteristic gives the later one an extra boost during classification.

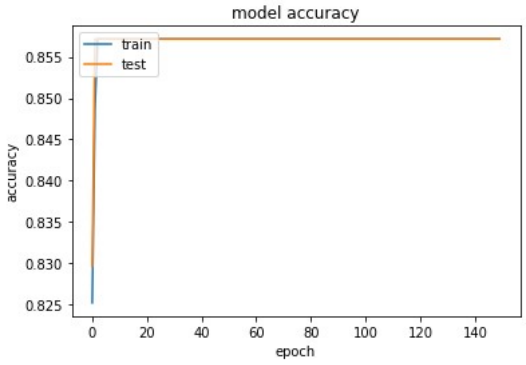
Although the CNN+RNN model has performed better than the CNN model, there are indeed some close calls observed. Before selecting the best possible model, further analysis of the performance of these two models has been done by using the validation accuracy curve and loss curve.



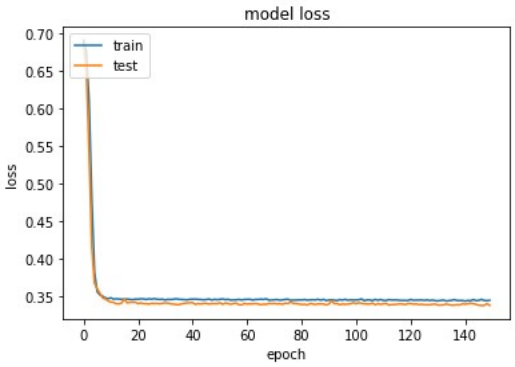
(a) grading accuracy



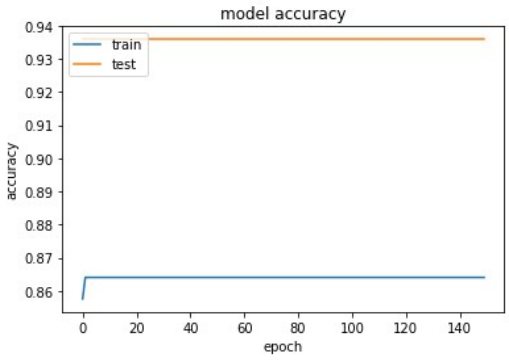
(b) grading loss



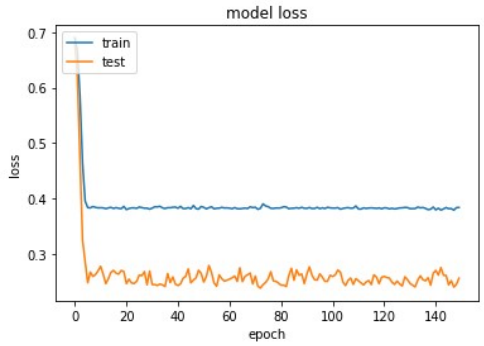
(c) T-staging accuracy



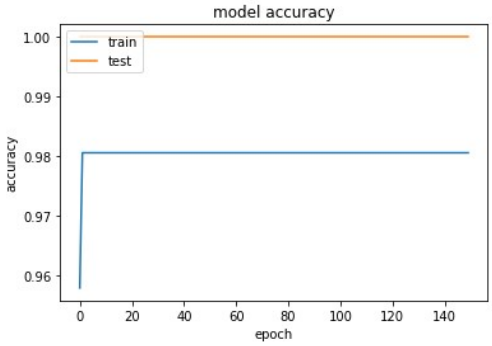
(d) T-staging loss



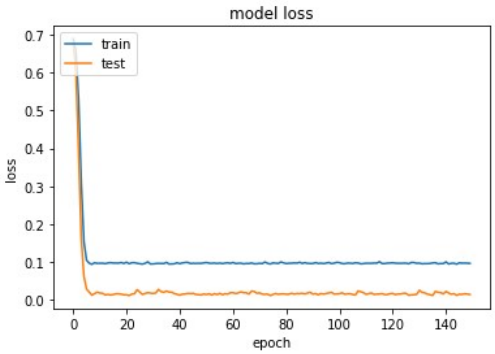
(e) N-staging accuracy



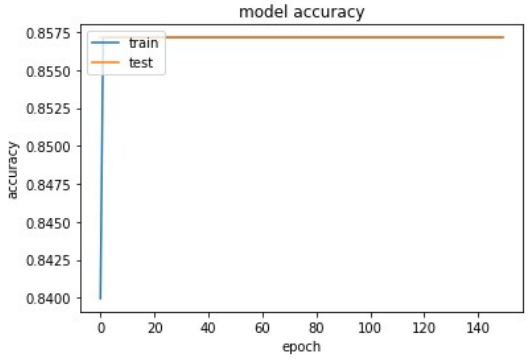
(f) N-staging Loss



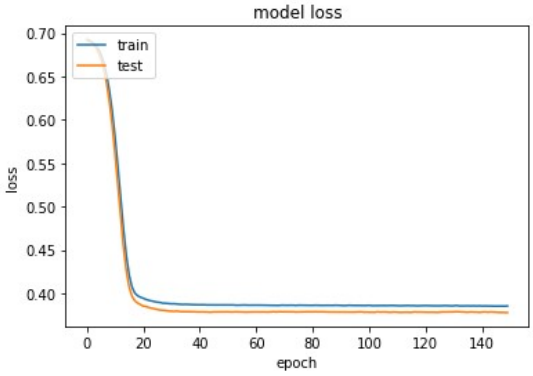
(g) M-staging accuracy



(h) M-staging loss

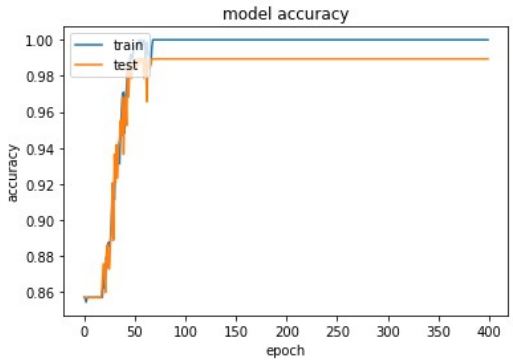


(i) AJCC staging accuracy

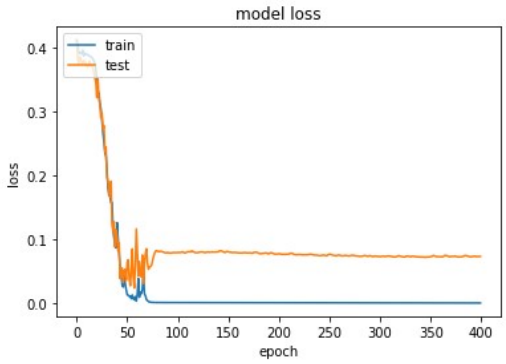


(j) AJCC staging loss

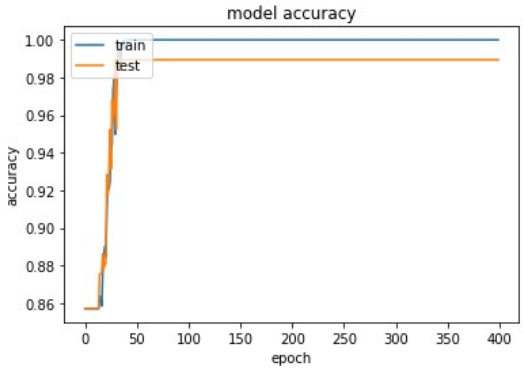
Figure 70: Train and test curves for CNN model accuracy and CNN model loss



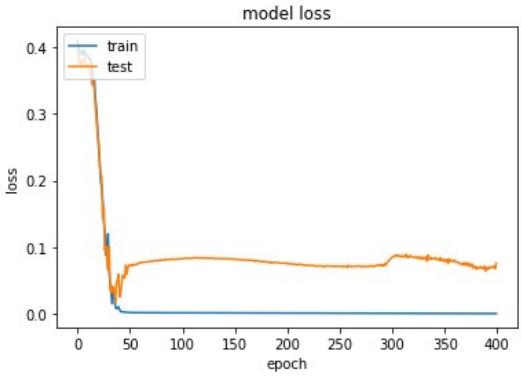
(a) grading accuracy



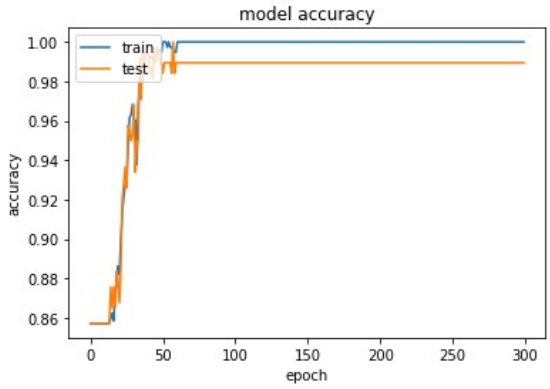
(b) grading loss



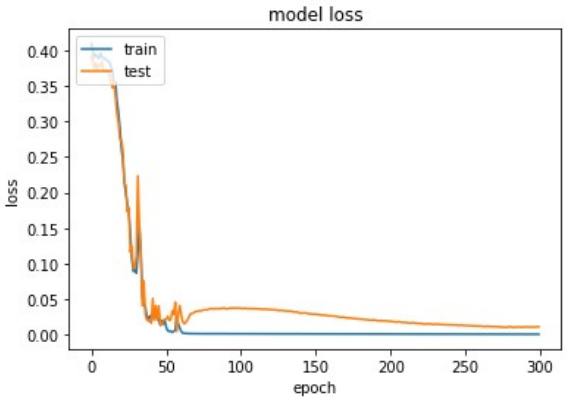
(c) T-staging accuracy



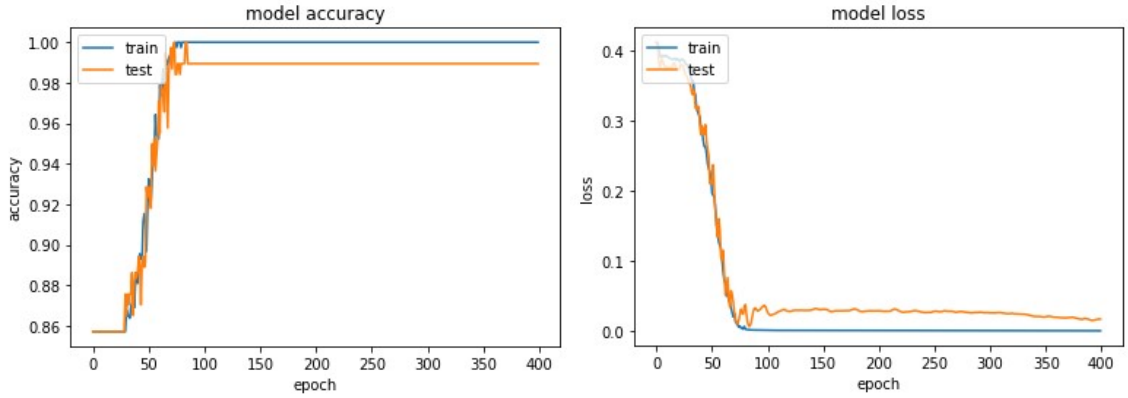
(d) T-staging loss



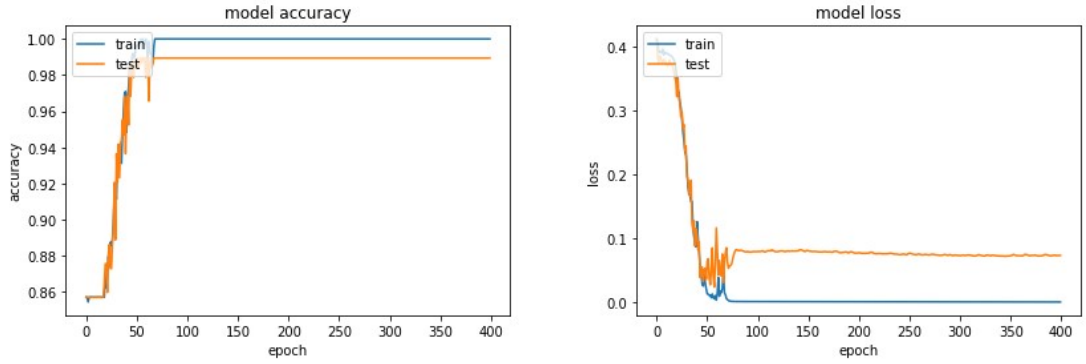
(e) N-staging accuracy



(f) N-staging loss



(g) M-staging accuracy (h) M-staging loss



(i) AJCC staging accuracy (j) AJCC staging loss

Figure 71: Train and test curves for CNN+RNN model accuracy and loss

After comparing figures 70 and 71, it is quite clear that the CNN model suffers from the problem of overfitting. There are frequent intersections between training and validation curves and fluctuations are also observed at later stages of iterations. In some cases, the test accuracy has been more than the training accuracy. This happens as many features become dormant due to down-sampling, dropout, and regularization operations. Thus, training accuracy becomes lower and loss becomes higher, however, during the test phase, all the features remain active and result in higher accuracy and lower loss. In this way, the CNN model becomes less trustworthy for validating new cases. On the other hand, the CNN+RNN model shows fewer intersections between the train and test curves. Initial fluctuations are stabilized by regularizes which is evident from figure 71. The training accuracy is always higher than the test accuracy and the training loss is always lower than the test loss. This shows that the CNN+RNN model is quite stable and has not lost important features during the training phase due to the presence of recurrent layers.

Thus, this model is more trustworthy for validating new or unknown cases. These results of analysis undoubtedly give a verdict in favor of the CNN+RNN model.

10.2 Two Dimensional CNN

Two dimensional CNN model has many facets: it starts with a simple CNN model, and then gradually turned into a CNN combined with bidirectional RNN (CNN+BiRNN). After that, it becomes a sequential ensemble of CNN and bidirectional RNN. Later, a new non-sequential recurrent model has been introduced.

10.2.1 Outcome of Experiment

All the experiments have been carried out for 5000 epochs with *early stop* callback value along with a 10-Fold Repeated Stratified Cross-validation and *patience=200*. The repeated stratification ensures that the cases selected for validation are not correlated. Models have been employed mainly to detect AJCC staging as TNM staging can easily be retrieved from it. In the case of the CNN model, the best result has been found at epoch 400 of iteration 6. The best result of the single CNN+BiRNN model has been found at epoch 450 of iteration 3. The CNN+BiRNN model ensemble achieved the best result at epoch 492 of iteration 5. The non-sequential model ensemble has attained the best result at epoch 300 of iteration 2.

Model	Validation Accuracy	F1-Score	Cohen's Kappa	ROC AUC
Non-sequential CNN Model Ensemble	0.93	0.92	0.86	0.98
Sequential CNN+BiRNN Model Ensemble	0.84	0.84	0.75	0.9
Sequential CNN+BiRNN Model	0.80	0.79	0.70	0.87
Sequential CNN Model	0.65	0.67	0.59	0.76

Table8: Best results recorded for various models used in the study

From table8, it is quite clear that the non-sequential model has almost outperformed other models w.r.t. all the important evaluation parameters. The same may be observed from the comparison of confusion matrices produced by different models (figure72). The credit of correctly classifying the highest number of instances per class goes to the non-sequential model. These results depict that the non-sequential model is pretty consistent.

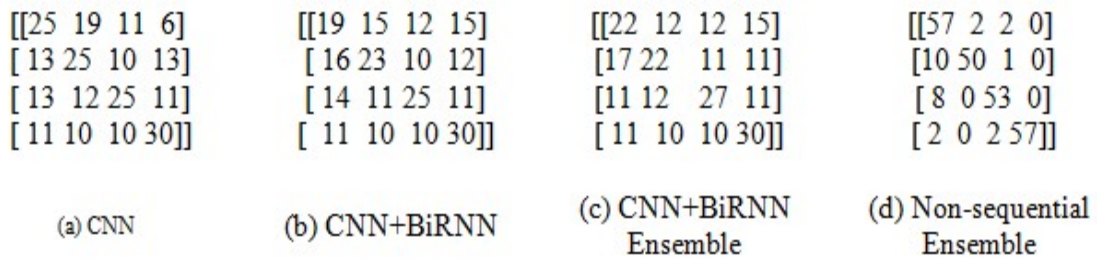


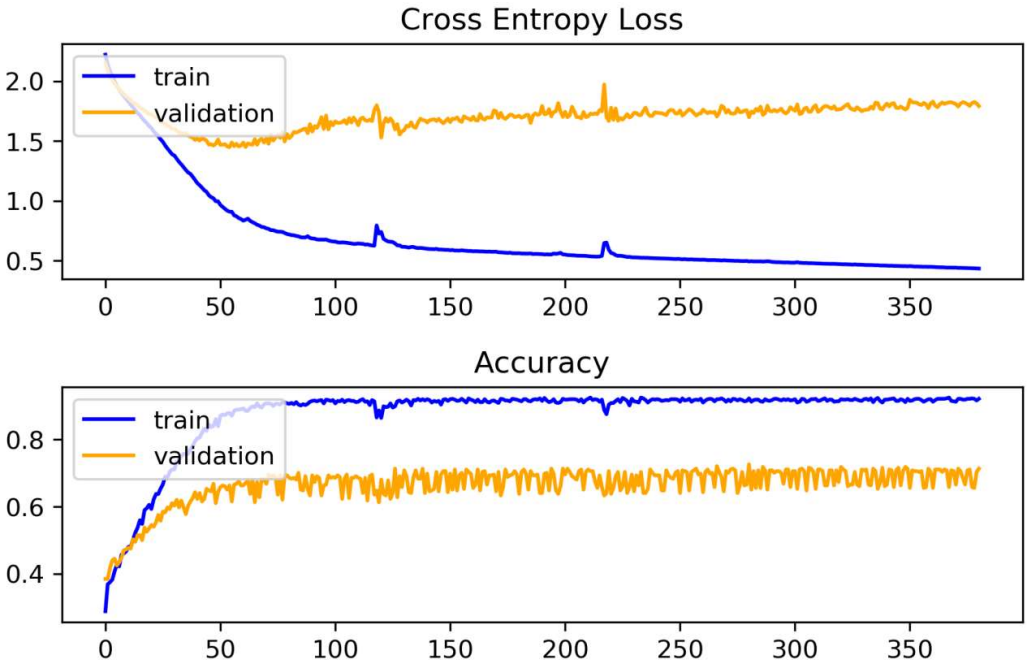
Figure 72: Sample confusion matrices produced by different models used in the study

Model	Validation Accuracy	F1-Score	Cohen's Kappa	ROC AUC
Non-sequential CNN Model Ensemble	0.91±0.001	0.9±0.004	0.84±0.005	0.97±0.002
Sequential CNN+BiRNN Model Ensemble	0.69±0.01	0.68±0.03	0.62±0.01	0.76±0.02
Sequential CNN+BiRNN Model	0.62±0.02	0.6±0.01	0.59±0.03	0.73±0.02
Sequential CNN Model	0.59±0.03	0.56±0.02	0.57±0.01	0.69±0.01

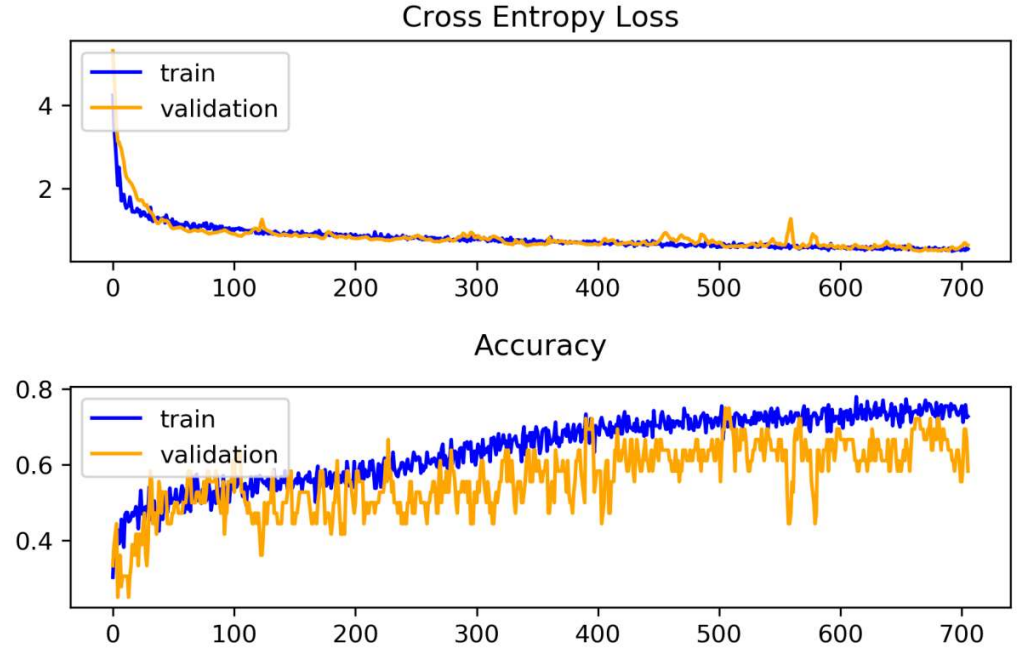
Table9: Average results (with standard deviations) of different models evaluated by various metrics

From the average results tabulated (Table 9), it may be observed that the validation accuracy of the non-sequential model is also on the higher side along with a high F1-score and ROC-AUC score. As accuracy may get inflated as a result of pre-processing bias, often precision and recall are considered to measure the false positives and false negatives, respectively. The problem is that both precision and recall come at each other's cost and can't have arbitrarily high value at the same time. Thus, if the weighted average of precision and recall, i.e., F1-score is high, it may be concluded that the performance of the model is free from type-I and type-II errors to a significant extent. Moreover, a high ROC-AUC score and kappa statistics confirm the high efficacy of the model by depicting the higher true positive rate over lower false-positive rate with fewer deviations.

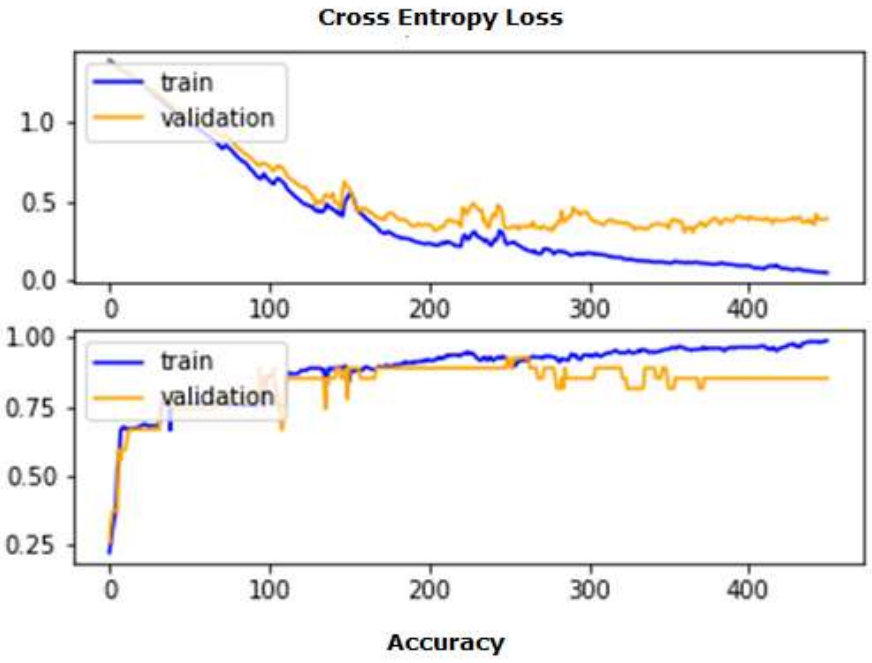
10.2.2 Analysis of Result



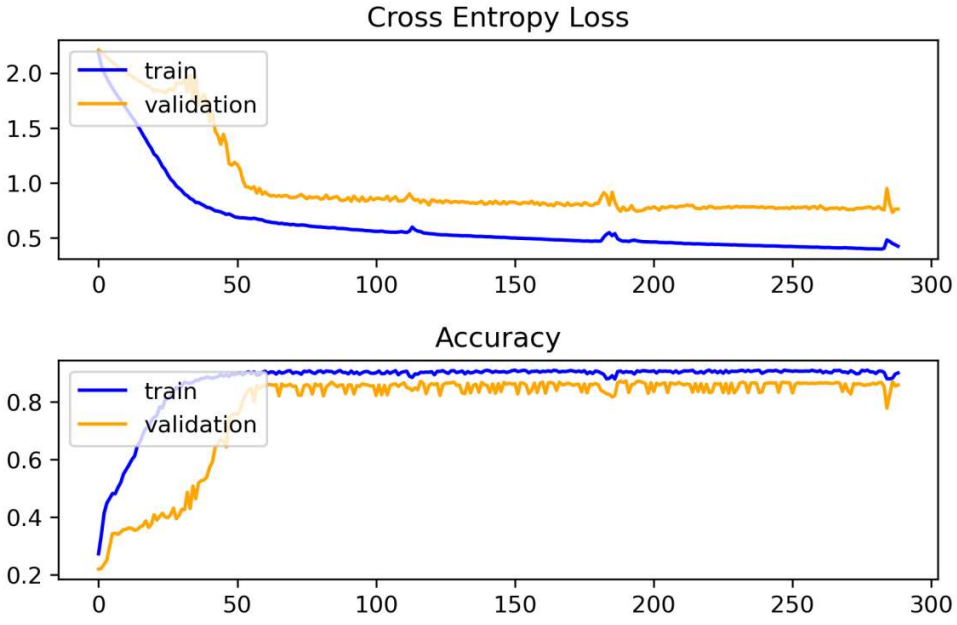
(a) Accuracy and Loss in the CNN model



(b)Accuracy and loss in CNN+BiRNN model



(c) Accuracy and loss in CNN+BiRNN model ensemble



(d) Accuracy and loss in the non-sequential model ensemble

Figure 73: Train and validation curves for AJCC staging by different two-dimensional CNN models

From figure 73 it may be observed that the validation accuracy of the newly developed model has been always higher than other models used in the study. On the other hand, the cross-entropy loss of the non-sequential model has been always lower than other models. Moreover, in the case of the CNN model, the validation loss tends towards higher value

and uneven fluctuations have also been witnessed. The CNN+BiRNN model or its ensemble has shown a lot of intersections between the train and validation accuracy. These observations show that these models are suffering from the problem of overfitting. The non-sequential model, on the contrary, shows less fluctuation and less intersection both in the case of accuracy and loss. Thus, the newly developed non-sequential model ensemble has a clear edge over other models under consideration.

The sheer accuracy of the non-sequential recurrent model is due to the inception and residual layers which act as efficient pre-processors. The point-wise convolution layer like (1X1) or (3X3) plays the role of cheap filters like the Sobel filter or edge detecting filter, etc. These layers also do not let the important features die out of down-sampling or dimensionality reduction process. Moreover, the bidirectional recurrent layers memorize the past and future incidents. Thus, the important features never get lost during the training or the validation phase. The non-sequential model has also used less number of timedistributed layers than other sequential models. It has also been observed that the memory usage during the execution of the non-sequential model was much less than those of other sequential models. All these observations have made the non-sequential recurrent model a lightweight yet efficient classifier. In the existing literature, it can hardly be observed that a deep learning model has been able to classify such a varied mix of tumor images with high accuracy. Every other model did it within a limited periphery of tumor genre, scanner modality, degree of classification, etc. Thus, none of the existing models could be a match for the newly developed model. The non-sequential model used in the study may further be explored by bringing in several other types of tumors, embedding different meta-learners, experimenting with hyper-parameters, etc.

10.3 Conclusion

In this chapter, the efficiency of both the 1D CNN models and 2D CNN models have been evaluated based on their performances as recorded during various experiments conducted so far. In the case of 1D CNN, the CNN+RNN model, and in the case of 2D CNN, the non-sequential recurrent model ensemble has emerged as the most powerful model in their respective domain. When the hardware resources are scarce, the manual feature extraction and implementation of the 1D CNN model is obvious. Otherwise, the 2D CNN model may be employed to classify tumors.

Bibliography

[1] World Health Organization, South-East Asia Regional Office. [Online] 11 9, 2017. [Cited: 08 18, 2019.] http://www.searo.who.int/india/topics/cancer/en/.

[2] Artificial Intelligence Techniques for Cancer Detection and Classification: Review Study. Alaá Rateb Mahmoud Al-shamasneh, Unaizah Hanum Binti Obaidellah. 3, 2017, European Scientific Journal, Vol. 13. 1857- 7431.

[3] Cooper GM. The Cell: A Molecular Approach. 2nd edition. Sunderland (MA): Sinauer Associates; 2000. The Development and Causes of Cancer. Available from: https://www.ncbi.nlm.nih.gov/books/NBK9963/.

[4] Cancer Detection at Early Stage Using PET/CT Imaging Technique. LungP.Nivetha, Mr.R.Manickavasagam. 3, s.l. : International Journal of Innovative Research in Computer and Communication Engineering, 2014, Vol. 2. 2320-9798.

[5] Second ESMO consensus Conference on lung cancer: pathology and molecular biomarkers for non-small-cell lung cancer. K. M. Kerr, L. Bubendorf, M. J. Edelman, A. Marchetti, T. Mok, S. Novello, K. O'Byrne, R. Stahel, S. Peters, E. Felip, Panel Members, Rolf Stahel, Enriqueta Felip, Solange Peters, Keith Kerr, Benjamin Besse, Johan Vansteenkiste, Wilfried Eberhardt, Martin. 9, s.l. : Annals of Oncology, 2014, Vol. 25.

[6] David N. Louis, Arie Perry, Guido Reifenberger, Andreas von Deimling, Dominique Figarella-Branger, Webster K. Cavenee. Hiroko Ohgaki, Otmar D. Wiestler, Paul Kleihues, David W. Ellison (2007) WHO classification of tumours of the central nervous system. In: World Health Organization classification of tumours, 4th edn. International Agency for Research on Cancer, Lyon, DOI 10.1007/s00401-016-1545-1.

[7] Brierley, J.D., Gospodarowicz, M.K. and Wittekind, Ch. TNM classification of malignant tumours (8th ed.). Chichester, West Sussex, UK : Wiley-Blackwell, 2017. 978-1-4443-3241-4.

[8] Prediction of Prognosis for Prostatic Adenocarcinoma by Combined Histological Grading and Clinical Staging. Gleason, Donald F, and Mellinger, George T. 2S, 2017, The Journal of Urology, Vol. 197, pp. S134-S139.

[9] TUMOR BRAIN DETECTION THROUGH MR IMAGES: A REVIEW OF LITERATURE. MOHAMMED SABBIH HAMOUD AL-TAMIMI, GHAZALI SULONG. 2, 2014, Journal of Theoretical and Applied Information Technology, Vol. 62. 1992-8645.

[10] Clinical Applications of PET in Brain Tumors. Chen, Wei. 9, 2007, THE JOURNAL OF NUCLEAR MEDICINE, Vol. 48.

[11] Role of 18F-FDG PET/CT in pre and post treatment. Bundhit Tantiwongkosi, Fang Yu, Anand Kanard, Frank R Miller. 5, 2014, World Journal of Radiology, Vol. 6, pp. 177-191.

[12] The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository. Clark, K, et al. 6, 2013, Journal of Digital Imaging, Vol. 26, pp. 1045-1057.

[13] CHOLLET, FRANÇOIS. Deep Learning with Python. NY : Manning Publications Co., 2018. 9781617294433.

[14] Multiple sclerosis identification by 14-layer convolutional neural network with batch normalization, dropout, and stochastic pooling. Wang, S. H., et al. 2018, Frontiers in Neuroscience, Vol. 12, pp. 1-11.

[15] Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. Yu-Dong Zhang, Zhengchao Dong, Xianqing Chen, Wenjuan Jia, Sidan Du, Khan Muhammad & Shui-Hua Wang. 3, s.l. : Multimedia Tools and Applications, 2019, Vol. 78, pp. 3613–3632. https://doi.org/10.1007/s11042-017-5243-3.

[16] Going deeper with convolutions. Szegedy Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Boston, MA : IEEE, 2015. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1-9.

[17] Application of Neural Networks to Data Mining. Kovacheva, Zlatinka Svetoslavova. 2, s.l. : Sultan Qaboos University, 2007, SQU Journal For Science, Vol. 12, pp. 121-141.

[18] Critical Dimension in Data Mining. Divya Suryakumar, Andrew H. Sung, Qingzhong Liu. s.l. : IARIA, 2012. eKNOW 2012 : The Fourth International Conference on Information, Process, and Knowledge Management. 978-1-61208-181-6.

[19] Application of Data Mining in Banking Sector. Bhambri, Vivek. 2, 2011, International Journal of Computer Science and Technology, Vol. 2. 0 9 7 6 - 8 4 9 1.

[20] A logical calculus of the ideas immanent in nervous activity. Pitts, Warren S. McCulloch & Walter. 1943, The bulletin of mathematical biophysics, Vol. 5, pp. 115-133.

[21] The perceptron: a probabilistic model for information storage and organization in the brain. Rosenblatt, Frank F. 6, 1958, Psychological Review, Vol. 65.

[22] A fast learning algorithm for deep belief nets. Hinton, G. E., Osindero, S. and Teh, Y. 2006, Neural Computation, Vol. 18, pp. 1527-1554.

[23] Aston Zhang, Zack C. Lipton, Mu Li, Alex J. Smola. Dive into Deep Learning. https://d2l.ai/. [Online] 04 18, 2019.

[24] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman. 1D Convolutional Neural Networks and Applications: A Survey. 2019.

[25] Behavior Investigation of Islamic Bank Deposit Return Utilizing Artificial Neural Networks Model. Saiful Anwara, Rifki Ismalb, Kenji Watanabec. 1, 2012, Journal of Islamic Finance, Vol. 1, pp. 44-58.

[26] Courville, Ian Goodfellow and Yoshua Bengio and Aaron. Deep Learning. www.deeplearningbook.org. [Online] 2019. [Cited: 4 15, 2020.]

[27] Classification using deep learning neural networks for brain tumors. Heba Mohsen, El-Sayed A. El-Dahshan, El-Sayed M. El-Horbaty, Abdel-Badeeh M. Salem. 2017, Future Computing and Informatics Journal. https://doi.org/10.1016/j.fcij.2017.12.001.

[28] Hepatitis disease diagnosis using backpropagation and the naive bayes classifiers. KARLIK, Bekir. 1, 2011, Journal of Science and Technology, Vol. 1.

[29] The Use of Artificial Neural Network (ANN) for Prediction of Some Airborne Pollutants Concentration in Urban Areas. LUCICA BARBES, CORNELIU NEAGU, LUCIA MELNIC, CONSTANTIN ILIE, MIRELA VELICU. 3, 2009, Revista de Chimie (Bucure^{o}ti), Vol. 60.

[30] Predicting Future Depositor`s Rate of Return Applying Neural Network: A Case-study of Indonesian Islamic Bank. Saiful Anwar, Kenji Watanabe,. 3, 2010, International Journal of Economics and Finance, Vol. 2.

[31] Designing an Artificial Neural Network Model for the Prediction of Thrombo-embolic Stroke. D.Shanthi, Dr.G.Sahoo & Dr.N.Saravanan. 1, 2009, International Journals of Biometric and Bioinformatics (IJBB), Vol. 3.

[32] Review on Methods to Fix Number of Hidden Neurons in Neural Networks. Deepa, K. Gnana Sheela and S. N. s.l. : Hindawi Publishing Corporation, 2013, Mathematical Problems in Engineering.

[33] Feature-Based Lung Nodule Classification. Farag, A.A., et al. Las Vegas, NV, USA : s.n., 2010. International Symposium on Visual Computing. Vol. 3, pp. 79–88.

[34] Location classification of lung nodules with optimized graph construction. Song, Y., et al. Barcelona, Spain : s.n., 2012. 9th IEEE International Symposium on Biomedical Imaging (ISBI). pp. 1439–1442.

[35] Lung Nodule Classification with Multilevel Patch-Based Context Analysis. Zhang, F., et al. 2014, IEEE Trans. Biomed. Eng., Vol. 61, pp. 1155–1166.

[36] Lung cancer classification using neural networks for CT images. Kuruvilla, J. and Gunavathi, K. 1, 2014, Computer Methods and Programs in Biomedicine, Vol. 113, pp. 202-209.

[37] Lung Nodule Classification Using Deep Features in CT Images. Kumar, D., Wong, A. and Clausi, D. A. Halifax, NS, Canada : IEEE, 2015. 12th Conference on Computer and Robot Vision. pp. 133-138.

[38] Using Deep Learning for Classification of Lung Nodules on Computed Tomography Images. Song Q.Z., Zhao L., Luo X.K., Dou X.C. s.l. : Hindawi, 2017, Computer Vision in Healthcare Applications. doi:10.1155/2017/8314740..

[39] Automated Classification of Lung Cancer Types from Cytological Images Using Deep Convolutional Neural Networks. Atsushi Teramoto, Tetsuya Tsukamoto, Yuka Kiriyama, Hiroshi Fujita. 2017, BioMed Research International Volume. https://doi.org/10.1155/2017/4067832.

[40] Using deep-learning radiomics to predict lung cancer histology. Tafadzwa Lawrence Chaunzwa, David C. Christiani, Michael Lanuti, Andrea Shafer, Nancy Diao, Raymond H. Mak, and Hugo Aerts. 15, 2018, Journal of Clinical Oncology, Vol. 36. 10.1200/JCO.2018.36.15_suppl.8545.

[41] CNN models discriminating between pulmonary micro-nodules and non-nodules from CT images. Patrice Monkam, Shouliang Qi, Mingjie Xu, Fangfang Han, Xinzhuo Zhao and Wei Qian. 96, 2018, BioMedical Engineering OnLine, Vol. 17. https://doi.org/10.1186/s12938-018-0529-x.

[42] Early detection of lung cancer from CT images: nodule segmentation and classification using deep learning. Manu Sharma, Jignesh S. Bhatt, and Manjunath V. Joshi. 2018. Tenth International Conference on Machine Vision (ICMV 2017). https://doi.org/10.1117/12.2309530.

[43] Feature Representation Using Deep Autoencoder for Lung Nodule Image Classification. Mao, K.M., et al. 2018, Complexity in Medical Informatics. https://doi.org/10.1155/2018/3078374.

[44] Multi-view multi-scale CNNs for lung nodule type classification from CT images. Liu, X.L., Hou, F. and Hao, A. 2018, Pattern Recognition, Vol. 77, pp. 262–275.

[45] A Generalized Deep Learning-Based Diagnostic System for Early Diagnosis of Various Types of Pulmonary Nodules. Shaffie A, Soliman A, Fraiwan L, Ghazal M, Taher F, Dunlap N, Wang B, van Berkel V, Keynton R, Elmaghraby A, El-Baz A. s.l. : SAGE, 2018, Technology in Cancer Research and Treatment.

[46] Lung Cancer Detection: A Deep Learning Approach. Bhatia, Siddharth, Sinha, Yash and Goel, Lavika. 2019, Soft Computing for Problem Solving, Vol. 817, pp. 699-705.

[47] Breast cancer risk estimation with artificial neural networks revisited: Discrimination and calibration. T. Ayer, O. Alagoz, J. Chhatwal, J.W. Shavlik, C.E. Kahn, E.S. 14, s.l. : Elsevier, 2010, Cancer, Vol. 116, pp. 3310-3321.

[48] Breast Fine Needle Tumor Classification using Neural. George, Y. M., Elbagoury, B. M., Zayed, H. H., & Roushdy, M. I. 5, 2012, International Journal of Computer Science Issues, Vol. 9.

[49] Classification of neural network structures for breast cancer diagnosis. S Swathi, S Rizwana, G Anjan Babu, P Santhosh Kumar, PVGK Sarma. 1, 2012, International Journal of Computer Science and Communication, Vol. 3, pp. 227-231.

[50] Using three machine learning techniques for predicting breast cancer recurrence. Eshlaghy, A.T., Poorebrahimi, A., Ebrahimi, M., Razavi, A.R., Ahmad, L.G. 2, s.l. : Elsevier, 2013, Journal of Health & Medical Informatics, Vol. 4, pp. 124-130.

[51] Probabilistic neural network for breast cancer classification. El-Said, Ahmad Taher Azar & Shaimaa Ahmed. 2013, Neural Computing and Applications, Vol. 23, pp. 1737–1751. https://doi.org/10.1007/s00521-012-1134-8.

[52] A novel neural network based automated system for diagnosis of breast cancer from real time biopsy slides. Seema Singh, J Harini, BR Surabhi. s.l. : IEEE, 2014. International Conference on Circuits, Communication, Control and Computing. pp. 50-53.

[53] Investigations on impact of feature normalization techniques on classifier's performance in breast tumor classification. Bikesh Kumar Singh, Kesari Verma, AS Thoke. 19, s.l. : Foundation of Computer Science, 2015, International Journal of Computer Applications, Vol. 116.

[54] A Fuzzy-rough Nearest Neighbor Classifier combined with consistency-based subset evaluation and instance selection for automated diagnosis of breast cancer. Onan, Aytuğ. 20, 2015, Expert Systems and Applications, Vol. 42. 10.1016/j.eswa.2015.05.006.

[55] Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. Huynh BQ, Li H, Giger ML. 3, 2016, Journal of Medical Imaging, Vol. 3. doi: 10.1117/1.JMI.3.3.034501.

[56] A deep learning approach for the analysis of masses in mammograms with minimal user intervention. Dhungel N, Carneiro G, Bradley A. s.l. : Elsevier, 2017, Medical Image Analysis, Vol. 37, pp. 114-128. doi: 10.1016/j.media.2017.01.009.

[57] Classifying Mammographic Breast Density by Residual Learning. Xu J, Li C, Zhou Y, Mou L, Zheng H, Wang S. s.l. : https://arxiv.org/abs/1809.10241, 2018.

[58] Prediction of mortality after radical cystectomy for bladder cancer by machine learning techniques. Wang G, Lam K-M, Deng Z. 2015, Computers in Biology and Medicine. doi: 10.1016/j.compbiomed.2015.05.015.

[59] Preoperative prediction of muscular invasiveness of bladder cancer with radiomic features on conventional MRI and its high-order derivative maps. Xu X, Liu Y, Zhang X, Tian Q, Wu Y, Zhang G, et al. 2017, Abdominal Urology. doi: 10.1007/s00261-017-1079-6.

[60] Objective evaluation for the cystoscopic diagnosis of bladder cancer using artificial intelligence. A. Ikeda, Y. Hoshino, H. Nosato, T. Kojima, K. Kawai, Y. Ohishi, H. Sakanashi, M. Murakawa, N. Yamanouchi, H. Nishiyama. 2, 2018, European Urology, Vol. 17. https://doi.org/10.1016/S1569-9056(18)31702-0.

[61] Diagnostic classification of cystoscopic images using deep convolutional neural networks. Okyaz Eminaga, Nurettin Eminaga, Axel Semjonow, and Bernhard Breil. 2018, JCO Clinical Cancer Informatics. https://doi.org/10.1200/CCI.17.00126.

[62] Diagnostic accuracy of CT for prediction of bladder cancer treatment response with and without computerized decision support. Cha KH, Hadjiiski LM, Cohan RH, Chan HP, Caoili EM, Davenport MS, et al. 2018, Academic Radiology. doi: 10.1016/j.acra.2018.10.010.

[63] Augmented Bladder Tumor Detection Using Deep Learning. Eugene Shkolyar, Xiao Jia, Timothy C. Chang, et al. 2019, European Urology.

[64] Development of a noninvasive tool to preoperatively evaluate the muscular invasiveness of bladder cancer using a radiomics approach. Zheng J, Kong J, Wu S, Li Y, Cai J, Yu H, et al. 2019, Cancer. doi: 10.1002/cncr.32490.

[65] A radiogenomics signature for predicting the clinical outcome of bladder urothelial carcinoma. Lin P, Wen DY, Chen L, Li X, Li SH, Yan HB, et al. 2019, European Radiology. doi: 10.1007/s00330-019-06371-w.

[66] A novel machine learning approach reveals latent vascular phenotypes predictive of renal cancer outcome. Nathan Ing, Fangjin Huang, Andrew Conley, Sungyong You, Zhaoxuan Ma, Sergey Klimov, Chisato Ohe, Xiaopu Yuan, Mahul B. Amin, Robert Figlin, Arkadiusz Gertych, and Beatrice S. Knudsen. s.l. : Nature, 2017, Scientific Reports. doi: 10.1038/s41598-017-13196-4.

[67] A Machine Learning Approach for the Classification of Kidney Cancer Subtypes Using miRNA Genome Data. Ali Muhamed Ali, Hanqi Zhuang,Ali Ibrahim, Oneeb Rehman, Michelle Huang and Andrew Wu. 2018, Applied Sciences. doi:10.3390/app8122422.

[68] Clear Cell Renal Cell Carcinoma: Machine Learning-Based Quantitative Computed Tomography Texture Analysis for Prediction of Fuhrman Nuclear Grade. Bektas, Ceyda & Kocak, Burak & Yardimci, Aytul Hande & Turkcanoglu, Mehmet & Yucetas, Ugur & Koca, Sevim & Erdim, Cagri & Kilickesmez, Ozgur. 2018, European Radiology. 10.1007/s00330-018-5698-2.

[69] The Classification of Renal Cancer in 3-Phase CT Images Using a Deep Learning Method. Seokmin Han, Sung Il Hwang & Hak Jong Lee. s.l. : Springer Nature, 2019, Journal of Digital Imaging, Vol. 32, pp. 638–643.

[70] A Deep Learning-Based Radiomics Model for Differentiating Benign and Malignant Renal Tumors. Leilei Zhou, Zuoheng Zhang, Yu-Chen Chen, Zhen-Yu Zhao, Xin-Dao Yin, and Hong-Bing Jiang. 2, s.l. : Elsevier, 2019, Translational Oncology, Vol. 12, pp. 292-300. doi: 10.1016/j.tranon.2018.10.012.

[71] Pan-Renal Cell Carcinoma classification and survival prediction from histopathology images using deep learning. Sairam Tabibu, P. K. Vinod & C. V. Jawahar. s.l. : Springer Nature, 2019, Scientific Reports, Vol. 9.

[72] Automated clear cell renal carcinoma grade classification with prognostic significance. Katherine Tian, Christopher A. Rubadue ,Douglas I. Lin,Mitko Veta,Michael E. Pyle,Humayun Irshad,Yujing J. Heng. 10, 2019, PLOS ONE , Vol. 14. https://doi.org/10.1371/journal.pone.0222641.

[73] Radiogenomics in Clear Cell Renal Cell Carcinoma: Machine Learning–Based High-Dimensional Quantitative CT Texture Analysis in Predicting PBRM1 Mutation Status. Burak Kocak, Emine Sebnem Durmaz, Ece Ates and Melis Baykara Ulusan. 2019, American Journal of Roentgenology. 10.2214/AJR.18.20443.

[74] Initial Evaluation of Computer-Assisted Radiologic Assessment for Renal Mass Edge Detection as an Indication of Tumor Roughness to Predict Renal Cancer Subtypes. Rahul Rajendran, Kevan Iffrig, Deepak K Pruthi, Allison Wheeler, Brian Neuman, Dharam Kaushik , Ahmed M Mansour, Karen Panetta , Sos Agaian, and Michael A. 2019, Advances in Urology. https://doi.org/10.1155/2019/3590623.

[75] Neural network evaluation of PET scans of the liver: a potentially useful adjunct in clinical interpretation. Preis O, Blake MA, Scott JA. 2011, Radiology, pp. 714–721.

[76] Effective liver cancer diagnosis method based on machine learning algorithm. Kim, Sangman & Jung, Seungpyo & Park, Youngju & Lee, Jihoon & Park, Jusung. 2015. 7th International Conference on BioMedical Engineering and Informatics. pp. 714-718. 10.1109/BMEI.2014.7002866.

[77] Automatic detection of new tumors and tumor burden evaluation in longitudinal liver CT scan studies. Vivanti R, Szeskin A, Lev-Cohain N, Sosna J, Joskowicz L. 11, 2017, Int Journal of Computer Assisted Radiology and Surgery, Vol. 12, pp. 1945-1957.

[78] Deep learning based liver cancer detection using watershed transform and Gaussian mixture model techniques. Sabut, Sukanta & Das, Amita & Acharya, U Rajendra & Panda, Soumya. 2018, Cognitive Systems Research. 10.1016/j.cogsys.2018.12.009.

[79] Fully convolutional network and sparsity-based dictionary learning for liver lesion detection in CT examinations. Ben-Cohen A, Klang E, Kerpel A, Konen E, Amitai M, Greenspan H. 2018, Neurocomputing, pp. 1585–1594.

[80] Preliminary Study of Chronic Liver Classification on Ultrasound Images Using an Ensemble Model. Bharti P, Mittal D, Ananthasivan R. s.l. : SAGE, 2018, Search Results, Vol. 40, pp. 357–379.

[81] GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification. Frid-Adar M, Diamant I, Klang E, Amitai M, Goldberger J, Greenspan H. 2018, Neurocomputing, pp. 321–331.

[82] Machine-learning Approach for the Development of a Novel Predictive Model for the Diagnosis of Hepatocellular Carcinoma. Masaya Sato, Kentaro Morimoto, Shigeki Kajihara, Ryosuke Tateishi, Shuichiro Shiina, Kazuhiko Koike & Yutaka Yatomi. s.l. : Nature, 2019, Scientific Reports, Vol. 9.

[83] End-To-End Discriminative Deep Network For Liver Lesion Classification. Romero, F.P., et al. Venice, Italy : s.n., 2019. IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019).

[84] Multiple-instance learning with global and local features for thyroid ultrasound image classification. J. Ding, H. D. Cheng, J. Huang and Y. Zhang. Dalian : s.n., 2014. 7th International Conference on Biomedical Engineering and Informatics. pp. 66-70.

[85] Classifier Model Based on Machine Learning Algorithms: Application to Differential Diagnosis of Suspicious Thyroid Nodules via Sonography. Hongxun Wu, Zhaohong Deng, Bingjie Zhang, Qianyun Liu and Junyong Chen. 4, 2016, American Journal of Roentgenology, Vol. 207, pp. 859-864.

[86] Machine Learning Techniques for Thyroid Disease Diagnosis - A Review. Razia, Shaik & Rao, M. 2016, Indian Journal of Science and Technology, Vol. 9. 10.17485/ijst/2016/v9i28/93705.

[87] Image Processing Techniques for Determining Cold Thyroid Nodules. A. Torab-Miandoab, P. Rezaei-hachesu, T. Samad and S. Habibi-Chenaran. Slemani : s.n., 2017. International Conference on Current Research in Computer Science and Information Technology (ICCIT). pp. 133-136.

[88] Reliability of the ultrasound classification system of thyroid nodules in predicting malignancy. Farihah AG, Nurismah MI, Husyairi H, Shahrun Niza AS, Radhika S. 1, 2018, Medical Journal Malaysia, Vol. 73, pp. 9-15.

[89] Diagnosis of Thyroid Nodules: Performance of a Deep Learning Convolutional Neural Network Model vs. Radiologists. Vivian Y. Park, Kyunghwa Han, Yeong Kyeong Seong, Moon Ho Park, Eun-Kyung Kim, Hee Jung Moon, Jung Hyun Yoon & Jin Young Kwak. s.l. : Springer Nature, 2019, Scientific Reports, Vol. 9.

[90] Using deep convolutional neural networks for multi-classification of thyroid tumor by histopathology: a large-scale pilot study. Yunjun Wang, Qing Guan, Iweng Lao, Li Wang, Yi Wu,Duanshu Li, Qinghai Ji, Yu Wang, Yongxue Zhu, Hongtao Lu, and Jun Xiang. 18, 2019, Annals of Translational Medicine, Vol. 7. doi: 10.21037/atm.2019.08.54.

[91] Machine Learning–Assisted System for Thyroid Nodule Diagnosis. Bin Zhang, Jie Tian, Shufang Pei, Yubing Chen, Xin He, Yuhao Dong, Lu Zhang, Xiaokai Mo, Wenhui Huang, Shuzhen Cong, and Shuixing Zhang. 6, 2019, Thyroid Radiology and Nuclear Medicine, Vol. 29. https://doi.org/10.1089/thy.2018.0380.

[92] TJ Clark, D Voit, JK Gupta, C Hyde, F Song, and KS Khan. Accuracy of hysteroscopy in the diagnosis of endometrial cancer and hyperplasia: a systematic quantitative review. York (UK): Centre for Reviews and Dissemination (UK); 1995-. [Online] 2002. [Cited: March 15, 2020.] https://www.ncbi.nlm.nih.gov/books/NBK69169/.

[93] Application of machine learning to predict the recurrence-proneness for cervical cancer. C.-J. Tseng, C.-J. Lu, C.-C. Chang, G.-D. Chen. 2014, Neural Computing & Applications, Vol. 24, pp. 1311-1316.

[94] Endometrial cancer detection using image processing techniques. Patil, S.N., Kulhalli, Kshama and Patil, S.S. 11, 2014, International Journal of Software & Hardware Research in Engineering, Vol. 2, pp. 20-23. 2347-4890.

[95] A machine learning approach for distinguishing uterine sarcoma from leiomyomas based on perfusion weighted MRI parameters. Mahrooz Malek, Masoumeh Gity, Azadeh Alidoosti, Seyede Mahdieh Seyed Ebrahimi, Elnaz Tabibian, Mohammad Ali Oghabian.

2018, European Journal of Radiology, Vol. 110, pp. 203-211. https://doi.org/10.1016/j.ejrad.2018.11.009.

[96] Hao Sun, Xianxu Zeng, Tao Xu, Gang Peng & Yutao Ma. Computer-aided diagnosis in histopathological images of the endometrium using a convolutional neural network and attention mechanisms. s.l. : https://arxiv.org, 2019. https://arxiv.org/ftp/arxiv/papers/1904/.

[97] Machine Learning Classifiers for Endometriosis Using Transcriptomics and Methylomics Data. Akter Sadia, Xu Dong, Nagel Susan C., Bromfield John J., Pelch Katherine, Wilshire Gilbert B., Joshi Trupti. 2019, Frontiers in Genetics , Vol. 10. 1664-8021.

[98] Deep Convolution Neural Network for Malignancy Detection and Classification in Microscopic Uterine Cervix Cell Images. Shanthi P B, Faraz Faruqi, Hareesha K S, Ranjini Kudva. 11, 2019, Asian Pacific Journal of Cancer Prevention, Vol. 20, pp. 3447-3456. 10.31557/APJCP.2019.20.11.3447.

[99] Proteomic Biomarkers for the Detection of Endometrial Cancer. Kelechi Njoku, Davide Chiasserini, Anthony D. Whetton and Emma J. Crosbie. 10, 2019, Cancers, Vol. 11. https://doi.org/10.3390/cancers11101572.

[100] Deep convolutional neural networks for classifying head and neck cancer using hyperspectral imaging. Halicek, Martin, et al. 6, 2017, Journal of Biomedical Optics, Vol. 22.

[101] Deep learning based classification for head and neck cancer detection with hyperspectral imaging in an animal model. Ling Ma, Guolan Lu, Dongsheng Wang, Xu Wang, Zhuo Georgia Chen, Susan Muller, Amy Chen, and Baowei Fei. s.l. : SPIE--the International Society for Optical Engineering, 2017. Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging. https://doi.org/10.1117/12.2255562.

[102] Using deep learning to enhance head and neck cancer diagnosis and classification. P. Gupta and A. Kaur Malhi. Pondicherry : IEEE, 2018. ieee international conference on system, computation, automation and networking (icscan). pp. 1-6.

[103] Prediction of Metastasis in Head and Neck Cancer from Computed Tomography Images. Tzu-Yun Lo, Peiyin Wei, Chiaheng Yen, Jiing Feng Lirng, Muhhwa Yang, Penyuan Chu, Shinn-Ying Ho. 2018. ICRAI 2018: Proceedings of the 2018 4th International Conference on Robotics and Artificial Intelligence. pp. 18-23. https://doi.org/10.1145/3297097.3297108.

[104] Use of radiomics combined with machine learning method in the recurrence patterns after intensity-modulated radiotherapy for nasopharyngeal carcinoma: a preliminary study. Li S, Wang K, Hou Z, Yang J, Ren W, Gao S, Meng F, Wu P, Liu B, Liu J, Yan J. 648, 2018, Frontiers in Oncology, Vol. 8. doi:10.3389/fonc.2018.00648.

[105] Head and Neck Cancer Detection in Digitized Whole-Slide Histology Using Convolutional Neural Networks. Martin Halicek, Maysam Shahedi, James V. Little, Amy Y. Chen, Larry L. Myers, Baran D. Sumer & Baowei Fei. 2019, Scientific Reports, Vol. 9.

[106] Deep learning in head & neck cancer outcome prediction. André Diamant, Avishek Chatterjee, Martin Vallières, George Shenouda & Jan Seuntjens. 2019, Scientific Reports, Vol. 9.

[107] Adaptive deep learning for head and neck cancer detection using hyperspectral imaging. Ling Ma, Guolan Lu, Dongsheng Wang, Xulei Qin, Zhuo Georgia Chen & Baowei Fei. 18, 2019, Visual Computing for Industry, Biomedicine, and Art, Vol. 2.

[108] An Appraisal of Lung Nodules Automatic Classification Algorithms for CT Images. Wang X, Mao K, Wang L, Yang P, Lu D, He P. 1, 2019, Sensors, Vol. 19. doi: 10.3390/s19010194.

[109] Cancer Diagnosis Using Deep Learning: A Bibliographic Review. Khushboo Munir, Hassan Elahi,Afsheen Ayub,Fabrizio Frezza andAntonello Rizzi. 19, 2019, Cancers, Vol. 11. https://doi.org/10.3390/cancers11091235.

[110] Machine learning applications in cancer prognosis and prediction. Konstantina Kourou, Themis P.Exarchosa, Konstantinos P.Exarchosa, Michalis V.Karamouzis, Dimitrios I.Fotiadis. 2015, Computational and Structural Biotechnology Journal, Vol. 13, pp. 8-17.

[111] Kirk, S., Lee, Y., Lucchesi, F. R., Aredes, N. D., Gruszauskas, N., Catto, J., … Lemmerman, J. Radiology Data from The Cancer Genome Atlas Urothelial Bladder Carcinoma [TCGA-BLCA] collection. s.l. : The Cancer Imaging Archive, 2016. http://doi.org/10.7937/K9/TCIA.2016.8LNG8XDR.

[112] Linehan, M., Gautam, R., Kirk, S., Lee, Y., Roche, C., Bonaccio, E., … Jarosz, R. Radiology Data from The Cancer Genome Atlas Cervical Kidney renal papillary cell carcinoma [KIRP] collection. s.l. : The Cancer Imaging Archive, 2016. http://doi.org/10.7937/K9/TCIA.2016.ACWOGBEF.

[113] Erickson, B. J., Kirk, S., Lee, Y., Bathe, O., Kearns, M., Gerdes, C., … Lemmerman, J. Radiology Data from The Cancer Genome Atlas Liver Hepatocellular Carcinoma [TCGA-LIHC] collection. s.l. : The Cancer Imaging Archive, 2016. http://doi.org/10.7937/K9/TCIA.2016.IMMQW8UQ.

[114] Non–Small Cell Lung Cancer: Identifying Prognostic Imaging Biomarkers by Leveraging Public Gene Expression Microarray Data—Methods and Preliminary Results. Gevaert, O., Xu, J., Hoang, C. D., Leung, A. N., Xu, Y., Quon, A., … Plevritis, S. K. s.l. : Radiological Society of North America (RSNA), 2012, Radiology. http://doi.org/10.1148/radiol.12111607.

[115] Bakr, Shaimaa, et al. Data for NSCLC Radiogenomics Collection. s.l. : The Cancer Imaging Archive, 2017. http://doi.org/10.7937/K9/TCIA.2017.7hs46erv.

[116] Decoding Tumour Phenotype by Noninvasive Imaging Using a Quantitative Radiomics Approach. Aerts HJWL, Velazquez ER, Leijenaar RTH, Parmar C, Grossmann P, Carvalho S, Bussink J, Monshouwer R, Haibe-Kains B, Rietveld D, Hoebers F, Rietbergen MM, Leemans CR, Dekker A, Quackenbush J, Gillies RJ, Lambin P. 2014, Nature Communications, Vol. 5.

[117] Wee, L., & Dekker, A. Data from Head-Neck-Radiomics-HN1 [Data set]. s.l. : The Cancer Imaging Archive, 2019. https://doi.org/10.7937/tcia.2019.8kap372n.

[118] Lingle, W., Erickson, B. J., Zuley, M. L., Jarosz, R., Bonaccio, E., Filippini, J., … Gruszauskas, N. Radiology Data from The Cancer Genome Atlas Breast Invasive Carcinoma [TCGA-BRCA] collection. s.l. : The Cancer Imaging Archive, 2016. http://doi.org/10.7937/K9/TCIA.2016.AB2NAZRP.

[119] The study of PET image segmentation using clustering algorithm. Huiting Qiao, Mengqi Mei, Qi Zhang, Nian Liu, Xin Li, Qiaoyang Huang, Deyu Li. 2013. World Congress on Medical Physics and Biomedical Engineering, IFMBE. pp. 1836–1839.

[120] Artificial Neural Network-Statistical Approach for PET Volume Analysis and Classification. Mhd Saeed Sharif, Maysam Abbod, Abbes Amira, and Habib Zaidi. s.l. : Hindawi Publishing Corporation, 2012, Advances in Fuzzy Systems. doi:10.1155/2012/327861.

[121] A review on segmentation of positron emission tomography images. Brent Foster, Ulas Bagci, Awais Mansoor, Ziyue Xu, Daniel J. Mollura. 1, 2014, Vol. 50.

[122] Comparative methods for PET image segmentation in pharyngolaryngeal squamous cell carcinoma. Habib Zaidi, Mehrsima Abdoli, Carolina Llina Fuentes, and Issam M. El Naqa. 5, 2012, European Journal of Nuclear Medicine and Molecular Imaging, Vol. 39, pp. 881–891. 10.1007/s0025901120530.

[123] Tumor Segmentation on 18F FDG-PET Images Using Graph Cut and Local Spatial Information. Mizuho Nishio, Atsushi K. Kono, Kazuhiro Kubo, Hisanobu Koyama, Tatsuya Nishii, Kazuro Sugimura. 2015, Open Journal of Medical Imaging, Vol. 5, pp. 174-181.

[124] PET image segmentation using a Gaussian mixture model and Markov random fields. Thomas Layer, Matthias Blaickner, Barbara Knäusl, Dietmar Georg, Johannes Neuwirth, Richard P Baum, Christiane Schuchardt, Stefan Wiessalla, and Gerald Matz. 9, 2015, EJNMMI Physics, Vol. 10. doi: 10.1186/s4065801501107.

[125] A Survey of PET Image Segmentation: Applications in Oncology, Cardiology and Neurology. Abir Baâzaouia, Walid Barhoumia, Ezzeddine Zagroubaa and Rostom Mabroukb. 2016, Current Medical Imaging Reviews, Vol. 12, pp. 13-27.

[126] Hierarchies and shape-space for PET image segmentation. Grossiord, É., et al. New York, United States : IEEE, 2015. International Symposium on Biomedical Imaging: From Nano to Macro (ISBI 2015). pp. 1118-1121.

[127] Spatial Fuzzy C-Means PET Image Segmentation of Neurodegenerative Disorder. Raja, A. Meena and K. 1, 2013, Indian Journal of Computer Science and Engineering, Vol. 4, pp. 50-55. ISSN : 0976-5166.

[128] MATLAB and Statistics Toolbox Release 2015a. Natick, Massachusetts, United States : The MathWorks Inc.

[129] A Threshold Selection Method from Gray-Level Histograms. Otsu, N. 1, 1979, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, pp. 62-66.

[130] Segmentation and classification of medical images using texture-primitive features: Application of BAM-type artificial neural network. Neeraj Sharma, Amit K. Ray, Shiru Sharma, K. K. Shukla, Satyajit Pradhan, and Lalit M. Aggarwal. 3, 2008, Journal of Medical Physics, Vol. 33, pp. 119–126. doi: 10.4103/0971-6203.42763.

[131] Robust wide-baseline stereo from maximally stable extremal regions. Matas, J., O. Chum, M. Urba, and T. Pajdla. 2002. Proceedings of British Machine Vision Conference. pp. 384–396.

[132] SURF:Speeded Up Robust Features. Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool. 3, 2008, Computer Vision and Image Understanding, Vol. 110, pp. 346–359.

[133] Efficient Maximally Stable Extremal Region (MSER) Tracking. Michael Donoser, Horst Bischof. 2006. Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 553-560. doi:10.1109/CVPR.2006.107.

[134] Automatic Aerial Target Recognition using A Robust SURF-MSER Feature-based Algorithm. A. FOUAD, H. S. TAHA, Y.Z. ELHALWAGY. Military Technical College, Kobry Elkobbah, Cairo, Egypt : s.n., 2015. 16th International Conference on AEROSPACE SCIENCES & AVIATION TECHNOLOGY, ASAT - 16.

[135] Image Texture Feature Extraction Using GLCM Approach. P. Mohanaiah, P. Sathyanarayana, L. GuruKumar. 5, 2013, International Journal of Scientific and Research Publications, Vol. 3. ISSN 2250-3153.

[136] Machine Learning for High-Speed Corner Detection. Rosten, E., and T. Drummond. 2006. 9th European Conference on Computer Vision. Vol. 1, pp. 430–443.

[137] Histograms of Oriented Gradients for Human Detection. Dalal, N. and B. Triggs. 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 1, pp. 886–893.

[138] On Lines and Planes of Closest Fit to Systems of Points in Space. Pearson, K. 11, 1901, Philosophical Magazine, Vol. 2, pp. 559–572. doi:10.1080/14786440109462720.

[139] Fast and robust fixed-point algorithms for independent component analysis. Hyvarinen, A. 3, 1999, IEEE Transactions on Neural Networks, Vol. 10, pp. 626–634. doi:10.1109/72.761722.

[140] Independent component analysis: An introduction. Tharwat, Alaa. 2018, Applied Computing and Informatics. ISSN 2210-8327.

[141] Independent component analysis: Algorithms and applications. Hyvärinen, A. and Oja, E. 4-5, 2000, Neural Networks, Vol. 13, pp. 411–430. doi:10.1016/S0893-6080(00)00026-5..

[142] Synthetic Minority Over-sampling Technique. Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer. 2002, Journal of Artificial Intelligence Research, Vol. 16, pp. 321-357.

[143] SMOTE: synthetic minority over-sampling technique. N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer. 2002, Journal of artificial intelligence research, pp. 321-357.

[144] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014. arXiv:1412.3555..

[145] Machine Learning for Detection and Diagnosis of Disease. Sajda, Paul. 2006, Annual Review of Biomedical Engineering, Vol. 8. doi: 10.1146/annurev.bioeng.8.061505.095802.

[146] Fuzzy sets. Zadeh, L. 1965, Information and Control, Vol. 8, pp. 338-353.

[147] Rough sets. Pawlak, Z. 1982, International Journal of Computer and Information Sciences, Vol. 11, pp. 341-356.

[148] Instance-based learning algorithms. D. Aha, D. Kibler. 1991, Machine Learning, Vol. 6, pp. 37-66.

[149] Rough sets theory. B. Walczak, D.L. Massart. 1999, Chemometrics and Intelligent Laboratory Systems, Vol. 47, pp. 1-16.

[150] A New Approach to Fuzzy-Rough Nearest Neighbour Classification. R. Jensen, C. Cornelis. 2008. 6th International Conference on Rough Sets and Current Trends in Computing. pp. 310-319.

[151] Fuzzy-rough nearest neighbor algorithms in classification. Sarkar, Manish. 2007, Fuzzy Sets and Systems, pp. 2134 – 2152.

[152] Ridge Estimators in Logistic Regression. le Cessie, S., van Houwelingen, J.C. 1, 1992, Applied Statistics, Vol. 41, pp. 191-201.

[153] Random Forestsx. Breiman, Leo. 1, 2001, Machine Learning, Vol. 45, pp. 5-32.

[154] Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. Powers, David M W. 1, 2011, Journal of Machine Learning Technologies, Vol. 2, pp. 37–63.

[155] Another look at measures of forecast accuracy. Hyndman, Rob J. and Koehler, Anne B. 4, 2006, International Journal of Forecasting, Vol. 22, pp. 679–688. doi:10.1016/j.ijforecast.2006.03.001.

[156] A coefficient of agreement for nominal scales. Cohen, Jacob. 1, 1960, Educational and Psychological Measurement, Vol. 20, pp. 37–46. doi: 10.1177/001316446002000104.

[157] Long Short-Term Memory. Schmidhuber, Sepp Hochreiter and Jürgen. 8, 1997, Neural Computation, Vol. 9, pp. 1735-1780. doi: 10.1162/neco.1997.9.8.1735.

[158] Going Deeper with Convolutions. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. 2014. Conference on Computer Vision and Pattern Recognition. https://arxiv.org/abs/1409.4842..

[159] Deep Residual Learning for Image Recognition. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2015. Conference on Computer Vision and Pattern Recognition. https://arxiv.org/abs/1512.03385.

[160] Hybrid PET/MRI of intracranial masses: Initial experiences and comparison to PET/CT. Boss, Andreas & Bisdas, Sotirios & Kolb, Armin & Hofmann, Matthias & Ernemann, Ulrike & Claussen, Claus & Pfannenberg, Christina & Pichler, Bernd & Reimold, Matthias & Stegger, Lars. 8, s.l. : Society of Nuclear Medicine, 2010, Journal of nuclear medicine, Vol. 51, pp. 1198-1205. doi: 10.2967/jnumed.110.074773.

[161] Kirk, S., Lee, Y., Roche, C., Bonaccio, E., Filippini, J., & Jarosz, R. (2016). Radiology Data from The Cancer Genome Atlas Thyroid Cancer [TCGA-THCA] collection. The Cancer Imaging Archive. http://doi.org/10.7937/K9/TCIA.2016.9ZFRVF1B

[162] Erickson, B. J., Mutch, D., Lippmann, L., & Jarosz, R. (2016). Radiology Data from The Cancer Genome Atlas Uterine Corpus Endometrial Carcinoma (TCGA-UCEC) collection. The Cancer Imaging Archive. http://doi.org/10.7937/K9/TCIA.2016.GKJ0ZWAC

[163] Bi WL, Hosny A, Schabath MB, et al. Artificial intelligence in cancer imaging: Clinical challenges and applications. CA Cancer J Clin. 2019;69(2):127-157. doi:10.3322/caac.21552

[164] Bratulic, S., Gatto, F. & Nielsen, J. The Translational Status of Cancer Liquid Biopsies. Regen. Eng. Transl. Med. (2019). https://doi.org/10.1007/s40883-019-00141-2

[165] Bidgood WD Jr, Horii SC, Prior FW, Van Syckle DE. Understanding and using DICOM, the data interchange standard for biomedical imaging. J Am Med Inform Assoc. 1997;4(3):199-212. doi:10.1136/jamia.1997.0040199

List of Figures

List of Tables

Appendices

Appendix-A: Definitions of Important Terms

Activation Functions	In a typical neural network architecture, an activation function calculates the probability of the result received from its preceding neuron
Cancer	A group of lethal diseases which can spread to other organs of the body
Classification	A method of clubbing data items into different labeled categories according to their similar characteristics
Convolutional Neural Network	A deep neural network where input passes through sets of convolutional layers and pooling layers followed by fully connected layers and generates class scores as output
Data	Raw facts that may be used for producing meaningful information
Deep Learning	A subset of machine learning that has deeper convolutional layers to solve problems expressed as supervised, unsupervised or reinforcement learning
Feature Extraction	A process of culling out inherent attributes or characteristics from a data set
Feature Selection	A process of selecting important features from a set of extracted features by using an algorithm
Image Processing	To mathematically analyze an image by using different algorithms
Image Segmentation	Grouping and separating similar pixels found in a typical image
Loss Function	A loss function measures the difference between the actual and predicted value obtained from a probability model
Medical Imaging	A process of generating imagery impression of

	internal parts of the body for clinical prognosis
Recurrent Neural Network	A branch of artificial neural network that can memorize earlier incidents by including a temporal feedback loop
Review of literature	A study of the existing studies that have contributed to a particular domain of knowledge
Tensor	A generalized form of vectors being used as the data in deep learning that has rank according to its number of axes
Thresholding	A value that acts as a limit for an algorithm to accept or reject a result
Tumor	Abnormal growth in a body part as a result of uncontrolled cell division

Appendix B: Important Abbreviations

AJCC	American Joint Committee on Cancer
ANN	Artificial Neural Network
BLCA	Urothelial Bladder Carcinoma
BRCA	Breast Invasive Carcinoma
CNN	Convolutional Neural Network
CT	Computed Tomography
DNN	Deep Neural Network
FAST	Features from Accelerated Segment Test
FRNN	Fuzzy Rough Nearest Neighbour
GRU	Gated Recurrent Unit
HN	Head-Neck
HOG	Histogram of Oriented Gradients
ICMR	Indian Council of Medical Research
KIRP	Kidney Renal Papillary Cell Carcinoma
LIHC	Liver Hepatocellular Carcinoma
LR	Logistic Regression
LSTM	Long Short Term Memory
MLP	MultiLayer Perceptron
MRI	Magnetic Resonance Imaging
MSER	Maximally Stable Extermal Regions
NSCLC	Non-Small Cell Lung Cancer
PET	Positron Emission Tomography
ReLU	Rectified Linear Unit
RF	Random Forest

SURF	Speeded Up Robust Features
SVM	Support Vector Machines
TCGA	The Cancer Genome Atlas
TCIA	The Cancer Imaging Archive
THCA	Thyroid Cancer
TNM	Tumor-Nodes-Metastases
UCEC	Uterine Corpus Endometrial Carcinoma
WHO	World Health Organization

Appendix C: List of papers presented in seminars/conferences/workshops

1. Dipanjan Moitra, *Performance Evaluation of BioPerl, Biojava, BioPython, BioRuby and BioSmalltalk for Executing Bioinformatics Tasks,* National Conference on Computational Technologies – 2015 (NCCT '15), February 20, 2015, Department of Computer Science & Application (CSA), University of North Bengal (NBU) & Computer Society of India (CSI) Div -V (Education & Research), Region-II, Siliguri Chapter
2. Dipanjan Moitra, *A Study on Bio* Tools for Bioinformatics Tasks,* Third International Conference on Computing & Systems (ICCS-2016), 21st & 22nd January 2016, Department of Computer Science, The University of Burdwan
3. Dipanjan Moitra, *Review of Brain Tumor Detection using Pattern Recognition Techniques,* National Conference on Computational Technologies – 2017 (NCCT '17), February 24, 2017, Department of Computer Science & Application (CSA), University of North Bengal (NBU) & Computer Society of India (CSI) Div -V (Education & Research), Region-II, Siliguri Chapter

Appendix D: List of Publication

1. Dipanjan Moitra, Rakesh Kr. Mandal, Classification of non-small cell lung cancer using one-dimensional convolutional neural network, Expert Systems with Applications (**Elsevier**), Volume 159 (2020), ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113564. (Impact Factor: 5.4; SCI indexed)
2. Dipanjan Moitra, Rakesh Kr. Mandal, Prediction of Non-small Cell Lung Cancer Histology by a Deep Ensemble of Convolutional and Bidirectional Recurrent Neural Network. Journal of Digital Imaging (2020) (**Springer**). https://doi.org/10.1007/s10278-020-00337-x (Impact Factor: 3.6; indexed in SCI and Medline)
3. Dipanjan Moitra, Rakesh Kr. Mandal, Automated grading of non-small cell lung cancer by fuzzy rough nearest neighbour method. Network Modeling Analysis in Health Informatics and Bioinformatics (**Springer**) 8, 24 (2019). https://doi.org/10.1007/s13721-019-0204-6 (Indexed in Scopus)
4. Dipanjan Moitra, Rakesh Kr. Mandal, Automated AJCC (7th edition) staging of non-small cell lung cancer (NSCLC) using deep convolutional neural network (CNN) and recurrent neural network (RNN). Health Information Science and Systems (**Springer**) 7, 14 (2019). https://doi.org/10.1007/s13755-019-0077-1 (Indexed in Pubmed Central)
5. Dipanjan Moitra, Classification of Malignant Tumors: A Practical Approach, LAP LAMBERT Academic Publishing, 2019, ISBN: 978-613-9-47500-1.
6. Dipanjan Moitra, Comparison of Multimodal Tumor Image Segmentation Techniques. International Journal of Advanced Computer Research. Vol. 9. No. 3 (2018). pp. 129–31. ISSN: 0976-5697. https://doi.org/10.26483/ijarcs.v9i3.6010.
7. Dipanjan Moitra, Rakesh Kr. Mandal, Segmentation strategy of pet brain tumor image. Indian Journal of Computer Science and Engineering. Vol. 8, Issue. 5 (2017); pp. 575-577. ISSN: 0976–5166. (indexed in Scopus)
8. Dipanjan Moitra, Rakesh Kr. Mandal, Review of Brain Tumor Detection using Pattern Recognition Techniques, International Journal of Computer Sciences and Engineering, Vol.5, Issue.2 (2017), pp.121-123, ISSN : 2347-2693.
9. Dipanjan Moitra, Rakesh Kr. Mandal. Classification of malignant tumors by a non-sequential recurrent ensemble of deep neural network, International Journal of Medical Informatics (**Elsevier**), ISSN: 1386-5056. Submitted on 6th July 2020 (Communication Ref. No. IJMI_2020_991) (Impact Factor: 3; indexed in SCI and PubMed/Medline)

Appendix E: Copies of two of the papers published in International Journals

Expert Systems with Applications 159 (2020) 113564

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Classification of non-small cell lung cancer using one-dimensional convolutional neural network

Dipanjan Moitra [a,*], Rakesh Kr. Mandal [b]

[a] Department of Management, University of North Bengal, Darjeeling, India
[b] Department of Computer Science and Application, University of North Bengal, Darjeeling, India

ARTICLE INFO

Article history:
Received 22 July 2019
Revised 19 February 2020
Accepted 12 May 2020
Available online 23 May 2020

Keywords:
Lung cancer
Deep learning
Staging
Grading

ABSTRACT

Non-Small Cell Lung Cancer (NSCLC) is a major lung cancer type. Proper diagnosis depends mainly on tumor staging and grading. Pathological prognosis often faces problems because of the limited availability of tissue samples. Machine learning methods may play a vital role in such cases. 2D or 3D Deep Neural Networks (DNNs) has been the predominant technology in this domain. Contemporary studies tried to classify NSCLC tumors as benign or malignant. The application of 1D CNN in automated staging and grading of NSCLC is not very frequent. The aim of the present study is to develop a 1D CNN model for automated staging and grading of NSCLC. The updated NSCLC Radiogenomics Collection from The Cancer Imaging Archive (TCIA) was used in the study. The segmented tumor images were fed into a hybrid feature detection and extraction model (MSER-SURF). The extracted features were clubbed with the clinical TNM stage and histopathological grade information and fed into the 1D CNN model. The performance of the proposed CNN model was satisfactory. The accuracy and ROC-AUC score were higher than the other leading machine learning methods. The study also did well compared to state-of-the-art studies. The proposed model shows that 1D CNN is equally useful in NSCLC prediction like a conventional 2D/3D CNN model. The model may further be refined by carrying out experiments with varied hyper-parameters. Further studies may be conducted by considering semi-supervised or unsupervised learning techniques.

1. Introduction

Non-Small Cell Lung Cancer (NSCLC) is a predominant lung cancer (Horn et al., 2013). The survival rate of patients is more if a tumor is detected early (Nivetha & Manickavasagam, 2014). Biomedical imaging plays an important role in detecting a malignant tumor when clubbed with a Computer-Aided System (CAD) (Moitra & Mandal, 2017), Review of Brain Tumor Detection using Pattern Recognition Techniques, 2017). Besides early detection, proper staging and grading of tumors are also very important to plan proper treatment. Staging shows how much a tumor has already spread and grading shows at what pace it may spread. A typical 4-tier grading system is as follows (Gleason & Mellinger, 2017):

G1: Well differentiated
G2: Moderately differentiated
G3: Poorly differentiated

Other (Type I: Well to moderately differentiated and Type II: Moderately to poorly differentiated).

The most popular tumor staging method is the Tumor-Node-Metastasis (TNM) staging propounded by the American Joint Committee on Cancer (AJCC). The staging nomenclature is as follows (Brierley, Gospodarowicz, & Wittekind, 2017):

T: Size of the primary tumor and if the nearby tissues are affected {Ti; i = 0–4}. Further details may be added by using the lowercase letters a, b & m (multiple)

N: Whether the nearby lymph nodes are affected {Ni; i = 0–3}

M: If other body parts are affected {Mi; i = 0–1}.

2. Objective

In today's context, the use of deep learning has been considerably prevalent as a machine learning technique (Zhang et al., 2019), (Wang et al., 2018). Many studies have so far been conducted to classify Lung Cancer by using Convolutional Neural Networks (Nasrullah et al., 2019; Polat & Danaei Mehr, 2019). Only a few studies tried to identify the stage or grade of a NSCLC (Vesselle, Turcotte, Wiens, & Haynor, 2003), (Vesselle & Toney, 2014). The aim of the present study is to develop a one-dimensional Deep Learning Model for automated staging and grading of NSCLC with better accuracy. The objective was also to com-

* Corresponding author.
E-mail address: dip_anjan@nbu.ac.in (D. Moitra).

https://doi.org/10.1016/j.eswa.2020.113564
0957-4174/

pare the proposed model with a few leading machine learning methods along with a few notable contemporary investigations. The image processing tasks were done by using MATLAB R2015a and the Deep Learning Model was developed in Python 3.6.8 with TensorFlow and Keras. The remaining sections of the paper are Related Studies, Data Acquisition, Methodology, Result, Discussion, and Conclusion.

3. Related studies

Many significant efforts have been made for automated classification of lung cancer by using ANN, CNN and other methods (Rossetto & Zhou, 2017; Wang et al., 2019). Kuruvilla & Gunavathi (2014) did lung cancer classification using artificial neural network. They used CT image and their model gave around 94% accuracy on an average. They extracted the Gray Level Co-occurrence Matrix (GLCM) features and feature selection was done using Principal Component Analysis. The number of features was less and the feature extraction, and the attribute selection method, was primitive. Kumar, Wong, & Clausi (2015) did Lung Nodule Classification by Deep Features in CT Images with LIDC (TCIA) dataset. The accuracy they achieved was near 75%. The system was proposed to classify a tumor as benign or malignant. Song, Zhao, Luo, & Dou (2017) studied Deep Learning for Classification of Lung Nodules on Computed Tomography Images. They used the LIDC-IDRI (TCIA) database and the accuracy of the proposed system was around 84%. Teramoto et al. (Teramoto, Tsukamoto, Kiriyama, & Fujita, 2017) did the Automated Classification of Lung Cancer Types from Cytological Images Using Deep Convolutional Neural Networks. The study achieved an accuracy of 71%. Chaunzwa et al. (2018) used deep-learning radiomics to predict lung cancer histology. They used the LASSO method with CT images and attained an accuracy of around 75%. Monkam et al. (2018) studied CNN models discriminating between pulmonary micro-nodules and non-nodules from CT images using LIDC-IDRI (TCIA) dataset. The system achieved around 88% of accuracy. Sharma, Bhatt, & Joshi (2018) also used the LIDC-IDRI dataset (TCIA) for early detection of lung cancer from CT images. The accuracy they achieved was around 84%. Shaffie et al. (2018) also conducted a study on deep learning-based classification of lung nodules using computed tomography scans. They used publicly available data from the Lung Image Database Consortium. Bhatia, Sinha, & Goel (2019) did Lung Cancer Detection using Deep Learning Residual Approach. The accuracy achieved was 84% on LIDC-IRDI (TCIA) dataset.

From the above review of literature, it may be concluded that none of the aforementioned studies did TNM staging or grading of NSCLC tumors or lesions. The key task that these studies performed was to detect if a tumor was benign or malignant. These studies did the forecast by extracting the spatial attributes directly from the 2-D or 3-D images. These studies mainly used 2-D Convolution layers which generated huge resource consuming architecture. The present study fetched the spatial features from segmented tumor images, clubbed it with the clinical staging and grading data and then used 1-D CNN to classify the information. The study used the spatial information as a rank 2 tensor dataset stored in a CSV (Comma Separated Value) file. 1-D convolutions extract subsequences from sequences. Larger convolution windows can be used with 1-D convnets which is not possible with a 2-D convolution layer. 1-D convents are also faster than 2-D con-

Case ID	Pathological T stage	Pathological N stage	Pathological M stage	AJCC Staging (Version 7)	Histopathological Grade
R01-001	T1a	N0	M0	IA	G2 Moderately differentiated
R01-002	T1a	N0	M0	IA	G1 Well differentiated
R01-003	T3	N0	M0	IIB	Other, Type I: Well to moderately differentiated
R01-004	T1b	N2	M0	IIIA	G2 Moderately differentiated
R01-005	T2a	N0	M0	IB	G3 Poorly differentiated
R01-006	T1b	N0	M0	IA	G1 Well differentiated
R01-007	T1a	N1	M0	IIA	G3 Poorly differentiated
R01-008	Tis	N0	M0	0	G1 Well differentiated
R01-009	T1a	N0	M0	IA	G3 Poorly differentiated
R01-010	T3	N0	M0	IIB	G2 Moderately differentiated
R01-011	T1a	N2	M0	IIIA	G3 Poorly differentiated
R01-012	T2a	N0	M0	IB	G2 Moderately differentiated
R01-013	T2a	N0	M0	IB	G2 Moderately differentiated
R01-014	Tis	N0	M0	0	G1 Well differentiated
R01-015	T2a	N0	M0	IIA	G3 Poorly differentiated

(a)

(b)

Fig. 1. (a) Glimpse of Clinical Data (b) Sample from TCIA NSCLC RadiogGenomics Collection (Case ID: R01-001).

vnets. A 1-D convnet can offer a computationally inexpensive alternative to a 2-D convent.

4. Data acquisition

The NSCLC Radiogenomics Collection (Bakr et al., 2017) (Created by Kirk Smith and last changed by Justin Kirby on Aug 20, 2019) from The Cancer Imaging Archive (TCIA) (Clark et al., 2013) was used in the study. The collection is comprised PET/CT (DICOM) tumor images of 211 patients, semantic annotations of the tumors, segmentation maps of tumors, and quantitative values got from the PET/CT scans. The clinical data includes survival outcomes along with other details like treatment details, genomics, pathology, etc. (Fig. 1) This dataset facilitated the development of predictive medical image biomarkers. The clinical data (CSV) was used for getting the relevant TNM staging and pathological grading information regarding each concerned tumor (Gevaert et al., 2012). Patients were imaged prior to surgery with both thin-section CT and whole-body PET/CT scans gained under IRB approval from Stanford University and the Veterans Administration Palo Alto Health Care System. 10 scans were taken from each of the 162 subjects where TNM staging and grading information were available and 10 scans were randomly taken from the remaining 49 subjects where TNM and grading information were not available. This made the total sample size to 1630. The missing values were replaced by the mode of the class variable.

5. Image processing

All the sample image sizes were reduced to [256*256] and denoised by the Gaussian Blurring technique. Texture based segmentation (Moitra, 2018, Comparison of Multimodal Tumor Image Segmentation Techniques, 2018), (Moitra & Mandal, 2017, SEGMENTATION STRATEGY OF PET BRAIN TUMOR IMAGE, 2017) was carried out with the threshold value determined by Otsu's histogram oriented method (Otsu, 1979) (Fig. 2).

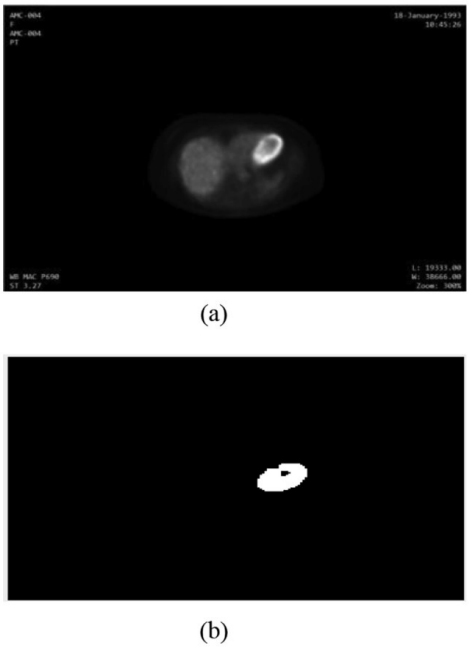


Fig. 2. (a) Original Image (b) Magnified segmented Image of NSCLC tumor.

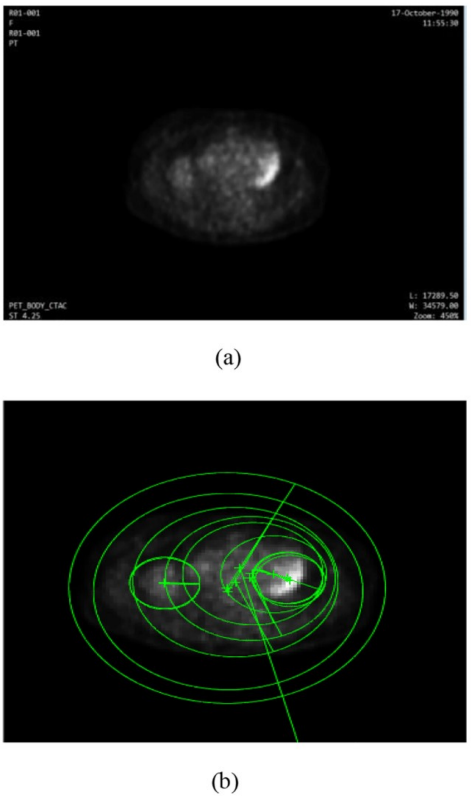


Fig. 3. (a) NSCLC Tumor Image (b) MSER regions and SURF points detected on it.

6. Feature extraction

The Maximally Stable Extremal Regions (MSER) (Matas, Chum, Urba, & Pajdla, 2002) first detected Binary Large Object features by encircling the classification friendly regions and then the Speeded Up Robust Features (SURF) (Bay, Ess, Tuytelaars, & Van Gool, 2008) extracted the scale-independent features from the segmented images (Fig. 3). The dimensionality Reduction was done by Independent Component Analysis (Hyvarinen, 1999). Transformation on numeric data was done using the FastICA algorithm (Hyvärinen & Oja, 2000). First, the data whitening (decoupling transform) was done. Then, the FastICA main loop was executed for 200 iterations with an error tolerance value 1.0E−4 (for solution convergence). Thirty-three (33) features were taken for final classification based on their respective ranks in ICA analysis.

6.1. Maximally stable Extremal regions (MSER)

Let $Q_1, \ldots, Q_i, \ldots \infty$ be a sequence of nested extremal regions ($Q_{i-1} \subset Q_i$). Extremal region Q_{i^*} is maximally stable iff $q(i) = |Q_{i+\Delta} \backslash - Q_{i-\Delta}|/|Q_i|$ has a local minima at i^*. The external regions R_i are

f1	f2	f3	f4	f5	...	f25	f26	f27	f28	f29	f30	f31	f32	f33	T-Stage
0.006897	0.082764	0.059435	79	67.70797	...	0.17429	0.068662	0.090976	0.149021	0.166608	0.237764	0.127928	0.122698	0.099106	T1a
0.010132	0.100149	0.081666	73.44578	49.63855	...	0.190752	0.29852	0.10056	0.062611	0.032788	0.014766	0.024693	0.120736	0.29852	T1a
0.023682	0.15206	0.161641	65.54897	64.35567	...	0.311401	0.161006	0.062116	0.050244	0.04441	0.028175	0.045779	0.124013	0.257344	T3
0.004578	0.067505	0.042163	68.76	73.88	...	0.308194	0.308194	0.188269	0.141006	0.065096	0.007018	0.002479	0.005268	0.015152	T1b
0.018066	0.133196	0.130442	62.19257	57.47297	...	0.246605	0.216608	0.141492	0.130275	0.200024	0.154105	0.074097	0.038471	0.012594	T2a
0.023193	0.150522	0.159013	68.28421	80.12105	...	0.083895	0.078524	0.042517	0.004371	0.032514	0.323828	0.323828	0.149492	0.059702	T1b
0.00885	0.093661	0.07307	65.57931	64.36552	...	0.207944	0.263077	0.21714	0.136991	0.00533	0.005793	0.024564	0.067887	0.105772	T1a
0.020996	0.143375	0.146997	78.29942	57.19477	...	0.276753	0.146699	0.071113	0.095483	0.039818	0.094928	0.050805	0.088894	0.152309	Tis
0.016235	0.126383	0.119746	62.38722	74.08271	...	0.006598	0.029807	0.018356	0.031114	0.033875	0.107816	0.1863	0.265212	0.265212	T1a
0.010498	0.101924	0.084077	73	56.53488	...	0.004886	0.117577	0.051444	0.054425	0.229287	0.295642	0.284459	0.224459	0.064874	T3
0.017212	0.130064	0.125486	82.93617	72.71986	...	0.134222	0.143584	0.192036	0.160101	0.212792	0.092397	0.09633	0.099099	0.125454	T1a
0.003418	0.058365	0.032925	71.5	86.41071	...	0.047549	0.216612	0.204857	0.20198	0.094418	0.107752	0.156852	0.25567	0.25567	T2a
0.010254	0.100744	0.082472	74.42857	60.93452	...	0.1631	0.183474	0.150857	0.141486	0.221363	0.130722	0.090742	0.092472	0.118165	T2a

Fig. 4. A glimpse of the feature set used in the study (T-Staging).

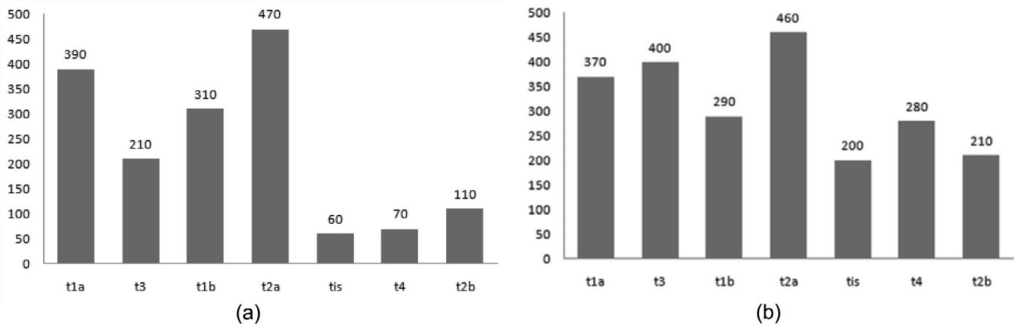


Fig. 5. (a) data before re-sampling, (b) data after re-sampling.

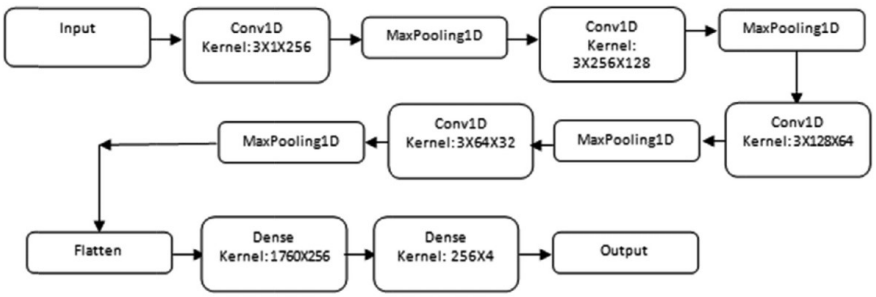


Fig. 6. The proposed 1D CNN model.

detected as follows (Moitra & Mandal, 2019, *Automated grading of non-small cell lung cancer by fuzzy rough nearest neighbour method, 2019a)*:

$$\forall p \in R_i, \forall q \in boundary\ (R_i) => I_{in}(p) >= I_{in}(q) \quad (1)$$

where, I_{in} is the input image.

6.2. Speeded up Robust features (SURF)

The sum of the original image within a rectangle may be evaluated quickly using the integral image,

$$S(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(x, y) \quad (2)$$

To find the point of interest, the Hessian matrix H(p, σ) at point p and scale σ, is used:

$$H(p, \sigma) = \begin{bmatrix} Lxx(p, \sigma) & Lxy(p, \sigma) \\ Lyx(p, \sigma) & Lyy(p, \sigma) \end{bmatrix} \quad (3)$$

Approximate Gaussian second-order derivative with a box filter and extract maxima of the determinant of the Hessian matrix as candidate interest points:

$$Det(H_{approx}) = D_{xx}D_{yy} - (bD_{xy})^2 \quad (4)$$

Here, β is a static constant to compensate for the error introduced by approximating the true Gaussian derivative masks.

Table 1
A glimpse of an Iteration of the Proposed CNN Architecture (Final Layer Outcome for Grading).

Layer (type)	Output Shape	Param #
conv1d_37 (Conv1D)	(None, 31, 256)	1024
max_pooling1d_37 (MaxPooling (None, 31, 256))	0	
conv1d_38 (Conv1D)	(None, 29, 128)	98,432
max_pooling1d_38 (MaxPooling (None, 29, 128))	0	
conv1d_39 (Conv1D)	(None, 27, 64)	24,640
max_pooling1d_39 (MaxPooling (None, 27, 64))	0	
conv1d_40 (Conv1D)	(None, 25, 32)	6176
max_pooling1d_40 (MaxPooling (None, 25, 32))	0	
flatten_10 (Flatten)	(None, 800)	0
dense_19 (Dense)	(None, 256)	205,056
dense_20 (Dense)	(None, 4)	1028

Table 2
Glimpse of an iteration of the proposed CNN architecture (final layer outcome for T-staging).

Layer (type)	Output Shape	Param #
conv1d_169 (Conv1D)	(None, 31, 256)	1024
max_pooling1d_169 (MaxPooling (None, 31, 256))	0	
conv1d_170 (Conv1D)	(None, 29, 128)	98,432
max_pooling1d_170 (MaxPooling (None, 29, 128))	0	
conv1d_171 (Conv1D)	(None, 27, 64)	24,640
max_pooling1d_171 (MaxPooling (None, 27, 64))	0	
conv1d_172 (Conv1D)	(None, 25, 32)	6176
max_pooling1d_172 (MaxPooling (None, 25, 32))	0	
flatten_43 (Flatten)	(None, 800)	0
dense_85 (Dense)	(None, 256)	205,056
dense_86 (Dense)	(None, 7)	1799

Table 3
Glimpse of an Iteration of the Proposed CNN Architecture (Final Layer Outcome for N-staging).

Layer (type)	Output Shape	Param #
conv1d_157 (Conv1D)	(None, 31, 256)	1024
max_pooling1d_157 (MaxPooling (None, 31, 256))	0	
conv1d_158 (Conv1D)	(None, 29, 128)	98,432
max_pooling1d_158 (MaxPooling (None, 29, 128))	0	
conv1d_159 (Conv1D)	(None, 27, 64)	24,640
max_pooling1d_159 (MaxPooling (None, 27, 64))	0	
conv1d_160 (Conv1D)	(None, 25, 32)	6176
max_pooling1d_160 (MaxPooling (None, 25, 32))	0	
flatten_40 (Flatten)	(None, 800)	0
dense_79 (Dense)	(None, 256)	205,056
dense_80 (Dense)	(None, 3)	771

Table 4
Glimpse of an iteration of the proposed CNN architecture (final layer outcome for M−staging).

Layer (type)	Output Shape	Param #
conv1d_37 (Conv1D)	(None, 31, 256)	1024
max_pooling1d_37 (MaxPooling (None, 31, 256))	0	
conv1d_38 (Conv1D)	(None, 29, 128)	98,432
max_pooling1d_38 (MaxPooling (None, 29, 128))	0	
conv1d_39 (Conv1D)	(None, 27, 64)	24,640
max_pooling1d_39 (MaxPooling (None, 27, 64))	0	
conv1d_40 (Conv1D)	(None, 25, 32)	6176
max_pooling1d_40 (MaxPooling (None, 25, 32))	0	
flatten_10 (Flatten)	(None, 800)	0
dense_19 (Dense)	(None, 256)	205,056
dense_20 (Dense)	(None, 6)	1542

Table 5
Comparison of Overall Accuracy of different algorithms.

Algorithm	T Stage	N Stage	M Stage	Grading
SVM	49% (9%)	81% (9%)	98% (2%)	50% (4%)
KNN	68% (8%)	82% (6%)	98% (2%)	83% (5%)
MLP	67% (9%)	82% (5%)	98% (2%)	76% (8%)
RF	67% (6%)	81% (7%)	98% (2%)	85% (3%)
CNN	96% (2%)	94% (2%)	99% (2%)	95% (3%)

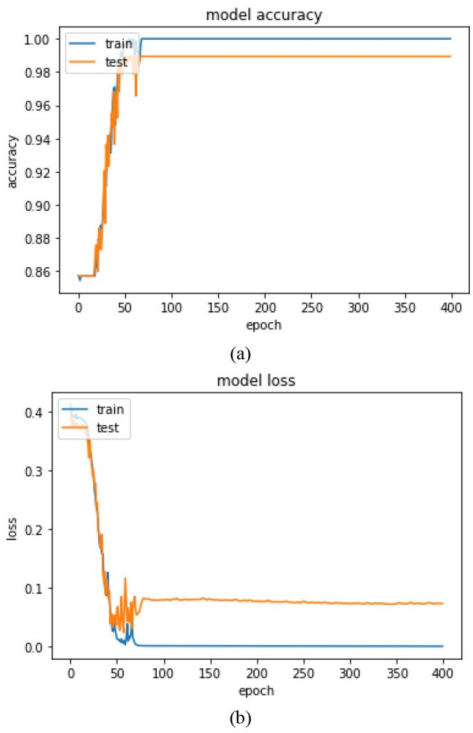


Fig. 7. Sample (a) model accuracy and (b) model loss (for grading).

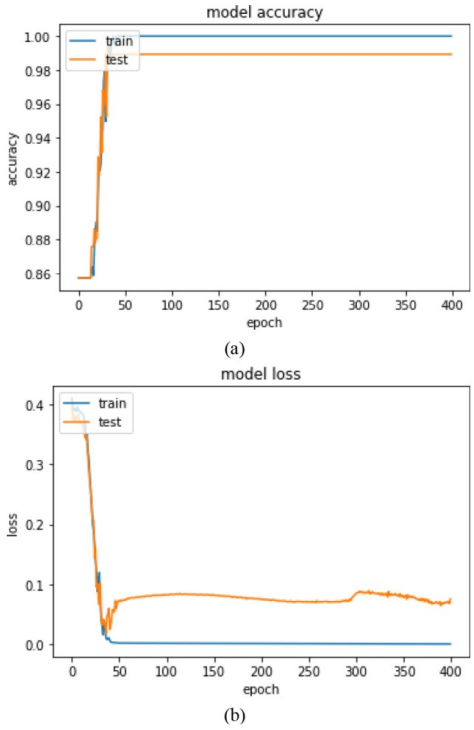


Fig. 8. Sample (a) model accuracy and (b) model loss (for T-staging).

6.3. Mser-SURF

Following is the MSER-SURF algorithm used in the study:
Step 1: De-noised image used as input;
Step 2: Extermal regions were detected using MSER (Eq. (1));
Step 3: Candidate interest points were determined using SURF (Eqs. (2), (3), (4));
Step 4: Finally merged feature vectors were extracted

$$\overrightarrow{F}_{MSER-SURF} = \left\{ \overrightarrow{F}_{MSER}, \overrightarrow{F}_{SURF} \right\} \quad (5)$$

7. Methodology

The pathological data related to the T stage, N stage, M stage and grade were retrieved from the clinical data and fused with the extracted features (Eq. (5)) to form the final dataset (Fig. 4). The dataset thus prepared is software-independent and compatible with any machine learning algorithm. During training the dataset was re-sampled by Synthetic Minority Oversampling TEchnique (SMOTE) (Chawla, Bowyer, Hall, & Kegelmeye, 2002) to balance the minority class, e.g., after class balancing the total sample size became 2210 (Fig. 5) for T-stage. If a dataset has s samples, and f features in the feature space, to oversample, a sample was taken from the dataset, and its k nearest neighbors was considered. A vector between one of the k neighbors was taken. The vector was multiplied by a random number × which lies between 0, and 1. This was added to the current data point to create the new, synthetic data point. The index of the non-empty minority class value (to which SMOTE should be applied) was selected along with 5 as the number of nearest neighbors and the average percentage of SMOTE instances to be created was 100 (with a random seed value of 1).

The re-sampled dataset was fed into the proposed 1D CNN encoder model. A convolution takes two vectors or matrices (all are tensors of different rank) as input – one of them is the original

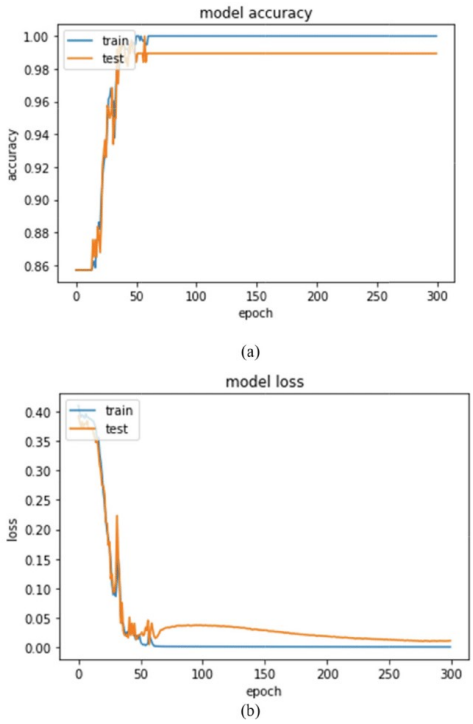


Fig. 9. Sample (a) model accuracy and (b) model loss (for N-staging).

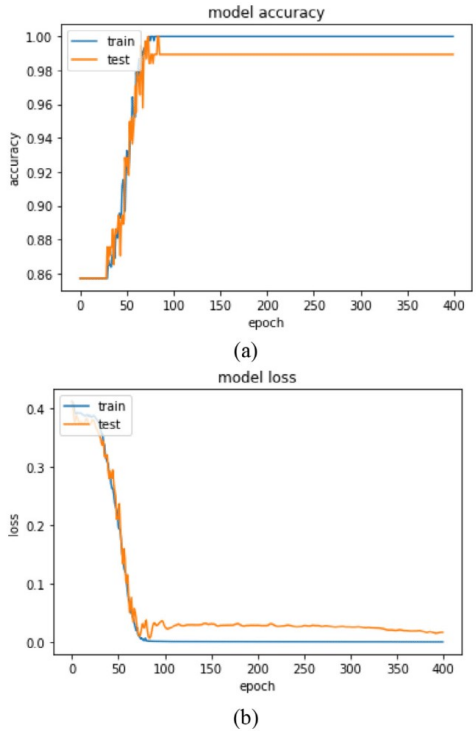


Fig. 10. Sample (a) model accuracy and (b) model loss (for M−staging).

input and the other is the kernel or filter (Kiranyaz et al., 2019). Upon feeding the input in an activation function, a third vector or matrix is generated as the output. If the input vector f has length n and the filter g has length m, the 1D convolution f*g may be defined as:

$$(\mathrm{f} * \mathrm{g})(\mathrm{i}) = \sum_{j=1}^{m} g(i) \cdot f\left(i - j + \frac{m}{2}\right) \tag{6}$$

The kernel is mapped over the input tensor. For each position of the kernel, the overlapping values of the kernel and input tensor are multiplied together and being summed up. This sum of products will be the value of the output at the point in the input where the kernel is centered. The 1D convolutional layers used in the study may be defined as:

$$y^l = b^l + \sum_{i=1}^{n_{l-1}} (f^{l-1} * g^{l-1})(\mathrm{i}) \tag{7}$$

Here, y^l is the l^{th} layer output, b^l is the l^{th} layer bias, f^{l-1} is the $(l-1)^{th}$ layer output and g^{l-1} is the $(l-1)^{th}$ layer filter. The dense layer output O of the model takes the form of a typical perceptron with b as bias, w as weight and x as input:

$$\mathrm{O} = b + \sum_{i=1}^{n} w^{(i)} x^{(i)} \tag{8}$$

The prediction $\hat{y}$ was done by the softmax activation function:

$$\hat{y} = Softmax(O) \tag{9}$$

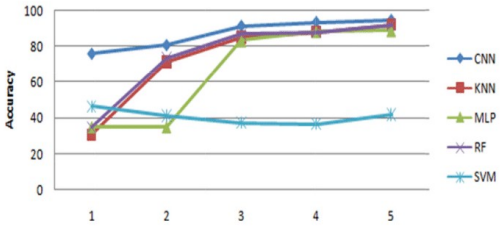


Fig. 11. Grading accuracy of different algorithms.

Where *Softmax* = (exp(oi)/Σjexp (oj)) and oi is the level of confidence for belongingness to category I (10)

The cross-entropy loss may be measured as:

$$l(y, \hat{y}) = -\sum_{j} y_j \log \hat{y}_j \tag{11}$$

Here, y is the actual and ŷis the predicted value. In the proposed model (Fig. 6), the input layer comprised kernel equal to the input image resolution i.e. 256 and in each subsequent layer, the kernel was halved by the Maxpooling layer with *pool size* = 1 and *strides* = 1 i.e., nothing was skipped during pooling (Eqs. (6) and (7)). All the

```
Epoch 499/500
108/108 [==============================] - 0s 4ms/step - loss: 0.0013 -
acc: 1.0000 - val_loss: 0.0260 - val_acc: 0.9894
Epoch 500/500
108/108 [==============================] - 0s 3ms/step - loss: 0.0013 -
acc: 1.0000 - val_loss: 0.0261 - val_acc: 0.9894
54/54 [==============================] - 0s 3ms/step
[0 0 3 1 0 2 0 1 0 0 6 1 0 2 1 1 3 1 1 3 0 3 1 2 0 2 3 3 1 6 3 0 2 1 4 4
0
 1 0 0 3 3 3 0 0 0 1 3 2 6 0 0 0 2]
['IA' 'IA' 'IIB' 'IB' 'IA' 'IIA' 'IA' 'IB' 'IA' 'IA' 'O' 'IB' 'IA' 'IIA'
 'IB' 'IB' 'IIB' 'IB' 'IB' 'IIB' 'IA' 'IIB' 'IB' 'IIA' 'IA' 'IIA' 'IIB'
 'IIB' 'IB' 'O' 'IIB' 'IA' 'IIA' 'IB' 'IIIA' 'IIIA' 'IA' 'IB' 'IA' 'IA'
 'IIB' 'IIB' 'IIB' 'IA' 'IA' 'IA' 'IB' 'IIB' 'IIA' 'O' 'IA' 'IA' 'IA'
 'IIA']
```

Fig. 12. Decoded predictions of AJCC staging by the proposed 1D CNN encoder.

Table 6
Comparison of the proposed CNN Accuracy with a few similar investigations.

Method	Accuracy	Reference
CNN	92%	Liu et al., (Liu & Kang, 2017)
CNN	90%	Dey et al. (Dey, Lu, & Hong, 2018)
CNN	82%	Zhao et al. (Zhao et al., 2018)
CNN	93%	Xie et al., 2018 (Xie, Zhang, Xia, & Zhang, 2018)
CNN	94%	Silva et al. (da Silva, da Silva, Silva, de Paiva, & Gattass, 2017)
SVM	90%	Paing et al., (Paing & Choomchuay, 2017)
Random Forest	82%	Ma et al. (Ma, Wang, Ren, Hu, & Zhao, 2016)
Proposed CNN	96%	The weighted average of staging & grading as found in the Present Study (Table 5)

convolutional layers have the *same* padding i.e., the output has the same length as the original input. L2 or Euclidean Norm regularizers were used to prevent overfitting. After three hidden conv1d layers, the output was flattened and fed into two subsequent fully connected dense layers (Eq. (8)). Here, the kernel of the first dense layer was again increased to match the image resolution and the kernel of the last dense layer was equal to the number of output, e.g., for grading it is equal to 4 as a 4-tier grading system was used in the study. Adam optimizer was used with a learning rate of 1e−4. The batch size used was 128; RelU was activation function for all layers and Softmax was activation function (Eqs. (9) and (10)) for the final output layer, respectively. All the hyperparameters were determined by performing repeated experiments.

The training and test accuracy for each of the T stage, N stage, M stage, and grade were separately measured and the results were also compared with a few of the leading machine learning techniques: KNearest Neighbour (KNN), Random Forest (RF), Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP). The mean accuracy for every method was calculated along with their respective standard deviations (Table 5). Accuracy was measured from the confusion matrix as (Moitra & Mandal, 2019b, Automated grading of non-small cell lung cancer by fuzzy rough nearest neighbour method, 2019b):

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FP} + \text{TN} + \text{FN}) \tag{12}$$

Here, TP = True Positive, FN = False Negative, FP = False Positive, TN = True Negative.

Finally, cross-entropy loss (Eq. (11)) and ROC-AUC scores were measured.

8. Result

Tables 1–4 show the CNN model in the action for Grading, T-staging, N-staging, and M−staging respectively. The initial fluctuations were regularized by the L2 norm. For example, the total number of trainable parameters for M−staging is 336,870 (Table 4) after 500 epochs of training. From Table 1 it may be observed that the activation shape of the input layer is (*None*, 31, 256). Here, *None* implies that the batch length is variable; 31 is the number of features to be fed next, typically calculated as (no. of features – kernel size + 1). Here, 1 is the filter bias. Now, as the kernel or filter size is 3 and the number of features is 33, thus, the number of features to be fed next= (33–3 + 1) = 31. 256 is the number of filters used in the layer. Here, the number of filters resembles the resolution of the image. At each successive iteration, it was halved along with the decrement in the spatial dimension of the input image. Likewise, in conv1d_38, the no. of input to be fed next = (31–3 + 1) = 29. The no. of parameters in conv1d_37 is calculated as (no. of filters*filter size) + layer bias = (256*3 + 256) = 1024. Similarly, the no. of parameters in conv1d_38= (no. of filters*filter size*output filter from conv1d_37) + layer bias= (128*3*256) + 128 = 98, 432 and so on. Maxpooling or flatten layers don't have any learnable parameters. For dense_20 layer, the no. of parameters= (input from dense_19*no. of filters) + bias = (256*4) + 4 = 1024 + 4 = 1028. The fully connected layer i.e. dense_19 has got the highest no. of parameters. Figs. 7–10 show the training and test accuracy and loss, respectively.

9. Discussion

From Table 5, it may be observed that the proposed 1D Convolutional Neural Network (CNN) model had performed satisfactorily. The overall grading accuracy of CNN was approximately (96 ± 3)% which was higher than its nearest competitor Random Forest (85 ± 3)% (Table 5). Similarly, T-stage accuracy of the proposed CNN model was (96 ± 2)% which was higher than its nearest competitor KNN (68 ± 8)% and N-stage accuracy of the CNN was (94 ± 2)% which was higher than its nearest competitor MLP (82 ± 5)% (Table 5). Even the M−stage accuracy of CNN was (99 ± 2)% which was higher than its nearest competitor RF

(98 ± 2)% (Table 5). The initial fluctuations caused by overfitting and dropouts were regularized by the use of L2 norm in the succeeding epochs (Fig. 7, Fig. 8, Fig. 9, and Fig. 10). It was also found that the model accuracy of the proposed CNN model was always higher than the other machine learning algorithms under consideration (Fig. 11). The average ROC-AUC score for the proposed 1D CNN was 0.94 and the average ROC-AUC score (Moitra, 2019) for other machine learning methods was 0.9. The model was even employed to predict the AJCC staging (Edge & Compton, 2010) of NSCLC and the validation accuracy observed was approximately 99% (Fig. 12). All these observations depicted the persistence of the proposed model.

The 1D CNN used in the present study has clearly got an edge over other leading studies conducted in the similar domain (Table 6). Most works done earlier in this domain of study were based on the TCIA LIDC-IDRI dataset and modeled on 2D or 3D CNN. The studies in Table 6 identified lung cancer either as benign or malignant. Thus, most of the works portrayed a binary classification problem and lacked proper staging and grading information. The present study used 1D CNN to predict NSCLC whereas the normal trend is to use 1D CNN in time series problems. The study not only achieved higher overall accuracy than the contemporary studies but also exhibited a prominent way of classifying NSCLC. Such a network may be trained further to get a better result in the staging and grading of NSCLC. Such a study may be useful for NSCLC treatment and also help researchers to pave new ways of automated oncology investigations.

10. Conclusion

The study depicted that 1D CNN can be used efficiently in assessing rank 2 tensors beyond its comfort zone i.e., sequence processing. The model presented in the study worked better than the traditional CNN and other machine learning methods. The aim of developing a lightweight model that may accurately detect NSCLC tumors while consuming less time and resources was successful to a significant extent. The traditional models otherwise need GPU and cloud-enabled services. Such a model may act as a decision support system for oncologists and radiologists. In the future, a semi-supervised or unsupervised model may also be proposed by using 1D CNN to detect NSCLC or other types of tumors.

CRediT authorship contribution statement

Dipanjan Moitra: Conceptualization, Methodology, Software, Data curation, Visualization, Investigation, Formal analysis, Writing - original draft, Writing - review & editing. **Rakesh Kr. Mandal:** Supervision.

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 this paper.

Acknowledgment

The authors wish to thank Dr. Sandy Napel (Stanford) and the entire team of doctors for providing the required NSCLC related Clinical Information (TCIA) used in the study.

References

Bakr, S., Gevaert, O., Echegaray, S., Ayers, K., Zhou, M., Shafiq, M., ... Kadoch, M. (2017). Data for NSCLC Radiogenomics Collection. *The Cancer Imaging Archive*. https://doi.org/10.7937/K9/TCIA.2017.7hs46erv.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). SURF: Speeded up robust features. *Computer Vision and Image Understanding (CVIU), 110*(3), 346–359.

Bhatia, S., Sinha, Y., & Goel, L. (2019). Lung cancer detection: A deep learning approach. *Soft Computing for Problem Solving, 817*, 699–705. https://doi.org/10.1007/978-981-13-1595-4_55.

Brierley, J., Gospodarowicz, M., & Wittekind, C. (2017). *TNM classification of malignant tumours* (8th ed.). Chichester, West Sussex, UK: Wiley-Blackwell.

Chaunzwa, T. L., Christiani, D. C., Lanuti, M., Shafer, A., Diao, N., Mak, R. H., & Aerts, H. (2018). Using deep-learning radiomics to predict lung cancer histology. *Journal of Clinical Oncology, 36*(15). https://doi.org/10.1200/JCO.2018.36.15_suppl.8545.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeye, W. P. (2002). Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research, 16*(1), 321–357.

Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., ... Prior, F. (2013). The Cancer Imaging Archive (TCIA): Maintaining and operating a public information repository. *Journal of Digital Imaging, 26*(6), 1045–1057.

da Silva, G., da Silva, N. O., Silva, A., de Paiva, A., & Gattass, M. (2017). Lung nodules diagnosis based on evolutionary convolutional neural network. *Multimedical Tools Appllication, 76*, 19039–19055. https://doi.org/10.1007/s11042-017-4480-9.

Dey, R., Lu, Z., & Hong, Y. (2018). In *Diagnostic Classification of Lung Nodules Using 3D Neural Networks* (pp. 1–5). Washington, DC, USA: IEEE.

Edge, S., & Compton, C. (2010). The american joint committee on Cancer: the 7th Edition of the AJCC cancer staging manual and the future of TNM. *Annals of Surgical Oncology, 17*(6), 1471–1474. https://doi.org/10.1245/s10434-010-0985-4.

Gevaert, O., Xu, J., Hoang, C. D., Leung, A. N., Xu, Y., Quon, A., ... Plevritis, S. K. (2012). Non-small cell lung cancer: identifying prognostic imaging biomarkers by leveraging public gene expression microarray data-methods and preliminary results. *Radiology, 264*(2), 387–396. https://doi.org/10.1148/radiol.12111607.

Gleason, D. F., & Mellinger, G. T. (2017). Prediction of prognosis for prostatic adenocarcinoma by combined histological grading and clinical staging. *The Journal of Urology, 197*(2S), S134–S139. https://doi.org/10.1016/S0022-5347(02)80309-3.

Horn, L., Eisenberg, R., Gius, D. R., Kimmelshue, K. N., Massion, P. P., Putnam, J. B., ... Carbone, D. P. (2013). *Cancer of the lung: non-small cell lung cancer and small cell lung cancer. In Abeloff, Clinical Oncology* (Fifth Edition. Elsevier Inc..

Hyvarinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks, 10*(3), 626–634. https://doi.org/10.1109/72.761722.

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks, 13*(4–5), 411–430. https://doi.org/10.1016/S0893-6080(00)00026-5.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. (2019). 1D Convolutional Neural Networks and Applications: A Survey. arXiv preprint arXiv:1905.03554.

Kumar, D., Wong, A., & Clausi, D. A. (2015). In *Lung Nodule Classification Using Deep Features in CT Images* (pp. 133–138). Halifax, NS, Canada: IEEE. https://doi.org/10.1109/CRV.2015.25.

Kuruvilla, J., & Gunavathi, K. (2014). Lung cancer classification using neural networks for CT images. *Computer Methods and Programs in Biomedicine, 113*(1), 202–209. https://doi.org/10.1016/j.cmpb.2013.10.011.

Liu, K., & Kang, G. (2017). Multiview convolutional neural networks for lung nodule classification. *International Journal of Imaging Systems and Technology, 27*, 12–22. https://doi.org/10.1002/ima.22206.

Radiomics Approach. Proceedings of the Medical Imaging 2016: PACS and Imaging Informatics: Next Generation and Innovations. San Diego, CA, USA. doi: https://doi.org/10.1117/12.2220768

Matas, J., Chum, O., Urba, M., & Pajdla, T. (2002). Robust wide-baseline stereo from maximally stable extremal regions. In *Proceedings of British Machine Vision Conference* (pp. 384–396).

Moitra, D. (2018). Comparison of multimodal tumor image segmentation techniques. *International Journal of Advanced Research in Computer Science, 9*(3), 129–131. https://doi.org/10.26483/ijarcs.v9i3.6010.

Moitra, D. (2019). *Classification of Malignant Tumors: A Practical Approach. Heinrich-Böcking-Straße 6, 66121*. Saarbrücken, Germany: LAP LAMBERT: Academic Publishing.

Moitra, D., & Mandal, R. (2017). Review of brain tumor detection using pattern recognition techniques. *International Journal of Computer Sciences and Engineering, 5*(2).

Moitra, D., & Mandal, R. (2019). Automated AJCC (7th edition) staging of non-small cell lung cancer (NSCLC) using deep convolutional neural network (CNN) and recurrent neural network (RNN). *Health Information Science and Systems, 7*(1). https://doi.org/10.1007/s13755-019-0077-1.

Moitra, D., & Mandal, R. (2019b). Automated grading of non-small cell lung cancer by fuzzy rough nearest neighbour method. *Network Modeling Analysis in Health Informatics and Bioinformatics, 8*. https://doi.org/10.1007/s13755-019-0077-1.

Moitra, D., & Mandal, R. K. (2017). Segmentation strategy of pet brain tumor image. *Indian Journal of Computer Science and Engineering, 8*(5), 575–577.

Monkam, P., Qi, S., Xu, M., Han, F., Zhao, X., & Qian, W. (2018). CNN models discriminating between pulmonary micro-nodules and non-nodules from CT images. *BioMedical Engineering OnLine, 17*(96). https://doi.org/10.1186/s12938-018-0529-x.

Nasrullah, N., Sang, J., Alam, M., Mateen, M., Cai, B., & Hu, H. (2019). Automated lung nodule detection and classification using deep learning combined with multiple strategies. *Sensors, 19*(17), 1–19.

Nivetha, P., & Manickavasagam, R. (2014). Cancer Detection at Early Stage Using PET/CT Imaging Technique. *International Journal of Innovative Research in Computer and Communication Engineering, 2*(3), 3358–3363.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics, 9*(1), 62–66.

Paing, M., & Choomchuay, S. (2017). Classification of Margin Characteristics from 3D Pulmonary Nodules. Proceedings of the 2017 10th Biomedical Engineering International Conference, (pp. 1-5). Hokkaido, Japan.

Polat, H., & Danaei Mehr, H. (2019). Classification of pulmonary ct images by using hybrid 3D-deep convolutional neural network architecture. *Applied Sciences, 9*(5), 1–15.

Rossetto, A. M., & Zhou, W. (2017). In *Deep Learning for Categorization of Lung Cancer CT Images* (pp. 272–273). Philadelphia, PA, USA: IEEE. https://doi.org/10.1109/CHASE.2017.98.

Shaffie, A., Soliman, A., Fraiwan, L., Ghazal, M., Taher, F., Dunlap, N., ... El-Baz, A. (2018). A generalized deep learning-based diagnostic system for early diagnosis of various types of pulmonary nodules. *Technology in Cancer Research & Treatment, 17*. https://doi.org/10.1177/1533033818798800.

Sharma, M., Bhatt, J. S., & Joshi, M. V. (2018). Early detection of lung cancer from CT images: Nodule segmentation and classification using deep learning. *Tenth International Conference on Machine Vision*. https://doi.org/10.1117/12.2309530.

Song, Q., Zhao, L., Luo, X., & Dou, X. (2017). Using deep learning for classification of lung nodules on computed tomography images. *Computer Vision in Healthcare Applications*. https://doi.org/10.1155/2017/8314740.

Teramoto, A., Tsukamoto, T., Kiriyama, Y., & Fujita, H. (2017). Automated classification of lung cancer types from cytological images using deep convolutional neural networks. *BioMed Research International, 2017*, 1–6. https://doi.org/10.1155/2017/4067832.

Vesselle, & Toney. (2014). Neural Networks for Nodal Staging of Non–Small Cell Lung Cancer with FDG PET and CT: Importance of Combining Uptake Values and Sizes of Nodes and Primary Tumor. Radiology, 270(1).

Vesselle, H., Turcotte, E., Wiens, L., & Haynor, D. (2003). Application of a neural network to improve nodal staging accuracy with 18F-FDG PET in non-small cell. *The Journal of Nuclear Medicine, 44*(12), 1918–1926.

Wang, S. H., Tang, C., Sun, J., Yang, J., Huang, C., Phillips, P., & Zhang, Y. D. (2018). Multiple sclerosis identification by 14-layer convolutional neural network with batch normalization, dropout, and stochastic pooling. *Frontiers in Neuroscience, 12*, 1–11.

Wang, X., Mao, K., Wang, L., Yang, P., Lu, D., & He, P. (2019). An appraisal of lung nodules automatic classification algorithms for CT Images. *Sensors, 19*(1). https://doi.org/10.3390/s19010194.

Xie, Y., Zhang, J., Xia, Y., & Zhang, Y. (2018). Fusing texture, shape and deep model-learned information at decision level for automated classification of lung. *Information Fusion, 42*, 110–112. https://doi.org/10.1016/j.inffus.2017.10.005.

Zhang, Y., Dong, Z., Chen, X., Jia, W., Du, S., Muhammad, K., & Wang, S. (2019). Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications, 78*, 3613–3632. https://doi.org/10.1007/s11042-017-5243-3.

Zhao, X., Liu, L., Qi, S., Teng, Y., Li, J., & Qian, W. (2018). Agile convolutional neural network for pulmonary nodule classification using CT images. *International Journal of Computer Assisted Radiology and Surgery, 13*, 585–595. https://doi.org/10.1007/s11548-017-1696-0.

Prediction of Non-small Cell Lung Cancer Histology by a Deep Ensemble of Convolutional and Bidirectional Recurrent Neural Network

Dipanjan Moitra & Rakesh Kumar Mandal

Journal of Digital Imaging

ISSN 0897-1889

J Digit Imaging
DOI 10.1007/s10278-020-00337-x

Springer

Journal of Digital Imaging
https://doi.org/10.1007/s10278-020-00337-x

ORIGINAL PAPER

Prediction of Non-small Cell Lung Cancer Histology by a Deep Ensemble of Convolutional and Bidirectional Recurrent Neural Network

Dipanjan Moitra [1] · Rakesh Kumar Mandal [1]

Abstract
Histology subtype prediction is a major task for grading non-small cell lung cancer (NSCLC) tumors. Invasive methods such as biopsy often lack in tumor sample, and as a result radiologists or oncologists find it difficult to detect proper histology of NSCLC tumors. The non-invasive methods such as machine learning may play a useful role to predict NSCLC histology by using medical image biomarkers. Few attempts have so far been made to predict NSCLC histology by considering all the major subtypes. The present study aimed to develop a more accurate deep learning model by clubbing convolutional and bidirectional recurrent neural networks. The NSCLC Radiogenomics dataset having 211 subjects was used in the study. Ten best models found during experimentation were averaged to form an ensemble. The model ensemble was executed with 10-fold repeated stratified cross-validation, and the results got were tested with metrics like accuracy, recall, precision, F1-score, Cohen's kappa, and ROC-AUC score. The accuracy of the ensemble model showed considerable improvement over the best model found with the single model. The proposed model may help significantly in the automated prognosis of NSCLC and other types of cancers.

Keywords Lung cancer · Histology · Bidirectional · Recurrent · Neural network

Introduction

Non-small cell lung cancer (NSCLC) accounts for nearly 85% of all lung cancers and a leading cause of cancer-related death worldwide [19]. Prediction of histological subtypes is an important determinant of therapy in NSCLC as it may boost the histopathological grading workup to a significant extent [2]. Major NSCLC subtypes are lung adenocarcinoma and squamous cell carcinoma. Pathological diagnosis of NSCLC often experiences difficulties, as most NSCLC is detected at an advanced stage and samples got from surgical resection are tiny with limited tumor content [1]. This affects the biopsy results, and a proper histology subtype prediction may not be very easy for the radiologists or oncologists. With the advances in precision medicine, medical image biomarkers provide an immense improvement in characterizing a heterogeneous tumor, compared with genomic biomarkers [18]. Thus, there is a need for further study of histological prediction in NSCLC [5] by using non-invasive procedures.

Related Work

Many studies have so far been conducted to classify NSCLC tumors either as benign or malignant [16], but only a few attempts were aimed to predict NSCLC histology subtypes by using non-invasive methods [10]. Most of these studies were gene expression based [3, 7, 11], and others were traditional and advanced machine learning based [8, 9] powered by extracted image features. The gene expression–based methods could not classify non-specified NSCLC tumors successfully [4], and they sometimes lacked sensitivity [6]. Only a handful of notable works have been carried out in NSCLC histology prediction via machine learning methods. In 2016, Weimiao et al. [13] used different classification techniques to classify NSCLC histology. The study used 350 patients' pretreatment CT images along with 440 radiomic features extracted from

✉ Dipanjan Moitra
tataijal@gmail.com

Rakesh Kumar Mandal
rakesh_mndl@rediffmail.com

[1] University of North Bengal, Siliguri, India

Published online: 24 April 2020

the segmented tumor volume. ReliefF and its variants showed improved results as far as the feature selection methods were concerned. Naive Baye's classifier achieved the highest AUC (0.72). The prediction accuracy achieved in the study was not much high, and the study only considered adenocarcinoma and squamous cell carcinoma. Other NSCLC subtypes were not considered in the study. In 2018, Chaunzwa et al. [12] used deep learning to predict NSCLC histology. In the study, deep features were extracted from computed tomography (CT) images by using a pre-trained VGG-16 convolutional neural network. Besides CNN, the other machine learning models used in the study were as follows: K-nearest neighbors (KNN), random forest classifier (RF), and least absolute shrinkage and selection operator (LASSO). The principal component analysis was used to select features. The fully connected CNN had a pivotal performance with AUC = 0.751. The major lack in the study was that all the 157 patients had stage I NSCLC—no advanced stage NSCLC was considered. The study was aimed to identify between adenocarcinoma vs squamous cell carcinoma. This had made it a binary classification problem. In the same year, another study was conducted by Bo He et al. [14]. The aim of the study was to predict the survival status of NSCLC patients from histology. In total, 186 patients' CT images were segmented and 1218 features were extracted through Pyradiomics. SMOTE was the data re-sampling method used in the study. Different random forest models were trained by a hyper-parameters grid search with 10-fold cross-validation, and the model had a prediction accuracy of 89.33% (AUC = 0.9296). Coudray et al. [15] also did a NSCLC histopathology classification in 2018 by using deep learning. Although the AUC was high (around 0.97), it had also considered only adenocarcinoma and squamous cell carcinoma. Thus, there is a huge scope to conduct further studies in this domain to facilitate the identification of the underlying relationship between genomic and medical image features.

Objective

The aim of the present study is to use a deep cross-validated model averaging ensemble in the automatic prediction of NSCLC histology. Unlike its predecessors, the study includes a third NSCLC histology subtype named not otherwise specified (NOS) NSCLC alongside adenocarcinoma and squamous cell carcinoma. This has made the study a multiclass problem. The dataset used in the study [17] contains a cohort of 211 subjects having a diverse mix of AJCC (American Joint Committee on Cancer) staging [34] and histopathological grade (Fig. 1). This also takes the present study ahead of its earlier attempts. The model proposed in the study is a hybrid of convolutional and bidirectional recurrent neural network. The overall goal is to predict NSCLC histology in a more accurate way than its contemporary counterparts. This will help the radiologists and oncologists to detect NSCLC in a more affirmative manner and will also ease the way to plan the relevant treatment strategy. The rest of the paper is divided into the following segments: background, methodology, experiment, result, discussion, and future work.

Background

Recurrent neural network (RNN) and convolutional neural network (CNN) have been used in various fields [20, 21] including computer vision [22, 29]. The semi-automated prognosis of NSCLC is also on the list [23]. But a fully automated diagnosis of NSCLC histology has hardly been tried ever with these deep network techniques. Recurrent networks are mostly used in temporal problems, such as time-series or natural language processing. The problem at hand is spatial. In the present study, a hybrid of CNN and bidirectional RNN is used. This has made it a typical supervised spatio-temporal problem. Here lies the novelty of the study. The description of each technique used in the study is shown below.

A time distributed convolutional model having (H*W) input image array with c channels and k layers may be defined as:

$$H_t[\mathrm{i,j,k}] = \sum_{a=-\Delta}^{\Delta}\sum_{b=-\Delta}^{\Delta}\sum_{c} W[a,b,c,k].X[i+a,j+b,c] \quad (1)$$

In Eq. 1, $X[i,j]$ and $H[i,j]$ represent pixel location (i,j) of an image and hidden state or activation, respectively; W stands for the weight tensor; a and b are convolutional offsets running all over the input images and belong to the range $[-\Delta, \Delta]$ regarding (i,j); t is the time dimension across which the convolution takes place. The output shape $[n_h - k_h + 1] \times [n_w - k_w + 1]$ is given by the difference (1 is the bias) between the input shape $[n_h \times n_w]$ and the convolutional kernel shape $[k_h \times k_w]$. Then, by considering c_i as input channels and c_o as output channels, the output shape will be $ci \times c_o \times [n_h - k_h + 1] \times [n_w - k_w + 1]$.

The output of time distributed CNN can easily be fed into an RNN. A recurrent network can learn from its previous iterations and may be defined as:

$$H_t = \varnothing(X_t \ W_{xh} + H_{t-1} \ W_{hh} + b_h) \quad (2)$$

In Eq. 2, $H_t \in R^{nxh}$ is the hidden state at time t; ø is the activation function; $X_t \in R^{nxd}$, $(t = 1, \ldots, T)$ is the mini-batch of instances with sample size n and d inputs at tth iteration; $W_{xh} \in R^{dxh}$ is the weight parameter with h is the number of hidden states; H_{t-1} is the hidden state from the previous time-step along with its weight parameter $W_{hh} \in R^{hxh}$; b_h is the bias parameter.

the segmented tumor volume. ReliefF and its variants showed improved results as far as the feature selection methods were concerned. Naive Baye's classifier achieved the highest AUC (0.72). The prediction accuracy achieved in the study was not much high, and the study only considered adenocarcinoma and squamous cell carcinoma. Other NSCLC subtypes were not considered in the study. In 2018, Chaunzwa et al. [12] used deep learning to predict NSCLC histology. In the study, deep features were extracted from computed tomography (CT) images by using a pre-trained VGG-16 convolutional neural network. Besides CNN, the other machine learning models used in the study were as follows: K-nearest neighbors (KNN), random forest classifier (RF), and least absolute shrinkage and selection operator (LASSO). The principal component analysis was used to select features. The fully connected CNN had a pivotal performance with AUC = 0.751. The major lack in the study was that all the 157 patients had stage I NSCLC—no advanced stage NSCLC was considered. The study was aimed to identify between adenocarcinoma vs squamous cell carcinoma. This had made it a binary classification problem. In the same year, another study was conducted by Bo He et al. [14]. The aim of the study was to predict the survival status of NSCLC patients from histology. In total, 186 patients' CT images were segmented and 1218 features were extracted through Pyradiomics. SMOTE was the data re-sampling method used in the study. Different random forest models were trained by a hyper-parameters grid search with 10-fold cross-validation, and the model had a prediction accuracy of 89.33% (AUC = 0.9296). Coudray et al. [15] also did a NSCLC histopathology classification in 2018 by using deep learning. Although the AUC was high (around 0.97), it had also considered only adenocarcinoma and squamous cell carcinoma. Thus, there is a huge scope to conduct further studies in this domain to facilitate the identification of the underlying relationship between genomic and medical image features.

Objective

The aim of the present study is to use a deep cross-validated model averaging ensemble in the automatic prediction of NSCLC histology. Unlike its predecessors, the study includes a third NSCLC histology subtype named not otherwise specified (NOS) NSCLC alongside adenocarcinoma and squamous cell carcinoma. This has made the study a multiclass problem. The dataset used in the study [17] contains a cohort of 211 subjects having a diverse mix of AJCC (American Joint Committee on Cancer) staging [34] and histopathological grade (Fig. 1). This also takes the present study ahead of its earlier attempts. The model proposed in the study is a hybrid of convolutional and bidirectional recurrent neural network. The overall goal is to predict NSCLC histology in a more accurate way than its contemporary counterparts. This will help the radiologists and oncologists to detect NSCLC in a more affirmative manner and will also ease the way to plan the relevant treatment strategy. The rest of the paper is divided into the following segments: background, methodology, experiment, result, discussion, and future work.

Background

Recurrent neural network (RNN) and convolutional neural network (CNN) have been used in various fields [20, 21] including computer vision [22, 29]. The semi-automated prognosis of NSCLC is also on the list [23]. But a fully automated diagnosis of NSCLC histology has hardly been tried ever with these deep network techniques. Recurrent networks are mostly used in temporal problems, such as time-series or natural language processing. The problem at hand is spatial. In the present study, a hybrid of CNN and bidirectional RNN is used. This has made it a typical supervised spatio-temporal problem. Here lies the novelty of the study. The description of each technique used in the study is shown below.

A time distributed convolutional model having (H*W) input image array with c channels and k layers may be defined as:

$$H_t[\mathrm{i,j,k}] = \sum_{a=-\Delta}^{\Delta}\sum_{b=-\Delta}^{\Delta}\sum_{c} W[a,b,c,k].X[i+a,j+b,c] \quad (1)$$

In Eq. 1, $X[i,j]$ and $H[i,j]$ represent pixel location (i,j) of an image and hidden state or activation, respectively; W stands for the weight tensor; a and b are convolutional offsets running all over the input images and belong to the range $[-\Delta, \Delta]$ regarding (i,j); t is the time dimension across which the convolution takes place. The output shape $[n_h - k_h + 1] \times [n_w - k_w + 1]$ is given by the difference (1 is the bias) between the input shape $[n_h \times n_w]$ and the convolutional kernel shape $[k_h \times k_w]$. Then, by considering c_i as input channels and c_o as output channels, the output shape will be $ci \times c_o \times [n_h - k_h + 1] \times [n_w - k_w + 1]$.

The output of time distributed CNN can easily be fed into an RNN. A recurrent network can learn from its previous iterations and may be defined as:

$$H_t = \varnothing(X_t \ W_{xh} + H_{t-1} \ W_{hh} + b_h) \quad (2)$$

In Eq. 2, $H_t \in R^{nxh}$ is the hidden state at time t; ø is the activation function; $X_t \in R^{nxd}$, $(t = 1, \ldots, T)$ is the mini-batch of instances with sample size n and d inputs at tth iteration; $W_{xh} \in R^{dxh}$ is the weight parameter with h is the number of hidden states; H_{t-1} is the hidden state from the previous time-step along with its weight parameter $W_{hh} \in R^{hxh}$; b_h is the bias parameter.

Case ID	Histology	AJCC Staging (Version 7)	Histopathological Grade	PET SUVmax	PET SULmax
R01-020	Adenocarcinoma	IIB	G2 Moderately differentiated	5.86	3.86
R01-021	Adenocarcinoma	IA	G2 Moderately differentiated	3.66	2.21
R01-022	Adenocarcinoma	IA	G2 Moderately differentiated	2.68	1.94
R01-023	Adenocarcinoma	IA	G2 Moderately differentiated	9.35	6.67
R01-024	Adenocarcinoma	IIA	G1 Well differentiated	8.09	6.67
R01-025	Adenocarcinoma	IIA	G1 Well differentiated	2.62	2.24
R01-026	Adenocarcinoma	IA	G2 Moderately differentiated	7	5.68
R01-027	NSCLC NOS (not otherwise specified)	IIIA	G3 Poorly differentiated	22.53	18.1
R01-028	Adenocarcinoma	IB	G3 Poorly differentiated	9.53	7.35
R01-029	NSCLC NOS (not otherwise specified)	IIIA	G2 Moderately differentiated	11.49	16.54
R01-030	Adenocarcinoma	IIIA	G1 Well differentiated	2.48	1.96
R01-031	NSCLC NOS (not otherwise specified)	IA	G3 Poorly differentiated	4.3	3.48
R01-032	Adenocarcinoma	IB	Other, Type II: Moderately to poorly differentiated	10.79	9.23
R01-033	Adenocarcinoma	IIIA	Other, Type I: Well to moderately differentiated	1.32	1.05
R01-034	Adenocarcinoma	IIA	G2 Moderately differentiated	7.26	5.52
R01-035	Adenocarcinoma	IIIA	G1 Well differentiated	9.38	7.02
R01-036	Adenocarcinoma	IIIA	G3 Poorly differentiated	11.12	6.3
R01-037	Adenocarcinoma	IB	Other, Type II: Moderately to poorly differentiated	6.71	5.27
R01-038	Squamous cell carcinoma	IB	G2 Moderately differentiated	4.16	3.12
R01-039	Squamous cell carcinoma	IA	G1 Well differentiated	9.1	6.91
R01-040	Squamous cell carcinoma	IB	G2 Moderately differentiated	4.88	3.73
R01-041	Squamous cell carcinoma	IA	G2 Moderately differentiated	3.56	1.55
R01-042	Squamous cell carcinoma	IIA	Other, Type I: Well to moderately differentiated	22.81	17.68

Fig. 1 Glimpse of the clinical dataset used in the study

In Eq. 3, O_t depicts the output (q is the number of outputs):

$$O_t = H_t W_{hq} + b_q \tag{3}$$

Now, long short term memory (LSTM) [31] is a gated version of RNN that we may use along with a bidirectional wrapper. This will help the proposed model to learn from both sides, and the efficacy of the model will be on the higher side. A typical LSTM may be described as:

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i) \tag{4}$$

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f) \tag{5}$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \tag{6}$$

In Eqs. 4, 5, and 6, I_t, F_t, $O_t \in R^{nxh}$ are the input gate, forget gate, and output gate, respectively; W_{xi}, W_{xf}, $W_{xo} \in R^{dxh}$ are weight parameters attached with the input layers of input, forget, and output gates, respectively; W_{hi}, W_{hf}, $W_{ho} \in R^{hxh}$ are weight parameters attached with the hidden layers of input, forget, and output gates, respectively; b_i, b_f, $b_o \in R^{1xh}$ are bias parameters of the input, forget and output gates, respectively. Inputs are processed by a fully connected layer with a sigmoid (σ) activation function and all the gates have a range [0, 1].

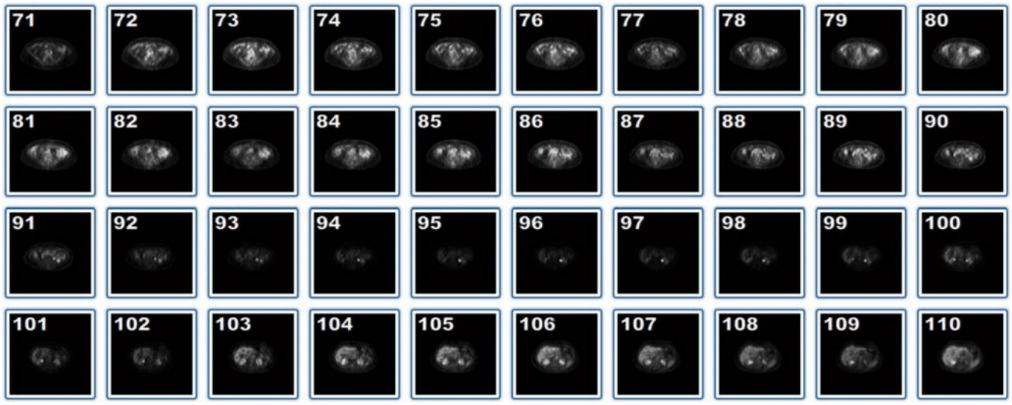


Fig. 2 TCIA NSCLC Radiogenomics (Subject ID R01-001; PET/CT Lung Cancer Series 2; Study UID: ...74044295)

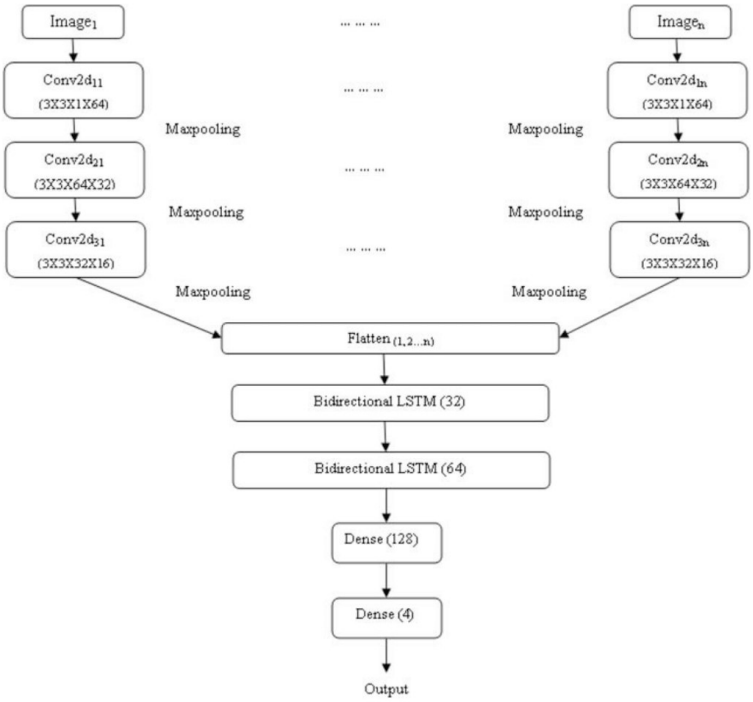


Fig. 3 Structure of the proposed CNN-BiRNN model with sequential image input to the time distributed convolutional layer for applying CNN layer to every temporal slice of the input

Bidirectional RNN is nothing but an improvisation on Eq. 2:

$$\overrightarrow{H}_t = \varnothing\left(X_t \;\; W_{xh}^{(f)} + \overrightarrow{H}_{t-1} \;\; W^{(f)}{}_{hh} + b^{(f)}{}_h\right) \tag{7}$$

$$\overleftarrow{H}_t = \varnothing\left(X_t \;\; W_{xh}^{(b)} + \overleftarrow{H}_{t+1} \;\; W_{hh}^{(b)} + b_h^{(b)}\right) \tag{8}$$

The output expression is the same as that of Eq. 3.

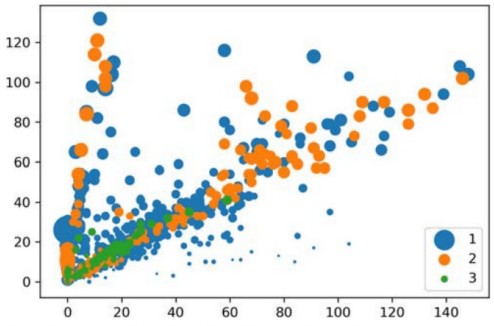


Fig. 4 Re-sampled data loaded for the experiment (1 = adenocarcinoma, 2 = squamous cell carcinoma, 3 = NOS)

The recurrent layered approach of the model may be shown as:

$$\overleftrightarrow{H}_t^{(l)} = \mathrm{f}\left(X_t, \overleftrightarrow{H}_{t-1}^{(l)}\right) \tag{9}$$

$$\overleftrightarrow{H}_t^{(l)} = \mathrm{f}\left(\overleftrightarrow{H}_{t-1}^{(l-1)}, \overleftrightarrow{H}_{t-1}^{(l)}\right) \tag{10}$$

$$O_t = \mathrm{g}\left(H_t^{(L)}\right) \tag{11}$$

In Eqs. 9, 10, and 11, X_t is the mini-batch at time t; l is the hidden layer; f is the layer activation function and g is the activation function of the output layer O_t. Here the forward and backward hidden states are concatenated to get the hidden state H_t and passed on as input to the next bidirectional layer. For all subsequent layers l, the hidden state of the previous layer $(l-1)$ is used in its place. $H^{(L)}$ is the concatenated hidden layer of the preceding bidirectional layers. The final layers are given by the Eqs. 12, 13, and 14:

$$\mathrm{h} = \mathrm{w_h x} + \mathrm{b_h} \tag{12}$$

$$\mathrm{o} = \mathrm{w_o h} + \mathrm{b_o} \tag{13}$$

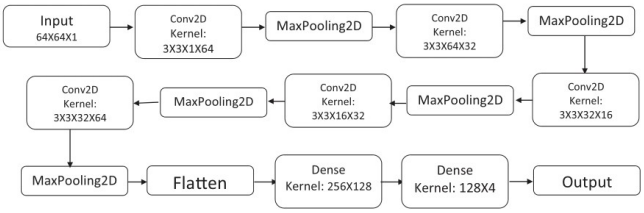


Fig. 5 Structure of a CNN having an architecture similar to the proposed CNN-BiRNN model

The prediction may be measured as

$$\widehat{Y} = \text{softmax}(o) \tag{14}$$

In Eq. 14, $\hat{Y}_i = (\exp(o_i)/\Sigma_j \exp.(o_j))$ and o_i is the level of confidence for belongingness to category i. The loss is measured as the cross-entropy loss and is given by Eq. 15:

$$l\left(Y, \widehat{Y}\right) = -\Sigma_j Y_j \log \widehat{Y}_j \tag{15}$$

Activation function for the final layer is softmax, and for other layers is rectified linear unit,

$$\text{ReLU}(z) = \max(0, z) \tag{16}$$

Methodology

NSCLC Radiogenomics dataset (last updated in August 2019) [24] was used in the study. The dataset comprised positron emission tomography (PET)/computed tomography (CT) [25] images of 211 patients with 1355 numbers of series and 285,411 numbers of images in DICOM format (Fig. 2). The dataset was accompanied by semantic annotations of the tumors, segmentation maps of tumors, and quantitative values got from the PET/CT scans. Thirty best scans per series were observed randomly, and their respective histology data (Fig. 1) were tagged with the concerned images. The original images ($128 \times 128 \times 1$) were down-sampled to ($64 \times 64 \times 1$) to reduce the execution load. Out of 211 subjects, 163 were kept for training and rest for a test. The training and test dataset were finally compressed separately for further use.

The training dataset was loaded and re-sampled by using SVM SMOTE [28] to balance the minority classes. The input layer was a reshaped and time distributed 2D convolutional one. As bidirectional RNN needs both past and future data to work properly, the bidirectional LSTM layers were preceded by hidden time distributed conv2D layers and followed by dense layers. Traditional machine learning algorithms need data pre-processing, feature selection, and segmentation [26, 27]. In the proposed model (Fig. 3), many of these prior works were accomplished by the convolutional layers implicitly. For example, edge detection was done by using the sobel filter. Convolutional layers also down-sample the feature set to have fewer numbers of dot products along the spatial dimensions with the help of maxpooling. The output of the convolutional layer was flattened and fed into the bidirectional LSTM, and again the output was injected in a dense layer which was succeeded by the fully connected layer. The outcome was tested by softmax activation function to measure probability distribution, and the loss was measured by categorical cross-entropy. All the codes were written and implemented by using Python 3.6.8 (IPython 7.5.0) [33] on CPU cores of an Intel(R) Core (TM) i5-3230m CPU @ 2.60GHz processor (x64-based processor).

Experiment

After loading and re-sampling the dataset (Fig. 4), a 10-fold stratified cross-validation was applied on it with a random state of 999, and in each iteration, the data was split into training and test samples. Test class was converted into categorical by using one-hot-encoding. Images were normalized by rescaling. After a prolonged experiment, the desired model was formed. The model comprised three time-distributed conv2d layers with filters, 64, 32, and 16 respectively. Conv2d layers had ReLU as a layer activation function with the same padding means padding the input such that the output has the same length as the original input and default strides. Each conv2d layer was followed by a

Table 1 Evaluation of different best models found

Model	Accuracy	Precision	Recall	F1-score	Cohen's kappa	ROC AUC
Ensemble CNN+BiRNN	96.29	96.29	96.09	96.67	93.61	98.88
Single CNN+BiRNN	92.59	92.59	92.05	92.49	86.95	96.67
CNN	90.00	90.00	90.00	90.01	82.65	95.92

Table 2 Average results of the proposed model observed after the 10th iteration of the experiment

Parameter	Mean with standard deviation
Validation accuracy	94.42 ± 0.005
Cohen's kappa statistics	89.22 ± 0.009
ROC-AUC score	98.47 ± 0.005

time-distributed maxpooling layer, and before injecting in bidirectional LSTM, the output was flattened by a time-distributed flatten layer. After two bidirectional LSTM layers, a fully connected dense layer was there, followed by the dense output layer. Hyper-parameters used were as follows: batch size = 128, learning rate = 1e-1, optimizer = adam. The experiment was carried out for 2000 epochs with early stopping (patience = 200). The best models were saved and later an average ensemble of ten such best models was formed. The ensemble model was executed with 10-fold repeated stratified cross-validation. This time, the experiment was carried out with the validation dataset. The best model thus found was tested with different metrics [30] like accuracy, precision, recall, F1-score, Cohen's kappa, and AUC [32]. The result was compared with the performance of a CNN model having similar architecture (Fig. 5). Thus, in the present study, three pivotal models were used. The first one was the CNN+BiRNN model (Fig. 3) which was the combination of convolutional neural network (CNN) and the bidirectional recurrent neural network (BiRNN). During the training phase, best models were saved and later ten such best models were averaged to form the CNN+BiRNN ensemble. The last model used was a simple CNN model (Fig. 5) having similar architecture to the proposed CNN+BiRNN model.

Result

In the first phase of the experiment, the best model was found at epoch 413 of iteration 6 while executing the single CNN-BiRNN model. The highest test accuracy achieved was

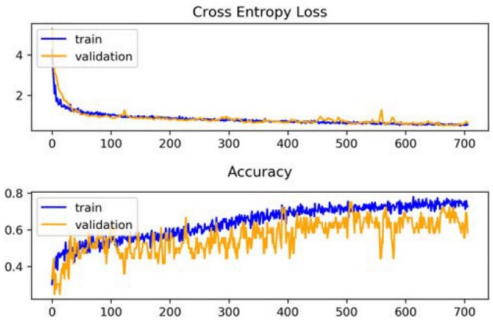


Fig. 7 Glimpse of training and validation loss and accuracy observed in the single CNN-BiRNN model

92.59% (the training accuracy was 81.06%). In the ensemble of CNN-BiRNN, the highest test accuracy was 96.29% found at epoch 450 of iteration 3 (the training accuracy was 85.18%). The test accuracy of a CNN model having similar architecture had 90% test accuracy with a training accuracy of 93% (epoch 492). The comparison of the best models found during the experiment was recorded in Table 1.

The average result depicted by the ensemble CNN-BiRNN model was recorded in Table 2.

The evolution of the experiment conducted was also observed by using evaluative metrics like training and validation loss, training and validation accuracy, kappa and ROC-AUC (receiver operating curve–area under curve) score (Fig. 6).

Discussion

After comparing Tables 1 and 2, the average results achieved by the proposed model ensemble were found far better than the best scores got by the single model or the CNN model. The AUC performance was also convincing than the results got by the contemporary studies discussed in the "Related Work" section.

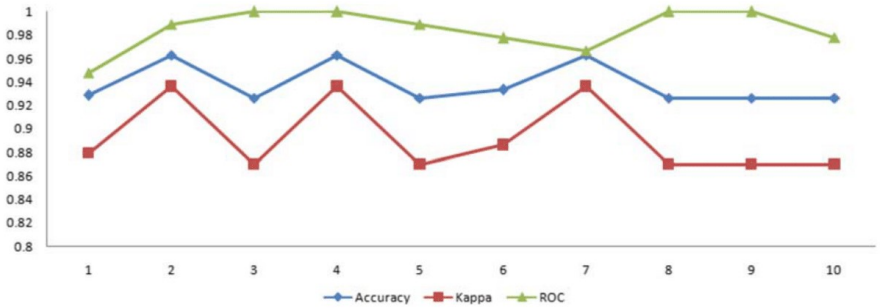


Fig. 6 Iteration wise result of the experiment carried out with CNN-BiRNN ensemble

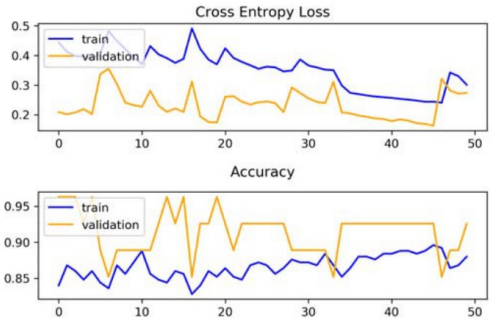


Fig. 8 Glimpse of training and validation loss and accuracy observed in the CNN-BiRNN model ensemble

The iteration-wise performance of the model was also found consistent (Fig. 6). The training and validation curve of the single model (Fig. 7) showed that training loss was less than the validation loss, and training accuracy was more than the validation accuracy. The training and validation curve of the model ensemble (Fig. 8) showed that validation accuracy was higher than the training accuracy. Validation accuracy was more because during training, dropout layers were used and some features became zero to minimize overfitting, but during validation all the features were present and thus it showed better accuracy. The overlapping or over-fitting was also less in the model ensemble. These results may draw a conclusion in favor of the proposed model. The model ensemble will definitely be useful in the automated prognosis of NSCLC, and it will also help radiologists and oncologists by acting as a supportive decision-making mechanism.

Future Work

The proposed model may be refined further by conducting experiments with varied hyper-parameters, e.g., hidden layers, batch size, and patience in early stopping. Experiments may also be carried out with other similar datasets or by combining other meta-learners with the existing model. These types of evaluative studies will develop the usage of prognostic medical image biomarkers with NSCLC and other types of cancers.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval For this type of study formal consent is not required.

Informed Consent Not applicable.

References

1. K. M. Kerr, L. Bubendorf, M. J. Edelman, A. Marchetti, T. Mok, S. Novello, K. O'Byrne, R. Stahel, S. Peters, E. Felip, Panel Members, Rolf Stahel, Enriqueta Felip, Solange Peters, Keith Kerr, Benjamin Besse, Johan Vansteenkiste, Wilfried Eberhardt, Martin Edelman, Tony Mok, Ken O'Byrne, Silvia Novello, Lukas Bubendorf, Antonio Marchetti, Paul Baas, Martin Reck, Konstantinos Syrigos, Luis Paz-Ares, Egbert F. Smit, Peter Meldgaard, Alex Adjei, Marianne Nicolson, Lucio Crinò, Paul Van Schil, Suresh Senan, Corinne Faivre-Finn, Gaetano Rocco, Giulia Veronesi, Jean-Yves Douillard, Eric Lim, Christophe Dooms, Walter Weder, Dirk De Ruysscher, Cecile De Pechoux, Paul De Leyn, Virginie Westeel, Second ESMO consensus conference on lung cancer: pathology and molecular biomarkers for non-small-cell lung cancer, Annals of Oncology, Volume 25, Issue 9, 2014, 1681–1690, https://doi.org/10.1093/annonc/mdu145
2. Wu Weimiao, Parmar Chintan, Grossmann Patrick, Quackenbush John, Lambin Philippe, Bussink Johan, Mak Raymond, Aerts Hugo J. W. L., Exploratory study to identify radiomics classifiers for lung cancer histology, Frontiers in Oncology, Volume 6, 2016, https://doi.org/10.3389/fonc.2016.00071
3. Matthew D. Wilkerson, Jason M. Schallheim, D. Neil Hayes, Patrick J. Roberts, Roy R.L. Bastien, Michael Mullins, Xiaoying Yin, C. Ryan Miller, Leigh B. Thorne, Katherine B. Geiersbach, Kenneth L. Muldrew, William K. Funkhouser, Cheng Fan, Michele C. Hayward, Steven Bayer, Charles M. Perou, Philip S. Bernard, Prediction of lung cancer histological types by RT-qPCR gene expression in FFPE specimens, The Journal of Molecular Diagnostics, Volume 15, Issue 4, 2013, Pages 485–497, ISSN 1525–1578, https://doi.org/10.1016/j.jmoldx.2013.03.007
4. A. Karlsson, H. Cirenajwis, K. Ericson-Lindqvist, C. Reuterswärd, M. Jönsson, A. Patthey, A.F. Behndig, M. Johansson, M. Planck, J. Staaf, 19P Single sample predictor of non-small cell lung cancer histology based on gene expression analysis of archival tissue, Journal of Thoracic Oncology, Volume 13, Issue 4
5. Travis W.D., Brambilla E., Geisinger K.R., Histological grading in lung cancer: One system for all or separate systems for each histological type?, European Respiratory Journal, 2016, 47 (3), pp. 720–723
6. S. Visser, J. Hou, K. Bezemer, L. L. de Vogel, J. P. J. J. Hegmans, B. H. Stricker, S. Philipsen & J. G. J. V. Aerts. Prediction of response to pemetrexed in non-small-cell lung cancer with immunohistochemical phenotyping based on gene expression profiles. BMC Cancer 19, 440 (2019). https://doi.org/10.1186/s12885-019-5645-x
7. Jun Hou, Joachim Aerts, Bianca den Hamer, Wilfred van IJcken, Michael den Bakker, Peter Riegman, Cor van der Leest, Peter van der Spek, John A. Foekens, Henk C. Hoogsteden, Frank Grosveld, and Sjaak Philipsen. (2010) Gene expression-based classification of non-small cell lung carcinomas and survival prediction. PLoS ONE 5(4): e10312. https://doi.org/10.1371/journal.pone.0010312
8. Pooya Mobadersany, Safoora Yousefi, Mohamed Amgad, David A. Gutman, Jill S. Barnholtz-Sloan, José E. Velázquez Vega, Daniel J. Brat, Lee A. D. Cooper, Predicting cancer outcomes from histology and genomics using convolutional networks, Proceedings of the National Academy of Sciences Mar 2018, 115 (13) E2970-E2979; DOI: https://doi.org/10.1073/pnas.1717139115
9. N. Zhang, J. Wu, M. Yang, J. Yu, R. Li, A composite model integrating imaging, histological, and genetic features to predict tumor mutation burden in non-small cell lung cancer patients, International Journal of Radiation Oncology Biology Physics, Volume 105, Issue 1, 2019
10. Digumarthy SR, Padole AM, Gullo RL, Sequist LV, Kalra MK, Medicine (Baltimore), Can CT radiomic analysis in NSCLC predict

histology and EGFR mutation status?, 2019;98(1):e13963. doi: https://doi.org/10.1097/MD.0000000000013963

11. Anna Karlsson, Helena Cirenajwis, Kajsa Ericson-Lindquist, Hans Brunnström, Christel Reuterswärd, Mats Jönsson, Cristian Ortiz-Villalón, Aziz Hussein, Bengt Bergman, Anders Vikström, Nastaran Monsef, Eva Branden, Hirsh Koyi, Luigi de Petris, Patrick Micke, Annika Patthey, Annelie F. Behndig, Mikael Johansson, Maria Planck & Johan Staaf. A combined gene expression tool for parallel histological prediction and gene fusion detection in non-small cell lung cancer. Scientific Reports 9, 5207 (2019) doi:https://doi.org/10.1038/s41598-019-41585-4
12. Tafadzwa Lawrence Chaunzwa, David C. Christiani, Michael Lanuti, Andrea Shafer, Nancy Diao, Raymond H. Mak, Hugo Aerts, Using deep-learning radiomics to predict lung cancer histology, Journal of Clinical Oncology 36, no. 15_suppl (2018), DOI: https://doi.org/10.1200/JCO.2018.36.15_suppl.8545
13. Wu Weimiao, Parmar Chintan, Grossmann Patrick, Quackenbush John, Lambin Philippe, Bussink Johan, Mak Raymond, Aerts Hugo J. W. L., Exploratory study to identify radiomics classifiers for lung cancer histology, Frontiers in Oncology, Vol. 6, 2016, DOI=https://doi.org/10.3389/fonc.2016.00071
14. Bo He, Wei Zhao, Jiang-Yuan Pi, Dan Han, Yuan-Ming Jiang Zhen-Guang Zhang, Wei Zhao, A biomarker basing on radiomics for the prediction of overall survival in non–small cell lung cancer patients, Respir Res (2018) 19: 199. https://doi.org/10.1186/s12931-018-0887-8
15. Nicolas Coudray, Paolo Santiago Ocampo, Theodore Sakellaropoulos, Navneet Narula, Matija Snuderl, David Fenyö, Andre L. Moreira, Narges Razavian & Aristotelis Tsirigos, Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. Nat Med 24, 1559–1567 (2018) doi:https://doi.org/10.1038/s41591-018-0177-5
16. Wang X, Mao K, Wang L, Yang P, Lu D, He P, Sensors (Basel), An appraisal of lung nodules automatic classification algorithms for CT images, 2019;19(1). E194. doi: https://doi.org/10.3390/s19010194
17. Clark K, Vendt B, Smith K, Freymann J, Kirby J, Koppel P, Moore S, Phillips S, Maffitt D, Pringle M, Tarbox L, Prior F. The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository, Journal of Digital Imaging, Volume 26, Number 6, 2013, pp 1045–1057
18. Olivier Gevaert, Jiajing Xu, Chuong D. Hoang, Ann N. Leung, Yue Xu, Andrew Quon, Daniel L. Rubin, Sandy Napel, Sylvia K. Plevritis (2012). Non–small cell lung cancer: identifying prognostic imaging biomarkers by leveraging public gene expression microarray data—methods and preliminary results. Radiology. Radiological Society of North America (RSNA). https://doi.org/10.1148/radiol.12111607, 264, 387, 396
19. Moitra, D. & Mandal, R.K., Automated grading of non-small cell lung cancer by fuzzy rough nearest neighbour method, Netw Model Anal Health Inform Bioinforma (2019) 8: 24. https://doi.org/10.1007/s13721-019-0204-6
20. Zhiyong Cui, Ruimin Ke, Yinhai Wang. Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv:1801.02143
21. Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, Wang-chun Woo. Convolutional LSTM network: a machine learning approach for precipitation nowcasting. arXiv:1506.04214
22. Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. arXiv:1411.4389
23. Moitra, D. & Mandal, R.K. Automated AJCC (7th edition) staging of non-small cell lung cancer (NSCLC) using deep convolutional neural network (CNN) and recurrent neural network (RNN). Health Inf Sci Syst (2019) 7: 14. https://doi.org/10.1007/s13755-019-0077-1
24. Bakr, Shaimaa; Gevaert, Olivier; Echegaray, Sebastian; Ayers, Kelsey; Zhou, Mu; Shafiq, Majid; Zheng, Hong; Zhang, Weiruo; Leung, Ann; Kadoch, Michael; Shrager, Joseph; Quon, Andrew; Rubin, Daniel; Plevritis, Sylvia; Napel, Sandy. (2017). Data for NSCLC Radiogenomics collection. The Cancer Imaging Archive. https://doi.org/10.7937/K9/TCIA.2017.7hs46erv
25. D. Moitra, R. Mandal, Review of brain tumor detection using pattern recognition techniques, International Journal of Computer Sciences and Engineering, Vol. 5, Issue 2, pp. 121–123, 2017
26. D. Moitra, Comparison of multimodal tumor image segmentation techniques, International Journal of Advanced Research in Computer Science, Volume 9, No. 3, 2018, https://doi.org/10.26483/ijarcs.v9i3.6010
27. D. Moitra, R. Mandal, Segmentation strategy of PET brain tumor image, Indian Journal of Computer Science and Engineering, Vol. 8 No. 5 2017, e-ISSN:0976–5166
28. H. M. Nguyen, E. W. Cooper, K. Kamei, Borderline over-sampling for imbalanced data classification, International Journal of Knowledge Engineering and Soft Data Paradigms, 3(1), pp.4–21, 2009
29. François Chollet, Deep learning with Python, Manning Publications Co., 2018, ISBN 9781617294433
30. Powers, David M W (2011) [2007], Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation, Journal of Machine Learning Technologies. 2 (1): 37–63
31. Sepp Hochreiter and Jürgen Schmidhuber, Long short-term memory, Neural Computation, Volume 9 Issue 8, 1997, Pages 1735–1780, doi: https://doi.org/10.1162/neco.1997.9.8.1735
32. Dipanjan Moitra, Classification of malignant tumors: a practical approach, LAP LAMBERT Academic Publishing, 2019, ISBN: 978-613-9-47500-1
33. Dipanjan Moitra, R. K. Samanta, Performance evaluation of BioPerl, Biojava, BioPython, BioRuby and BioSmalltalk for executing bioinformatics tasks, International Journal of Computer Sciences and Engineering, Vol.03, Issue.01, pp.157–164, 2015
34. AJCC Cancer Stating Manual 7th Edition, American Joint Committee on Cancer, ISBN 978-0-387-88440-0, Springer New York

Appendix F: Plagiarism Report

URKUND

Urkund Analysis Result

Analysed Document:	Dipanjan Moitra_Computer Science and Application.pdf (D77348935)
Submitted:	7/31/2020 12:49:00 PM
Submitted By:	nbuplg@nbu.ac.in
Significance:	1 %

Sources included in the report:

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6718995/
https://www.researchgate.net/publication/289535700_A_Combination_of_Shape_and_Texture_Features_for_Classification_of_Pulmonary_Nodules_in_Lung_CT_Images

Instances where selected sources appear:

2

Dipanjan Moitra

(Dipanjan Moitra)
Department of Management,
University of North Bengal
Date: 26/08/2020

(Dr. Rakesh Kr. Mandal)
Associate Professor,
Department of Computer Science and Application,
University of North Bengal
Date: 26/08/2020

Associate Professor
Department of Computer Science & Application
University of North Bengal

Index

S

T

U