Scuola Dottorale di Ingegneria
Sezione di Ingegneria dell'Elettronica Biomedica,
dell'Elettromagnetismo e delle Telecomunicazioni

XXXI CICLO DEL CORSO DI DOTTORATO

(2015-2018)

APPLICATION OF MACHINE AND DEEP LEARNING FOR COLORECTAL CANCER EVALUATION IN 3D MRI

Mumtaz Hussain Soomro

Advisors: Prof. Gaetano Giunta and Prof. Silvia Conforto

PhD Program Coordinator: Prof. Enrico Silva

To my lovely mother and my late father who always supported and prayed for my success. They gave me a spirit to pursue higher studies.

To my respected PhD advisors who always treated me nicely. Their advices, discussions and suggestions enlightened me to be a better person.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and respect to my respected supervisors, Prof. Gaetano Giunta and Prof. Silvia Conforto for giving me the opportunity to undertake a PhD study in biomedical signal and image processing research area, for their consistent support, advice, discussions, critical but invaluable views and comments, which enlighten me to probe and improve further on independent thinking and research skills.

My sincere appreciation also goes to Assoc. Prof. Maurizio Schmid, for providing me with extremely useful technical input and directions. Without his encouragement and helping hands, it was not easy to achieve this goal.

I also want to acknowledge Assistant Prof. Cristiano De Marchis for his significant contribution to this achievement. I much appreciate the time you have spent and the feedback you have given me in helping me to address the technical problems I encountered during my research.

I also would like to acknowledge Assoc. Prof. Stefan Klein and his PhD student Martijn Starmans for their help and invaluable guidance regarding radiomics during my short stay at Erasmus MC-University Medical Center Rotterdam.

Further, I want to extend my gratefulness to the Università degli studi Roma Tre for providing me the scholarship to pursue my PhD studies in applied electronics.

KEYWORDS

Colorectal Cancer Evaluation

3D Magnetic Resonance Imaging (MRI)

Deep Learning

Colorectal Tumor Segmentation

Machine Learning

Colorectal Tumor Classification

ABSTRACT

The aim of this research is twofold. First is related to segmentation of colorectal cancer in 3D MRI, and the second is to characterize the colorectal tumor into two groups; complete responders (CR) and non-responders (NR) to therapy in colorectal cancer. These two studies are conducted in parallel, independently.

Study I: *Objective*: An accurate segmentation of colorectal tumor in 3D magnetic resonance imaging (MRI) volume is an essential requirement in colorectal cancer chemo-radiotherapy. Manual segmentation of colorectal tumor in 3D MRI requires high expertise and subject to laborious work, time consumptions and, inter and intra-observer variability. The primary goal of this research work is to design and develop a straightforward deep learning based algorithm which automatically segments colorectal tumor in 3D T2-weighted (T2w) MRI with reasonable accuracy.

Material and Methods: In this study, T2-weighted (T2w) MRI volumes (those were acquired from 43 patients in a sagittal view on a 3.0 Tesla scanner without contrast agent) are used. These patients are diagnosed with a locally advanced colorectal tumor (cT3/T4). In this work, a novel CNN architecture based on a densely connected neural network for volumetric colorectal tumor segmentation is proposed. The proposed CNN architecture contains multi-scale dense inter-connectivity between layers of fine and coarse scales, thus by leveraging multi-scale contextual information in the network to get a better flow of information throughout the network. Additionally, the 3D level set algorithm was incorporated as a post-processing task to refine the contours of the network predicted segmentation. Cross-validation was performed in 100 rounds by partitioning the dataset into 30 volumes for training and 13 for testing. Three performance metrics were computed to assess the similarity between predicted segmentation and the true ground truth (i.e., manual segmentation by an expert radiologist/oncologist); including Dice similarity coefficient (DSC), recall rate (RR), and average surface distance (ASD).

Results: Above performance metrics were computed in terms of mean and standard deviation (mean ± standard deviation). The DSC, RR, and ASD were (0.84 ± 0.02), (0.85 ± 0.02), and (2.64 ± 2.8) before post-processing; and these performance metrics were (0.86 ± 0.02), (0.87 ± 0.02), and (2.54 ± 2.4) after post-processing, respectively.

Conclusion: We compared our proposed method with other existing volumetric medical image segmentation methods (particularly 3D U-net and DenseVoxNet) in our segmentation task. Experimental results reveal that the proposed method has achieved better performance in colorectal tumor segmentation in volumetric MRI than others have. Besides, the proposed method has total parameters approximately 0.7 million due to its simple network architecture, which is much fewer than DenseVoxNet with 1.8 million and 3D U-net with 19.0 million parameters.

Study II: *Objective*: An accurate diagnosis and staging of colorectal cancer at early basis is the supreme interest in the oncology where medical experts have to decide the treatment plan that a patient should go for either therapy or surgical operation. Radiomics is a semiautomatic/automatic quantitative diagnostic technique that decodes the encoded information in large medical imaging datasets, quantitatively. Radiomics measures tumor heterogeneity for diagnosis of several cancers types non-invasively, thus by providing an accurate prognostic or predictive model. Several studies have been carried out to create radiomics based prognostic model for different clinical issues such as patient survival outcome, treatment response, tumor grading, and more where several types of radiomics were used. Therefore, it is difficult to say that what radiomics features are useful in the assessment of colorectal cancer. Hence, the goal of this work is to find which of the radiomics features are the most appropriate in predicting complete tumor response to neoadjuvant therapy, and to assess the possible correlation among these features.

Methods: 3D MRI used in this study, was consisted on 43 patients. Consequently, among 43 patients, we have 23 patients observed as complete responders and 20 observed as non-responders. Two different types of radiomics features were extracted from our data; traditional handcrafted radiomics features and deep radiomics features. A total of 109 handcrafted radiomics features were calculated from each MRI volume in this study. Furthermore, 4096 deep radiomics features for each patient, are computed using transfer learning from a pre-trained convolutional neural network (CNN_S).

Since high accuracy, efficiency, and reliability are crucial factors in the obtained predictive and prognostic models, which totally depend on the success of radiomics based clinical biomarkers. Therefore, to examine the effectiveness of radiomics based features in achieving an accurate predictive model, it is necessary to validate and compare different machine learning models utilizing all possible radiomics features. For this purpose, in this thesis, the most widely explored supervised machine learning based classifiers were employed. Besides, radiomics have high space dimensionality problem like any high-throughput data-mining field. In this regard, we have assessed the performance of six different feature selection algorithms, which can improve the performance of radiomics based predictive models in different ways. Cross-validation was performed in 100 rounds by partitioning the data as 75% for training and 25% for testing.

Results: Using only handcrafted radiomics features, Artificial Neural Network (ANN) classifier and Fisher as feature selection algorithm have achieved the best predictive performance in term of mean area under the ROC curve, AUC, (i.e., AUCs [mean ± Std]; 0.79 ± 0.016 and 0.8 ± 0.01, respectively). The best prognostic performance using only deep radiomics features was achieved by linear support vector machine (LSVM) classifier and Relief based feature selection algorithm, as 0.8 ± 0.042 and 0.82 ± 0.04, respectively. Whereas, when using a combination of both handcrafted radiomics and deep radiomics features, almost all classifiers in combination with every feature selection algorithm gave better AUC and the best accuracy was given by the LSVM classifier and the Relief based feature selection, as 0.84 ± 0.025 and 0.87 ± 0.013, respectively.

Conclusion: we found that the integration of these both handcrafted and deep radiomics features increases the performance of the majority of predictive models. Moreover, the best performance was given by LSVM with all feature selection methods, and Relief based feature selection algorithms gave the best prognostic performance in combination with all classifiers.

PUBLICATION LIST

International Journal Papers

[1]. **M.H. Soomro**, M. Coppotelli, S. Conforto, M. Schmid, G. Giunta, L. Del Secco, E. Neri, D. Caruso, M. Rengo, A. Laghi, "Automated Segmentation of Colorectal Tumor in 3D MRI Using 3D Multiscale Densely Connected Convolutional Neural Network," *Journal of Healthcare Engineering*, vol. 2019, Article ID 1075434, 11 pages, 2019.

[2]. **M.H. Soomro**, S. Conforto, G. Giunta, S. Ranaldi, C. De Marchis, "Comparison of initialization techniques for the accurate extraction of muscle synergies from myoelectric signals via nonnegative matrix factorization," *Jr. of Applied Bionics and Biomechanics*, vol. 2018, Article ID 3629347, 10 pages, 2018.

[3]. D. Caruso, M. Zerunian, M. Ciolina, Domenico de Santis, **M.H. Soomro**, G. Giunta , S. Conforto, M. Schmid, E. Neri, A. Laghi, "Haralick's texture features for the prediction of response to therapy in colorectal cancer: a preliminary study," *Abdominal Radiology*, vol. 12 (111), pp. 1-7, Nov. 2017.

International Conference Papers

[4]. **M.H. Soomro,** G. De Cola, S. Conforto, M. Schmid, G. Giunta, E. Neri, A. Laghi, " Automatic Segmentation of Colorectal Cancer in 3D MRI by Combining Deep Learning and 3D Level-Set Algorithm: A preliminary study" in 4th IEEE Middle East Conference on Biomedical Engineering (MECBME 2018), March 28-30, 2018.

[5]. **M.H. Soomro**, G. Giunta, A. Laghi, D. Caruso, M. Ciolina, C. De Marchis, S. Conforto, M. Schmid, "Haralick's texture analysis applied to colorectal

T2-weighted MRI: A preliminary study of significance for cancer evolution," in Proc. of *13th IASTED (Biomed 2017),* pp. 16-19, Feb. 2017

[6]. **M.H. Soomro**, G. Giunta, A. Laghi, D. Caruso, M. Ciolina, C. De Marchis, S. Conforto, M. Schmid, "Segmenting MR Images by Level-Set Algorithms for Perspective Colorectal Cancer Diagnosis," In Proc. of *VipIMAGE* 2017, issue (27), pp. 396-406, Oct. 2017.

Publication under submission

[7]. **M.H. Soomro**, S. Conforto, M. Schmid, G. Giunta, L. Del Secco, E. Neri, D. Caruso, M. Rengo, A. Laghi, "Comparison of Prognostic Models for Classification of Colorectal Tumor Response to Therapy in 3D MRI by Combining Radiomics and CNN Features,".

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
CRT	Chemoradiotherapy
GI	Gastrointestinal
MRI	Magnetic Resonance Imaging
CT	Computed Tomography
CR	Complete Responders
NR	Non-responders
DCE-MRI	Dynamic Contrast-Enhanced MRI
DWI	Diffusion-Weighted Images
CNN	Convolutional Neural Network
TME	Total Mesorectal Excision
EPIC	European Prospective Investigation into Cancer and Nutrition
TRG	Tumor Regression Grade
KNN	K-Nearest Neighbour
LR	Logistic Regression
ANNs	Artificial Neural Networks
LD	Linear Discriminant
SVM	Support Vector Machines
NB	Naïve Bayes
MIM	Mutual Information Maximization
MrMr	Minimum redundancy Maximum relevance
3D FCNNs	3D Fully Connected Convolutional Neural Networks
3D MSDenseNet	3D Multiscale Densely Connected Neural Network
DSC	Dice Similarity Coefficient
RR	Recall Rate
ASD	Average Symmetric Surface Distance

CHAPTER 1

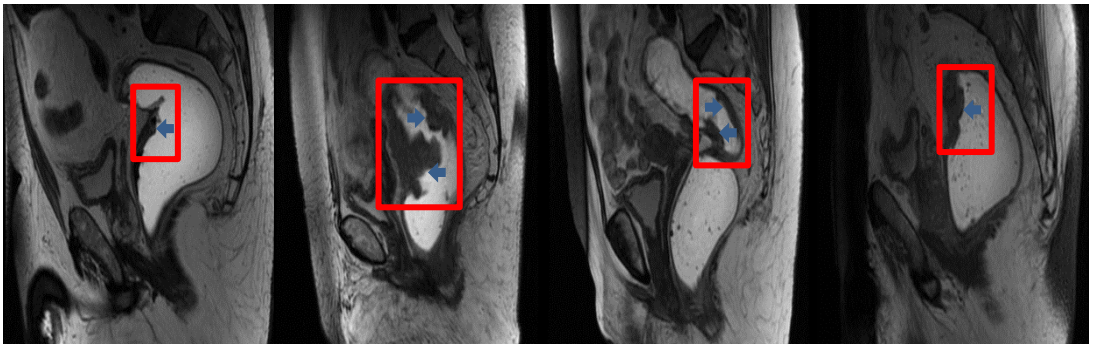
INTRODUCTION

This thesis presents two different research studies, conducted independently, considering the same application. First work is related to the evaluation of deep learning based algorithms for segmenting colorectal cancer/tumor in 3D magnetic resonance imaging (MRI) and second is related to the assessment of machine learning based approaches as being radiomics based prognostic models in prediction of tumor response to neoadjuvant chemoradiotherapy (CRT) in colorectal cancer using 3D MRI.

1.1 Motivation

Colon and rectum are fundamental parts of the gastrointestinal (GI), or digestive system. The colon, which is also called large intestine starts from the small intestine and connects to the rectum. Its primary function is to absorb minerals, nutrients, and water, and remove waste from the body [1-2]. Colorectal cancer or rectum/bowel cancer is diagnosed as the third most common in the United States (US) trailed by lung cancer and breast cancer. Colorectal cancer is the second leading cause of cancer death worldwide; about half of million people die due to this type of cancer where women and men are equally affected [3].

Early diagnosis and accurate staging of colorectal cancer plays a crucial role in oncologic patients' management, especially in personalized treatment plans. Generally, medical diagnosis of suspected cancer is carried out in terms of different medical tests; such as biopsy or medical diagnostic imaging. Nevertheless, the biopsy can provide an informative diagnosis, but it is an invasive diagnostic technique and may not provide heterogeneity of the tumor entirely, which is essential in the evaluation of response to therapy in colorectal chemoradiotherapy (CRT).



(a) (b) (c) (d)

Figure 1.1: An example of visualizing complete responders versus non-responders. (a) and (c) are related to responders and (b) and (d) are non-responders. Red rectangles cover tumor related colorectal parts where blue arrows indicate tumor.

On contrary, diagnostic imaging such as magnetic resonance imaging (MRI) and computed tomography (CT), which are non-invasive and can provide essential information related tumor's characteristics, such as, tumor size and its overall shape, tumor heterogeneity and tumor growth over time; these advantages of medical diagnostic imaging techniques make them more preferable than the biopsy. Currently, MRI is the most widely explored and excellent imaging modality in the loco-regional staging of colorectal cancer [4-5], and T2-weighted MRI has high contrast resolution which differentiates the standard rectal wall from diseased tissues more precisely [6]. Nevertheless, the role of this technique in the assessment of response to therapy is tough as T2-weighted MR images cannot discriminate fibrotic to viable residual tissue during neoadjuvant chemoradiotherapy (CRT) through morphologic approach [7]. Mostly, these studies were carried out based on visual inspection by expert oncologists.

Furthermore, visual evaluation presents many limitations compared to quantitative measurements such as inter-observer variability due to human eye error [8]. As an example, figure 1.1 presents two cases of each complete responders (CR) and non-responders (NR) to neoadjuvant chemoradiotherapy (CRT). Figure 1.1 (a) versus (b), and (c) versus (d) show a visual comparison between responders versus non-responders, respectively. In the first case, figure 1.1 (a) versus (b); one may differentiate between CR and NR. Whereas in the second case, figure 1.1 (c) versus (d); it is challenging to say which is CR and which is NR. For this reason, a possible approach to overcome this issue, a multiparametric approach was proposed [9-10]. The multiparametric

approach is a combination of any two different modalities including T2-weighted, diffusion-weighted images (DWI) [11-13] and dynamic contrast-enhanced MRI (DCE-MRI) [14-15]. However, the multiparametric approaches have achieved improved results but not optimal, which could not ensure a personalized treatment to patients, and they could not evaluate the tumor at the cellular level, as well.

Therefore, taking into account the weaknesses in the procedure of visual inspection, there exists a need to develop an accurate semiautomatic/automatic quantitative diagnostic technique [16]. In this regard, new MRI biomarkers, such as texture analysis, were recently investigated [16-20], [22-24]. Texture analysis is a non-invasive quantitative technique, which assesses tissue heterogeneity of tumor lesions quantitatively [17]. Typically, these textural features assess the spatial variation of gray levels within the observed image/s by utilizing different statistics (i.e., first order, second order, or higher order statistics) [19, 21]. Recently, first order textural features were extracted from T2-weighted images of rectal cancer and these extracted first order textural features played a potential role as being prognostic biomarkers of tumor response to neoadjuvant CRT [17-18]. However, these studies were conducted at an initial experimental basis with limited datasets.

Presently, radiomics [25-33]; semiautomatic/automatic quantitative diagnostic technique that decodes the encoded information in large medical imaging datasets, quantitatively. Radiomics measures tumor heterogeneity for diagnosis of several cancers types invasively, thus by providing a prognostic or predictive model. More precisely, the radiomics is the method, which extracts features from the medical images quantitatively, and these extracted quantitative features are analyzed to get predictive or prognostic models in personalized treatment strategies. Additionally, radiomics based features can be divided into two categories: 1) handcrafted radiomics features, and 2) deep radiomics features [34-38]. The handcrafted radiomics features are based on the shape-based, first order, and second order or high order textural features. The handcrafted radiomics process is usually conducted on four primary tasks, including data acquisition and processing; segmentation of a region of interest (ROI) in the data; features extraction and quantification; data integration, feature selection, and model building. Whereas in deep radiomics, features are extracted by employing deep learning based approaches [32]. Several studies have been carried out to create radiomics based

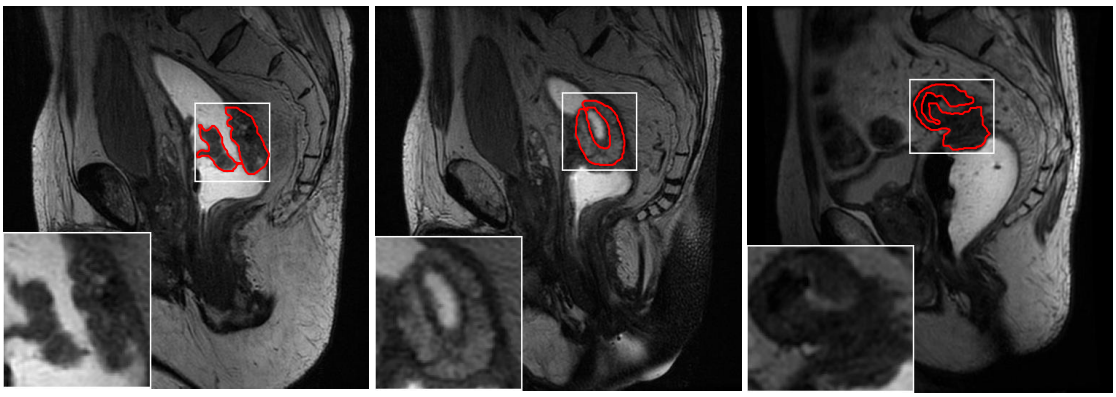


Figure 1.2: An illustration of colorectal tumor location, intensity and size variation in different slice of a same volume where cancerous region is contoured by red marker.

prognostic model for different clinical issues such as patient survival outcome [25, 38], treatment response [17-18], tumor grading [26-28], and more [34-37].

An accurate diagnosis and staging of colorectal cancer at early basis is the supreme interest in the oncology where medical experts have to decide the treatment plan that a patient should go for either therapy or surgical operation. In literature, multiple radiomics based features have been incorporated for different purposes; therefore, it is difficult to say that what radiomics features are useful in the assessment of colorectal cancer. Thus, the goal of this work is to find which of the radiomics feature are the most appropriate in the prediction of complete tumor response to neoadjuvant therapy and to assess the possible correlation among these features.

Beside this work, accurate segmentation of colorectal tumor in 3D magnetic resonance imaging (MRI) volume is an essential requirement in colorectal cancer chemoradiotherapy. Commonly, the oncologist or radiologist delineates colorectal tumor regions from volumetric MRI data manually. This manual delineation or segmentation is time-consuming and laborious and presents inter and intra-observer variability. Therefore, there exists a need for efficient automatic colorectal tumor segmentation methods in clinical radiotherapy practices to segment colorectal tumor from large volumetric data, as this may save time, and reduce human interventions. In contrast to natural images, medical imaging is generally more chaotic, as the shape of the cancerous regions may vary from slice to slice, as shown in figure 1.2. Hence, automatically segment colorectal tumor is a very challenging task, not only because its

size may be tiny, but also because of its somewhat inconsistent behavior in terms of shape and intensity distribution.

Lately, automatic segmentation of colorectal tumor from volumetric MRI data based on atlas [39] and super-voxel clustering [40] have been presented with some good performance. Newly, deep learning-based approaches have been explored with impressive results in medical image segmentation [41-49]: Trebeshi et al. [41], have presented a deep learning-based automatic segmentation method to localize and segment rectal tumor in multiparametric MRI by incorporating a fusion between T2-weighted (T2w) MRI and diffusion-weighted imaging (DWI) MRI. Despite their method displaying good performance, it is unclear whether only T2w modality, which provides more anatomy information than DWI modality, could be useful for colorectal tumor segmentation. Secondly, they employed their implementation on 2D data, as most deep learning algorithms are intrinsically designed in 2D nature, while medical data, such as CT (Computed Tomography) and MRI are in 3D volumetric form. These 2D Convolutional Neural Network (CNN) algorithms segment the volumetric MRI or CT data in a slice-by-slice sequentially [42-44], where 2D kernels are used by aggregating axial, coronal, and sagittal planes in a one-to-one association, individually. Although these 2D CNN-based methods demonstrated vast improvement in segmentation accuracy in comparison to traditional machine learning algorithms [50], the inherent 2D nature of the kernels limits their application when using volumetric spatial information. Taking into account the problems in 2D CNN for segmenting colorectal tumor in 3D MRI, there exists a need of 3D CNN, such as 3D U-net [45], where 3D kernels are used instead of 2D to extract spatial information across all three volumetric dimensions.

1.2 Objectives

From the above discussion in Section 1.1, this study has two-fold objectives consequently. First is related to segmentation of colorectal cancer in 3D MRI, and second is to characterize the colorectal tumor into two groups; complete responders (CR) and non-responders (NR) to therapy in colorectal cancer. Accordingly, the research objectives related to this study are enlisted as follows,

a) To design and develop deep learning based algorithm that can automatically segment colorectal tumor in 3D MRI volumetric data.

b) To design and develop radiomics based prognostic model as biomarkers for prediction of tumor response to therapy in colorectal cancer.

1.3 Contributions to Knowledge

In this thesis, we presented three different novel contributions related to automatic segmentation of colorectal tumor in volumetric MRI data and modeling of radiomics based predictive model. First contribution: we have incorporated three different 3D CNN based segmentation algorithms, which have successfully been applied in the segmentation of volumetric medical images for different applications. These methods were employed in our application to segment colorectal tumor in 3D MRI. Second contribution: Based on the pros and cons of those 3D CNN based methods, we proposed a novel algorithm, 3D Multiscale Densely connected neural network (3D MSDenseNet). The third contribution of this thesis is; assessment of predictive model using radiomics features in three different cases; Case 1: Predictive model using only handcrafted radiomics features, Case 2: Predictive model using only deep radiomics features, and Case 3: Predictive model using a combination of handcrafted and deep radiomics features, respectively. Considering radiomics features using any case from the above three cases; different feature selection and several predictive modeling can produce an optimal radiomics based biomarker in tumor response to colorectal therapy. In this regard, we incorporated six different feature selection algorithms and seven different classification algorithms.

1.4 Thesis Organization

This thesis presents a series of self-reliant chapters and these chapters can be studied at large extent, individualistically. Chapters 2, 3, 4 and 5 present the theoretical backgrounds related to the workflow of this research study. The experimental works

related to core contributions of this thesis are presented in chapters 6 and 7, respectively. A summary of each chapter of this thesis is presented below:

Chapter 2: Background on Colorectal Cancer

Chapter 2 deals with the comprehensive details of colorectal cancer. It begins from colorectal anatomy, pathological anatomy of colorectal cancer, risk factors of getting colorectal cancer and followed by quantitative and non-quantitative methodologies for colorectal cancer staging. This chapter is written with the collaboration of the Department of Radiological Sciences, University of Pisa, Via Savi 10, 56126 Pisa, Italy.

Chapter 3: Background on Feature Selection and Machine Learning Algorithms

Since high accuracy, efficiency and reliability are crucial factors in the obtained predictive and prognostic models, which totally depend on the success of radiomics based clinical biomarkers. Thus, in order to examine the effectiveness of radiomics based features in obtaining an accurate predictive model; it is necessary to validate and compare different machine learning models utilizing all possible radiomics features. Besides, radiomics have high space dimensionality problem like any high-throughput data-mining field. In this regard, different feature selection algorithms can improve the performance of radiomics based predictive models in different ways. Consequently, different feature selection algorithms should be assessed along with different machine learning models. This chapter gives a comprehensive revision on widely explored supervised machine learning algorithms and commonly used filter-based feature selection algorithms.

Chapter 4: Background on Deep Learning

Chapter 4 presents the advantages of deep learning and basic building blocks of deep learning.

Chapter 5: 3D Colorectal Tumor Segmentation

This chapter discusses 3D deep learning algorithms, which have successfully been applied in different medical image segmentation applications, mainly 3D fully connected convolutional neural network (3D FCNNs), 3D U-net and 3D DenseVoxNet. These methods have been incorporated in our application for segmenting colorectal tumor in 3D MRI. Furthermore, this chapter presents the pros and cons of above methods and, based on these pros and cons, this chapter presents a novel methodology

for segmenting the colorectal that is: 3D Multiscale Densely connected neural network (3D MSDenseNet).

Chapter 6: 3D Colorectal Tumor Segmentation — Experimental Results

This chapter gives comprehensive experimental detail on segmentation of colorectal tumor in volumetric MRI. This chapter presents the validation of the proposed segmentation method for volumetric segmentation of colorectal tumor in 3D MRI. In addition, the proposed method was compared with other deep learning-based baseline methods those were discussed in chapter 5.

Chapter 7: Classification of Responders and Non-Responders –Experimental Results

This chapter discusses three different types of radiomics features, handcrafted based radiomics, deep features based radiomics and their combination. Based on these radiomics features, this chapter evaluates the seven different families of supervised machine learning algorithms and six different feature selection algorithms as being a prognostic model for prediction of response to therapy in colorectal cancer.

Chapter 8: Conclusion

This chapter summarizes the thesis and offers some future research directions.

CHAPTER 2

BACKGROUND ON COLORECTAL CANCER

This chapter is written in collaboration with staffs of Department of Radiological Sciences, University of Pisa, Via Savi 10, and 56126, Pisa, Italy. The aim of this chapter deals with an understanding of the basic functional anatomy of colorectal, risk of getting colorectal cancer, colorectal cancer epidemiology, and to know about medical diagnostic techniques for colorectal cancer evaluation. This chapter presents the treatment procedures for colorectal cancer, especially, neoadjuvant chemoradiotherapy (CRT). Also, assessment of neoadjuvant chemoradiotherapy is explained in this chapter. Basis on this evaluation process, patients may be categories into complete responders and non-responders to therapy.

2.1 Anatomy of Colorectum

Colon or large bowel is a vital part of the gastrointestinal tract, which starts from the esophagus to anus [1], as shown in figure 2.1. Colon is further divided into five

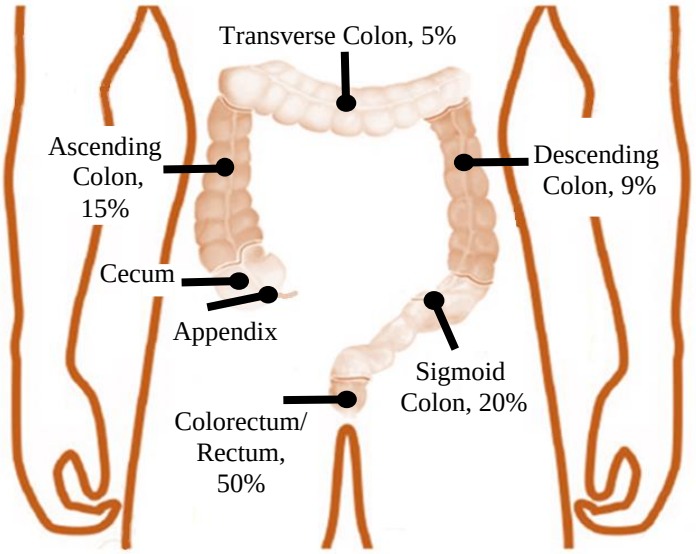


Figure 2.1: Basic anatomy of colorectal/Colorectum

parts based on distance from anal margin: rectum (3.5-7.5cm), sigmoid colon (7.5-12cm), descending colon (25 cm), transverse colon (50 cm) and ascending colon (15-20 cm) [1], as shown in figure 2.1. In figure 2.1, numbers in percentage (%) show how different part of the colon can be affected by cancer. It is shown that rectum is the part of the colon, which is the most affected, i.e., 50% [1].

2.2 Epidemiology of Colorectal Cancer

According to recent statistics, this type of cancer is diagnosed as the third most common in men (10%) worldwide after prostate cancer (20.3%) and lung cancer (17.2%). While it is the second most common cancer in female (9.4% of the total) worldwide after breast cancer (30.9%). Considering these statistics, colorectal cancer is the second leading cause of cancer death among oncological patients in both sexes [3]. This type of cancer is increasing in the past years due to several risk factors.

Nevertheless, modern medical imaging diagnostic techniques and awareness have provided positive results to control the spread of this cancer. "From 1980 until 2000-2002, five-year survival for all patients increased from 51% to 60% in northern Europe; from 52% to 62% in western European registries and from 45% to 58% in southern European registries [51]." These significant results are achieved because of improvement in surgical techniques (i.e., TME: Total Mesorectal Excision, which was first, performed in 1982 and became standard surgical therapy and currently, neoadjuvant chemo-radiotherapy is widely utilized in colorectal cancer treatments). Besides this, better medical diagnostic imaging quality has presented an accurate evaluation of pre-operative neoplasm, and this is essential for drawing up an appropriate therapeutic procedure and reducing local recurrence cancer risk [52].

2.3 Risk Factors

Nowadays, the main risk factor of getting colorectal cancer is diet intake: a diet with full of vegetables and fiber is prescribed as a protective factor by an EPIC (European Prospective Investigation into Cancer and Nutrition) in epidemiological

study, because fiber allows some important biological mechanism to fight against colorectal cancer risk [53].

The high intake of fat, red meat, and consumption of alcohol (between 30-59.9 g/day), and cigarette chain-smoking may increase the risk of getting colorectal cancer [54-56]. Also, environmental factors and lifestyle are also effective factors in developing colorectal cancer [57]. There is another significant increase in the development of colorectal cancer after rectal irradiation: patients who underwent radiation therapy have approximately a 70% changes of developing rectal cancer compared to those underwent surgery therapy [58]. A meta-analysis of 2001 shows that another important risk factor of developing this type of cancer is from inflammatory bowel disease (IBD) [59]. The rest factors are colorectal cancer etiology, genetic and hereditary factors, which have an incredibly significant role [53-59].

2.4 Pathological Anatomy of Colorectal Cancer

Colorectal carcinoma is invasive neoplasia that originates as intestinal epithelium TIS (tumor in situ) and can evolve in different morphological ways. It is demonstrated that adenomatous polyps can be a precursor of invasive carcinoma, even thug only 5-10% of them will turn in a malignant tumor. Neoplastic evolution risk is upper according to polyps' number, size polyps greater than 2 cm, villous adenoma and the presence of dysplasia and polyps hereditary syndrome index [60]. Colorectal tumors can be divided into four shape: vegetative tumor that appear as sessile formation fungoid-like jutting into the intestinal lumen; ulcerative tumor that looks like a malignant ulcer with irregular and raised edges; infiltrative tumor that usually presents a central ulcer and a scattered and extensive wall thickening and annular tumor that has an irregular surface with broad areas of necrosis [61].

According to the World Health Organization Histological Classification, we can describe five different types of rectal tumors: Epithelial tumors, Carcinoid tumors, Non-epithelial tumors, Hematopoietic and Lymphoid neoplasms and Under classified tumors. The adenocarcinoma known as epithelial tumors is a main histological type of colon cancer. 90-95% of tumors in the colon are adenocarcinomas [60].

2.5 Neoplastic Dissemination

Intestinal cancer can disseminate itself in many different ways, and directions and the dissemination depends on primary localization of the tumor where pararectal lymph nodes can be involved in the upper and middle rectum, the intermediate lymphatic drainage involves internal iliac (or hypogastric) lymph nodes and sacral group lymph nodes. In the lowest lymphatic drainage, below the dentate line, inguinal nodes and external iliac chain can be involved [62].

During the growth of rectal cancer can infiltrate small venous vessels transferring neoplastic cells in the circulation; liver and lung are the most commonly involved sites of colorectal cancer metastasis [60].

2.6 Colorectal Cancer Diagnosis

The modern technology has contributed prominently in the medical field. Due to the use of these technologies widely, cancer-causing deaths have been reduced. According to European Society of Gastrointestinal Endoscopy (ESGE) and European Society of Gastrointestinal and Abdominal Radiology (ESGAR) guidelines: CT, MRI, colonography (virtual colonoscopy), and Endorectal Ultrasound (EUS) are the recommended medical diagnostic imaging techniques for colorectal cancer diagnosis. Among these techniques, MRI is more preferable and widely explored for colorectal cancer evaluation [4-5].

2.6.1 Magnetic Resonance Imaging (MRI)

MRI, a phased-array surface coils technique, which is the main preoperative staging diagnostic imaging test for rectal cancer, particularly 3-Tesla examinations with a high accuracy level [63]. MRI has 65-86% overall prediction accuracies for T-staging. Whereas, it has better prediction accuracies in detecting larger tumors, T3 and T4, as 80-86% [64]. The development of T2-weighted fast spin-echo sequence increases the spatial and contrast resolution by obtaining a better anatomic assessment of rectal tumors details. Morphological study of rectal tumors is based on high-spatial-resolution

T2-weighted fast spin-echo sequences on the axial, sagittal and coronal planes. Moreover, images are obtained on the axial-oblique plane, perpendicular to the major tumor axis [63]. The main diagnostic features that must be analyzed for rectal cancer are: Localization/Detection; Tumor (T) staging; Nodal (N) staging; Anal complex sphincter and puborectal muscle involvement; Circumferential resection margin (CRM); Extramural vascular invasion [63].

2.7 Treatment for Colorectal Cancer

The primary goal of surgical therapy is taking control of the primitive tumor, maintenance or reconstitution of intestinal continuity with normal anal continence, preserving bladder and sexual function, minimizing morbidity and mortality associated with surgery [69]. A procedure can include:

- Complex rectal resections
- Polypectomy: during a colonoscopy where there is polyp with benign appearance.
- Local excision: it consists of a full-thickness resection with 1 cm of free margin, through endoscopic microsurgery (TEM) for proximal wounds [69].

Local excision, in the case of elective surgery, is appropriate for wound lower than 4 cm, carcinoma in situ or stage C T1 N0 M0, the absence of lymphatic and vascular invasion, a suitable distance from the anal verge [70]. The Local excision technique presents a better post-surgery, compared to resective procedures. For stage C T2 N0 M0 the local excision is avoided with resective surgery of the rectum [51]. The resection of the rectum involves adequate disease-free margins and to the removal of regional lymph node stations to the tumor site excision.

For middle and low rectum carcinoma the treatments include [70]:

- Total Mesorectal Excision (TME)
- Preservation of autonomic innervations using a nerve-sparing technique

- Protective ileostomy or colostomy packaging

The TME, Total Mesorectal Excision, consists of complete removal of mesorectum, with lymphovascular tissue, and mesorectal fascia too. Local recurrences are dramatically reduced because tumor cells in mesorectum are entirely removed in this procedure [71]. The general resective are the anterior resection of the rectum (RAR), and abdominoperineal resection (APR) called Miles operation. The RAR is a golden standard of rectal cancer surgery, and it leads a colorectal or colo-anal anastomosis with the aim to preserve the sphincter. Furthermore, the APR is performed when the anal sphincter is invaded, when its functions are compromised or when enough tumor-free distal margin is not obtained [69]. After surgery an anatomo-pathologist, in general, analyze these parameters:

- Histotype.
- Differentiation degree.
- Tumor budding.
- Resection Margin (proximal, distal, circumferential).
- Number of lymph nodes examined.
- Number of lymph nodes metastatic ones.

The surgeon removes at least 12 lymph nodes during surgery, to have reliable lymphatic staging [70].

2.7.1 Neoadjuvant Chemoradiotherapy (CRT)

The Neoadjuvant chemoradiotherapy (CRT) reduces local tumor recurrence, increases sphincters-preserving in patients with low-rectal cancer; thanks to down-staging and downsizing that show lower toxicity and greater effectiveness than adjuvant treatment **[**73**].** In the case of the locally advanced tumor (T3/T4 or any T with N+), the standard therapy is the combination of surgery (TME) and preoperative radio-chemotherapy **[**72**]**. In that way, the conventional preoperative radiotherapy technique is the three-dimensional conformal radiation therapy (3D-CRT) with a multi-leaf collimator, and the treatment unit is called Linear Accelerator (LINAC). Following the guidelines of the International Commission on Radiation Units (ICRU 62), the target volume must include all the mesorectum, the internal iliac and the

obturator lymph nodes. For a T4 tumor, are involved the external iliac lymph nodes. Usually, the preoperative radiotherapy is so-called long course radiotherapy, administered in 28 fractions. The standard preoperative chemotherapy consists of 5-Fluorouracil (225 mg/m2/die) in continuous intravenous infusion, or Capecitabine (825 mg/m2/BID) orally; both the drugs are administered for the entire RT duration. Usually, surgery is planned after 7/8 weeks of the preoperative radio-chemotherapy ending **[**72**].** These tumors when shows synchronous metastasis at the time of diagnosis should be treated using the preoperative radiotherapy, the rectal resection surgery, and hepatic metastasis resection in simultaneous operation, performing after the surgery some cycles of adjuvant therapy.

2.8 Evaluation of Neoadjuvant Therapy Response

The treatment we talked above is a combination of neoadjuvant radiochemotherapy and mesorectum total excision. The general indications for this therapy based on local recurrence risks like extramural vascular invasion, locally advanced rectum tumor (LARC) T3 and T4 with an invasion beyond muscolaris propria more than 5 mm, tumor distance from mesorectal fascia less than 1mm, lymph node involvement (N+) and sphincter proximity or involvement in lower rectum tumors [63]. A percentage of patients between 10% and 30%, at the end of the neoadjuvant treatment, have a *complete response* (pCR) characterized by the absence of neoplastic tissue. While others have a partial response and other ones are *non-responder*s [74]. Patients with complete response are considered for different therapeutic approaches as minimally invasive surgery, or clinical surveillance and; only for partial or non-responder patients are recommended for surgery. The outcome is evaluated before the surgery. During the assessment of clinical response, the nomenclature ycTN is used, where “y” means a neoadjuvant post-therapy evaluation (at the tumor and lymph node level); “c” means that are used clinical criteria [75]. The clinical criteria are clinically examined by endoscopy or other medical diagnostic imaging. Therefore, during the surgery, there is a pathological evaluation, with ypTN nomenclature where the *p* indicates the anatomopathological criterion. It can be introduced as TRG (Tumor Regression Grade), related to a classification of post-

irradiation fibrotic changes compared to residual neoplastic cells. With the aim to categorize therapy outcomes based on the number of regressive changes after the treatment, the TRG in according to Dworak can be shown as follow [76-77].

Table 2.1: The TRG system according to Dworak [76]

TRG0	No regression
TRG1	Dominant tumor with fibrosis and/or vasculopathy
TRG2	Significant fibrosis with groups of tumor cells (easy to find)
TRG3	Predominant fibrosis or mucin with very few tumor cells (challenging to find microscopically)
TRG4	No tumor cells, only fibrotic mass (total regression)

The outcome is evaluated with morphological, dimensional and intensity criteria on T2-weighted images using with the diffusion-weighted imaging technique. Furthermore, complete response to therapy can be assessed in two ways; visual inspection using multiparametric MRI and the quantitative approach by applying radiomics, which more preferable than visual inspection [9, 16-20].

CHAPTER 3

BACKGROUND ON FEATURE SELECTION AND MACHINE LEARNING ALGORITHMS

The aim of this chapter is to introduce the background of supervised machine learning algorithms and filtered based feature selection algorithms. These methods are utilized to build a prognostic model using radiomics features (as discussed in chapter 7); therefore, it is requisite to understand their background to interpret the experimental results yield by these algorithms.

3.1 Machine Learning

The term 'machine learning' generally refers to computational methods/models, which utilizes observed data (experience) for generating an accurate prediction [124, 138-139]. Machine learning approaches are programmable computational approaches based on artificial intelligence (AI). These approaches can learn the model from the

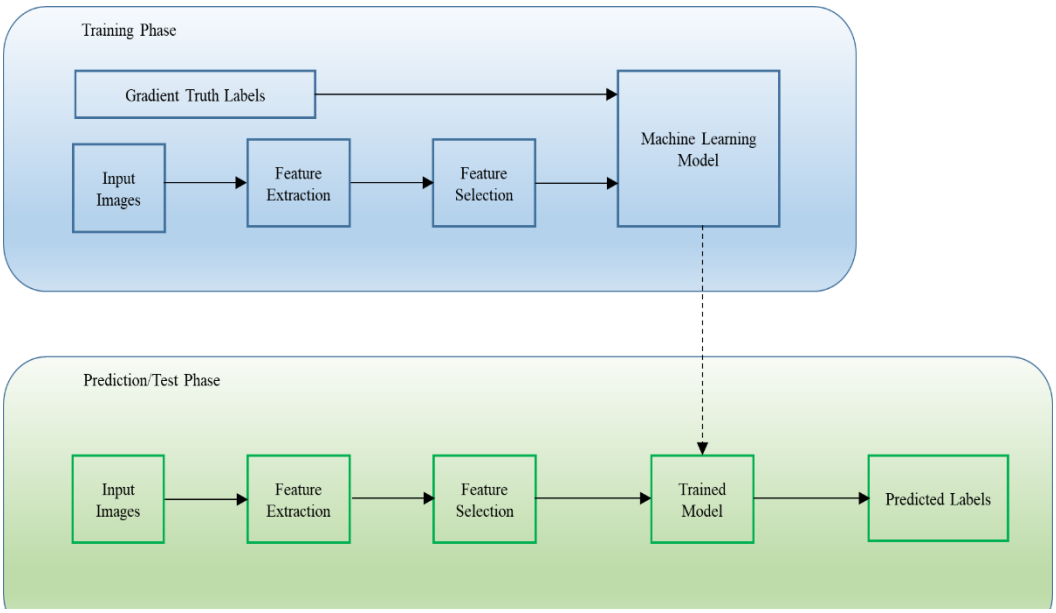


Figure 3.1: A basic block diagram of machine learning.

observed data automatically and thus can improve and automate the prediction process. In this study, we have assessed machine-learning approaches for radiomics based prediction of tumor response to neoadjuvant chemoradiotherapy (CRT) in colorectal cancer. Figure 3.1 represents a basic building block diagram of general machine learning based approach, which consists of three major blocks: feature extraction, feature selection, and training/testing model. Here, the given input is images, and then features are extracted using different statistics. Sometimes, these extracted features are in abundance as several numbers of features may increase training procedure time exponentially and increase the risk of overfitting as well. Fewer but relevant features are desirable which can reduce the computational complexity of the model and can provide a good fit. Therefore, conventional machine learning approaches incorporate feature selection methods in their operation to produce an accurate predictive model. Feature selection algorithms select particular and relevant features by discarding redundant and irrelevant features. Those selected features can give a better-fit model with high accuracy while requiring fewer features.

In medical imaging, radiomics [25-33], which decodes the encoded information within the medical images, i.e., tumor tissue heterogeneity for predicting or grading several types of cancers. More precisely, the radiomics is the method that extracts features from the medical images quantitatively, and these extracted quantitative features are analyzed to get predictive or prognostic models in personalized treatment strategies. Since high accuracy, efficiency and reliability are crucial factors in the attained predictive and prognostic models, which entirely depend on the success of radiomics based clinical biomarkers. Thus, in order to examine the effectiveness of radiomics based features in obtaining an accurate predictive model; it is necessary to validate and compare different machine learning models utilizing all possible radiomics features. Also, radiomics have high space dimensionality problem like any high-throughput data-mining field. In this regard, in [146], different feature selection algorithms are introduced for feature space dimensionality reduction. Different feature selection algorithms can improve the performance of radiomics based predictive models in different ways. Consequently, different feature selection algorithms should be assessed along with different machine learning models.

3.1.1 Machine Learning Types

There are several kinds of machine learning algorithm depending on the specific task and the characteristic of the data. There are some ML types:

- Supervised learning
- Unsupervised learning
- Semi-supervised learning
- Self-taught learning
- Reinforced learning

Supervised Learning: The supervised learning is one of the traditional ML setting [124], where a tuple of (x_i, y_i) is given as input; x_i is the input data and y_i the corresponding target vector (i.e., input labels) [128]. If the target is discrete, we have a recognition problem where the machine learns a map (function) made by finite categories doing a subsequent classification. If the problem is continuous, we have a regression problem [124]. In this thesis, we have incorporated a supervised learning based classifiers.

Unsupervised Learning: unsupervised learning presents training data that includes an example of the input vectors without any corresponding input labels [124]. In unsupervised learning, we have different objects like clustering, density estimation and visualization [124]. The term clustering refers to discover such groups or pairs of similar examples by measuring the resemblances between the examples [124]. The density estimation and visualization identifies the data distribution and transforms the data from high-dimensional space to low-dimensional space (i.e., dimensional reduction [137]).

Semi-supervised Learning: In this ML type, a considerable extent of unlabeled data is given to the algorithm for learning good feature representation of the input. It's a halfway between supervised and unsupervised learning, the training samples are parted into two portions: the data samples $X_k = \{x_1, ..., x_k\}$ with corresponding known labels $Y_k = \{y_1, ..., y_k\}$, and the data samples $X_u = \{x_{k+1}, ..., x_{l+u}\}$ where the labels are not known [127]. Here, the unlabeled data has the same distribution as the labeled data.

Self-taught learning: The self-taught learning is also a halfway but unlike the semi-supervised learning; self-taught learning assumes unlabeled data X_u that does not come from the same distribution as the labeled data X_k [131].

Reinforced Learning: The reinforced learning solves problems involving interactions with the environment and chooses the correct actions because of individual circumstances. In this case, there are no examples, but the algorithm learns by itself through a trial and error process.

3.1.2 Hyper-Parameters

Hyper-parameters refer to a selection of model with particular parameters that can enhance the performance of the machine learning algorithm. We have different hyper-parameters setting, and they are characteristic for each model. For example, there is the learning rate (continuous value) and the number of neurons (discrete value) for the Artificial Neural Network (ANN) model. The researcher gives these values through the operation called-so "fine-tuning," and sophisticated techniques obtain them, but sometimes it uses the empirical method, running different algorithms in parallel and choosing the best combination.

3.1.3 Model Selection

The principal object is choosing the values of such hyper-parameters to obtain the best generalizability in the ML algorithm [127]. For avoiding the problem of the over-fitting, the data set is parted into three parts: the training set, and validation and test set. During training of the model, different hyper-parameter settings are incorporated where these hyper-parameters are compared on the validation to select the best-trained model [124]. The test set, the last fraction, is used to evaluate the model in terms of generalization to predict the output.

Another technique is often used when there are not enough data; it is called *cross-validation*. It consists of a division of the data set into two portions obtaining the training and the test set, respectively. K-fold cross-validation is a common type of cross-validation. In this type, the data is randomly partitioned into equal sized *k*-

subsamples [124]. In *K*-fold cross-validation, *K*-1 sub-samples are employed for the training of a model while the single rest sample is used as a validation set to validate the model [124]. This process is repeated for several rounds where every subsample is used once in the evaluation process, and selection of the final model is a mean of all rounds. Another type of cross-validation is *leave-one-out*. In this type, the size of *k* is equal to the number of samples. The number of rounds in the Cross-validation increases by the factor of *k*, which consequently increases the computational complexity. The computational aspects must be taken into account, and this is one aspect of the methods for boosting hyper-parameters, such as coordinate descent, grid search or random search, multi-resolution search [122].

- Coordinate descent: we change one hyper-parameter at time
- Grid search: it works on all the possible groupings of the chosen hyper-parameters [122], with an apparent problem of increasing number of parameters. The number of possible combinations increases exponentially (for example, with 5 parameters, where each of these takes 4 values, we will have 4^5 = 1024 combinations). Fortunately, it is possible to parallelize the work making it cheaper on time and computationally efficient.
- Random search: all the parameters are changed at the same time. It includes prior distribution knowledge to sample every hyper-parameter independently [122]. Random sampling is more efficient, especially when the number of parameters starts to grow (i.e., over 3).
- Multi-resolution search: the idea is starting with a selection of the hyper-parameters with mostly sized steps. Afterward some best configurations, the steps are reduced in dimension to optimize the configuration in detail [122].

3.1.4 Under-fitting and Over-fitting

In machine learning, the object is maximizing the predictive accuracy of new unknown data. When a model fits well with the training set while giving a bad predictive accuracy on new data [130], is called the over-fitting problem. Regarding under-fitting, we observe the opposite of the over-fitting that is a function, which not

follow well the training data set giving a poor representation of the exact function [124]. Naturally, the model which neither perform well with training nor with test data sets.

3.2 Supervised Classification Methods

The following is a discussion of the primary methods of classification in ML, those we used in our study. The methods are:

- K-Nearest Neighbour (KNN)
- Logistic Regression (LR)
- Artificial Neural Networks (ANN)
- Linear Discriminant (LD)
- Support Vector Machines (SVM)
- Naïve Bayes (NB)

3.2.1 K-Nearest Neighbour (KNN)

KNN is straightforwardly easy to understand, and commonly used classification method [129]. This method solves the classification problem using Euclidean distance/similarity function by following two steps. The first one is the calculation of the distance between two instances, and the performance of the method is obtained from the *distance metric*, which is used to identify the nearest neighbors [129, 132]. The second step involves determining the class of new data points, identifying the corresponding label of the *k* nearest neighbors, which have been known in the previous step [124, 129]. Finally, instances that are apart from each other at a small distance have more chances to be grouped in the same class comparing to those who have high distance.

KNN Hyper-Parameters: The hyper-parameter for the KNN method is the selection of K value (i.e., number of nearest neighbors). By changing this value, we obtain

different dimensions for each classification region (a smaller K value produces more but smaller classification regions and vice-versa) [124]. A good choice of K is a small value because it controls the degree of smoothing. In our study, we have set K = 4.

3.2.2 Logistic Regression

The LR is called in that way for the similarity with the linear regression, but it is a method of classification, not regression, it works on the probability of outcome *Y*, given a specific feature *X*, and then $\Pr(Y = 1 \mid X)$. Given a value *X*, it could predict if a given Y-class label is positive or negative in binary classification. For instance, any data point where it is verified $\Pr(X) \geq 0.5$, *Y* is classified as positive. The probability is estimated as follow, with the *logistic function* $\Pr(X)$, where β_0 and β_1 are the parameters of the model.

$$\Pr(X) = \frac{e^{(\beta_o + \beta_1 X)}}{1 + e^{(\beta_o + \beta_1 X)}} \tag{3.1}$$

Logistic Regression Hyper-Parameters: Manipulating with the logarithm the previous equation we arrive at the following relation

$$\log\left(\frac{\Pr(X)}{1 - \Pr(X)}\right) = \beta_0 + \beta_1 X \tag{3.2}$$

The values β_0 and β_1 can be estimated by the technique called *maximum likelihood* [133]. The maximum likelihood tries to find β_0 and β_1 so that probability $\Pr(X)$ approaches as much as possible to 1 for all data points ending in the positive class. As for the data points of the negative classes $\Pr(X)$ must be close to 0. This way is mathematically formalized by maximum likelihood function as follow:

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} \Pr(x_i) \prod_{i:y_i'=1} 1 - \Pr(x_i') \tag{3.3}$$

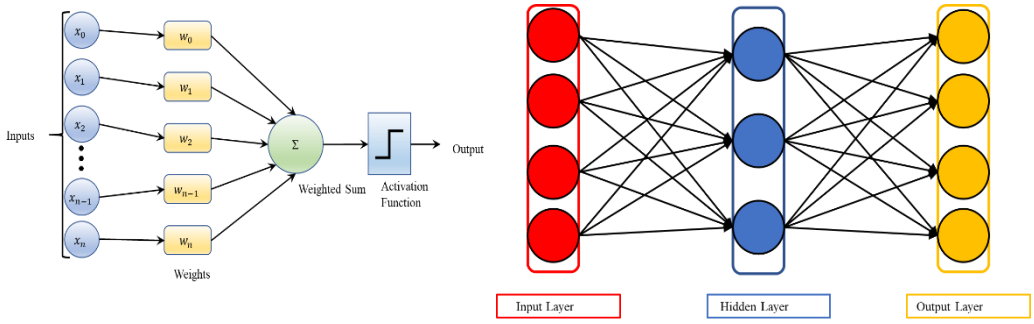
3.2.3 Artificial Neural Networks (ANNs)

It is called the ANNs because initially, it was an attempt to represent mathematically the information processed in the nerve cells, the neurons. The neuron is the basic unit with inputs and outputs, and each neuron is connected through a synapse, which controls the gain of the signal from each source. In the ANNs design the neuron, that mimics the behavior is a *perceptron* that takes several inputs with their associated weights w_i, if the input weight exceeds a certain defined threshold, a particular output signal is activated. Mathematically speaking, the perceptron can be defined as the following equation:

$$y = \psi\left\{\sum_{i=1}^{n} w_i x_i + b\right\} \tag{3.4}$$

Here, y is an output signal, ψ is the activation function, n the number of the connection of perceptron, x_i represents the values of the *i-th* connections, and b the threshold (or bias).

The strength of this method is given when more perceptrons are combined to work together forming an ANN. To form a network, we organize the perceptrons in layers where each of it takes inputs from the previous one, and applies weights and



(a) (b)

Figure 3.2: This block diagram represents the Perceptron model. (a) Every activity is multiplied by a weight and passes through a weighted sum and finally is computed using an activation function, and (b) ANN consists of three layers that are fully connected.

passes the signal to the next layer appropriately, as shown in figure 3.2. By incorporating hidden layer/s, we reduce the limitation associated with the use of a single perceptron to learn [134]. The training of an ANN occurs with the adjustment in each node of the weights so that the error between the desired output and the current one is reduced (but not eliminated) and this process requires the ANN to compute the error derivative of the weights [135]. In other words, there is an evaluation of the error variation when each weight changes its value. The random initialization of weights is a way to do that, feeding the network with examples; the error made by the network at the output is calculated and is used backward in the process of back-propagation. Repeating this process many times, the ANN can distinguish between different classes. *ANNs hyper-parameters*: We use backpropagation ANNs with five hidden considering 900 training epochs.

3.2.4 Linear Discriminant (LD)

Linear Discriminant (LD) is one of the machine-learning approaches that divide the data points into classes and categories [120]. LD applies statistical properties on the data to differentiate the data into classes. Assuming the conditional PDF (probability density function), $p(x \mid y=0)$ and $p(x \mid y=1)$ are normally distributed with mean and covariance parameters are (μ_0, ε_0) and (μ_1, ε_1), respectively. LD based prediction model utilizes Bayes theorem where the probability of second class is estimated by comparing the log of likelihood ratio with certain of threshold *T* as follow:

$$(x-\mu_0)^T \varepsilon_0^{-1}(x-\mu_0)+\ln|\varepsilon_0|-(x-\mu_1)^T \varepsilon_1^{-1}(x-\mu_1)-\ln|\varepsilon_1|>T \qquad (3.5)$$

Here, each class utilizes its own mean and covariance, and makes the classifier as quadratic discriminant analysis and using the assumption of homoscedasticity, we obtain:

$$\begin{aligned} x^T \varepsilon_0^{-1} x &= x^T \varepsilon_1^{-1} x \\ x^T \varepsilon_i^{-1} \mu_i &= \mu_i^T \varepsilon_i^{-1} x \end{aligned} \quad \text{with } \varepsilon_i \text{ Hermitian} \qquad (3.6)$$

We obtain from this criterion a threshold $w \cdot x > c$.

For some threshold constant C we, will have;

$$w = \varepsilon_1^{-1}(\mu_1 - \mu_0)$$
$$C = \frac{1}{2}(T - \mu_0^T \varepsilon_0^{-1} \mu_0 + \mu_1^T \varepsilon_1^{-1} \mu_1) \quad (3.7)$$

In other words, the criterion of the input *x* being in a class *y* depending on a linear combination of the known observations. LD has a closed form solution and therefore no hyper-parameters. The solution can be obtained by an empirical sample of class covariance matrix [121].

Figure 3.3: This figure shows a one-dimensional hyperplane $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$ and every region is identified observing how the hyperplane divides the space. In other words evaluating $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \leq 0$ and $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \geq 0$

3.2.5 Support Vector Machine (SVM)

The SVM is one of the classification approaches presented in the '90s by Boser et al. [125], and it presented the appreciable performance in solving problems related to classification and regression [122]. In contrast to ANNs, the SVM involves a problem of optimization of a convex function where each single local solution is considered as a global solution [124]. To understand the SVM technique, it is essential for introducing some concepts such as *hyperplanes*, *maximal margin classifiers*, and *support vector classifier*. A hyperplane is defined as a two-dimensional subspace of dimension *p*-1 in a *p*-dimensional space and is defined by a simple equation:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0. \quad (3.8)$$

A hyperplane is useful to identify the class of the data points depending on their location corresponding to the hyperplane and the regions it subdivides, as shown in

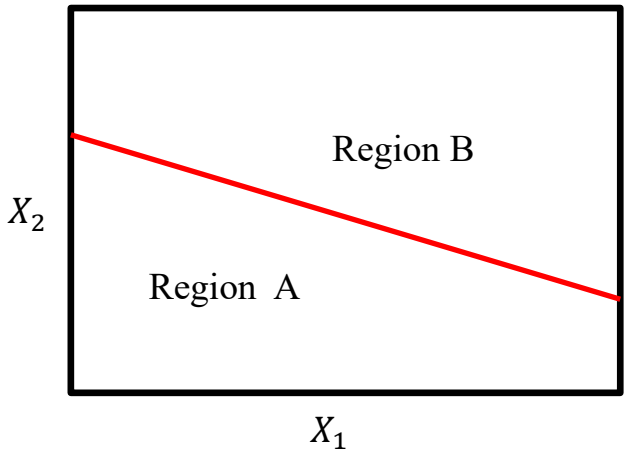


figure 3.3. To decide which hyperplane should be chosen, the *maximal margin classifier* is used in the decision process. The margin refers to the minimum distance between the hyperplane and some samples, where the closer data points from the hyperplane appear, known as *support vectors* [124]. The maximum margin is obtained from a problem of optimization as follow:

$$minimize \ \frac{1}{2}\|w\|^2 \ subject \ y_i(w^T x_i + b) \geq 1 \quad \forall i \ , \tag{3.9}$$

Minimize $\frac{1}{2}\| w \|^2$, maximize the margin (given by $\frac{2}{\| w \|}$), and a constraint for each single data point ensure that all the data points are correctly classified [124]. Using the Lagrange theory:

$$maximize \ \ W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \ \ subject \ to \ \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \ , \tag{3.10}$$

where α_i is the Lagrange multiplier for every single constraint, in such a way, we have a new object function co α_i ncerning [123]. We have a twofold problem because finding a value of α_i is a finding the value for *w*, and vice-versa. The decision function for a new *c* input is written as $sign\left[\sum_{i=1}^{n} \alpha_i y_i (x_i^T c) + b\right]$. If all values of α_i are zero then *w* is a linear combination of insufficient data points, and where α_i is not zero we obtain the support vectors those find the choice of the hyperplane [124]. The support vectors are the bounds and a function of the training data misclassification error and the complexity

related to the capacity of the model [123]. Until now, we have assumed the possibility to separate two classes correctly in a linear way, when it is impossible it is used the *soft margin* technique. With a soft margin, we lessen the effect of specific data points to let some of the training points to be misclassified (positioned on the wrong side of the hyperplane) [123]. We introduce *slack variables* $\xi_i \geq 0$ with $\xi_i \in \{\xi_1, \ldots, \xi_n\}$, with one slack variable for every single data point [124]. The underlying problem is transformed as follow:

$$\textit{minimize} \quad \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{n}\xi_i \ \textit{subject to}\ y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i \,, \tag{3.11}$$

The term C > 0 controls the trade-off between the necessities of a lower error and maximizes the margin obtaining less stringent constraint [124]. For $\xi_i = 0$ we have the point data inside the right margin, or on it. With $0 < \xi_i \leq 1$ data point is inside the margin where it is on the correct region of the hyperplane, and it is classified correctly on the decision boundary. If $\xi_i > 1$, the data is misclassified [124]. Considering a linear increase for the penalty of misclassification with ξ and $\sum_{i=1}^{n}\xi_i$ is an upper bound on the number of misclassified points. We can formulate the primary problem as a twofold representation as follow:

$$\textit{maximize}\ W(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j (x_i^T x_j)\,,$$

$$\textit{subject to} \quad C \geq \alpha_i \geq 0, \quad \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{3.12}$$

Differently from the previous case, here, we observe the upper bound on α_i [123] in a convex quadratic programming work where an optimum value for *C* cannot be found experimentally using a linear boundary. The idea is transforming the data points from the original input-space function into a high-dimensional feature-space; where it is possible to divide them using a linear function. The linear function in the new space is nonlinear in the input space [124]. We have a transformation that can be defined as

$\theta(x): R^n \to R^{n'}$, where $n << n'$ [123]. We can rearrange the dual optimization problem as follow:

$$\textit{maximize}\ \ W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \theta(x_i)^T \theta(x_j),$$

$$\textit{subject to}\ \ C \geq \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \tag{3.13}$$

Computing $\theta(x_i)$ is very costly, and the *kernel* is used to avoid the inner product. The *formula gives the kernel function*:

$$K(x_i, y_j) = \theta(x_i)^T \theta(x_j) \tag{3.14}$$

where we can observe the likeness extent between arguments [123].

The kernel uses the inner product between the mapped data points, which is assessed without the explicit mapping function. Using the kernel function in the previous twofold problem of SVM in training [124] :

$$\textit{Maximize}\ W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j),$$

$$\textit{subject to}\ \ C \geq \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \tag{3.15}$$

This "*kernel-trick"* is employed to avoid computing $\theta(x)$ explicitly. New points *z* are now classified as follow:

$$sign\left[\sum_{i=1}^{N_s} \alpha_i y_i \theta(s_i)^T \theta(z) + b\right] = sign\left[\sum_{i=1}^{N_s} \alpha_i y_i K(s_i, z) + b\right], \tag{3.16}$$

with s_i support vectors and N_s is the number of support vectors [124]. We have more than one kernel but for the practical seek we introduce two of them, and they are the *Polynomial of degree 'p'* and *Radial Basis Function* (*RBF*).

- *Polynomial of degree 'p'* $K(a,b)=(a\cdot b+1)^p$
- *Radial Basis Function* (*RBF*) $K(a,b)=e^{-\|a-b\|^2}/2\sigma^2$

In the polynomial kernel we need to determine *p*, and for RBF-Kernel we need to select an appropriate σ. Changing kernels, we obtain different non-linear classifiers with the same algorithm, but on other hands, the selection of the specific kernel function limits the transformation type to be applied on the data [123].

With the classification, we can use the SVM technique for regression tasks too. The $\in$-*insensitive error* is utilized to penalize the data points evaluating if the difference between the prediction function and the target is greater than $\in$ [124]. Assuming the presence of noise, constraints like $y_i - w*x_i + b \leq \in$ and $y_i + w*x_i + b \leq \in$ are well-defined letting a deviation ϵ from the predictable function [123]. We can visualize the constraints like a regression tube with size 2ϵ. As in the previous case, we introduce slack variables $\xi_i \geq 0$ and $\hat{\xi}_i > 1$ for every type of error, two of them in this case. Similar than before if $\xi_i = 0 = \hat{\xi}_i$ the point lies inside the tube, the sample above the tube has obviously $\xi_i > 0$ and $\hat{\xi}_i > 0$ [124]. In this case, for a linear $\epsilon-$ *insensitive error loss function* we have the following problem:

$$minimize\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\left(\xi_i + \hat{\xi}_i\right),$$

$$Subject \begin{cases} (w\cdot x_i - b - y_i) + \xi_i \geq \in \\ (w\cdot x_i - b - y_i) - \xi_i \leq \in \\ \xi_i, \hat{\xi} \geq 0 \end{cases} \qquad (3.17)$$

The term $\frac{1}{2}\|w\|^2$ is the regularization term to control the complexity of the regression algorithm, and *C* controls the trade-off between the number of errors and the model complexity, which are tolerated (for instance if a higher amount of errors leads with a more complex model) [123, 128]. Another term of control is ϵ, that term controls the width of the regression-tube where bigger is this value, lower is the number of data

points in the tube giving a more straightforward function [128]. We can use the kernel technique on the regression model if it presents a non-linear function with the same strategy as in the section above.

SVM Hyper-Parameters: SVM hyper-parameters are selected subject to the use of kernel-function, except the term C, different terms appear:

- C: C > 0 controls the trade-off maximizing the margin and minimizing the error due to misclassified points. If C is too large, we have a narrow and complicated margin so that over-fitting can occur for this term, with a large C [124, 126].
- The RBF, *Radial Basis Function* Kernel is controlled choosing the hyper-parameter γ. Choosing γ, we choose the complexity of the margin; for example, a smaller γ gives a more straightforward margin. A higher complicate margin [126].
- In the Polynomial Kernel, the degree of polynomial *P* is a hyper-parameter [126].
- When the Linear Kernel $K = (a, b) = a^T \cdot b$ is used, it is not necessary for selecting kernel/parameters [123, 126].

In our study, we used two types of the kernel in the SVM algorithm, linear (we named as, LSVM), and polynomial (we named as QSVM) with $P = 2$. We set the $C = 0.5$ for both kernels.

3.2.6 Naïve Bayes (NB)

The probabilistic nature of Bayesian classifiers is useful for implementation in computer decision support settings. This classifier works by merely assigning each observation to the most likely class *j*, given a particular feature x_0, as per it is conditional probability[133, 136]. Conditional probability is the probability that $Y = j$ given x_0 as per:

$$\Pr(Y = j \mid X = x_0) \tag{3.18}$$

It works choosing the class which conditional probability is largest, and the *Bayes error rate* represents the probability of misclassification on the instance as follow :

$$1 - \max_j \Pr(Y = j \mid X = x_0) \tag{3.19}$$

Classification is done by values of multiple features where all features are assumed to be conditionally independent given the class label. Even though this approach usually false, but the result works very well. Using probability distribution, we obtain the characterization of confidence in classifier prediction, allowing prediction with low confidence to be rejected and passed to a human.

3.3 Feature Selection Algorithms

Feature selection refers to an automatic selection of the most relevant variables or attributes while removing undesired irrelevant variables from data, which aids to create an accurate and more straightforward predictive model with those selected fewer variables [138-139].

Generally, there are three types of feature selection methods, including filter methods, wrapper methods, and embedded methods. The last two methods, wrapper, and embedded methods are classifier dependent approaches, whereas, the first ones, filter methods, are classifier independent.

Wrapper methods are the search methods, which search through the whole feature space and identify a relevant and non-redundant feature subset where several combinations are assessed, and compared to other combinations [140]. These methods are computationally expensive methods and may generate feature subsets those are overly particular to the classifiers and hence has low generalizability.

Embedded methods include feature selection as a part of the training process because these methods can learn the best attribute/variable, which enhances model accuracy while the model is being produced. They are computationally efficient as compared to the wrappers. However, they still use a quite strict model (classifier) structure assumption and hence lacks in the generalizability.

Filters are computationally efficient, and they have high generalizability and scalability, and these are some reasons why they are often used. These methods are classifier-independent methods, and they are the simple feature ranking methods based on some heuristic scoring criterion. Filters based methods employ simple statistics to describe a scoring to every attribute. The defining component of filter-based feature selection methods is the scoring/selection and ranking the scored features criterion, which is often known as 'relevance index.'

Filters methods are further categories into two types: univariate and multivariate methods. The univariate filters methods includes scoring criterion based on considering attribute relevancy and eliminating attribute redundancy (some examples of these methods are Fisher score, Relief, T-test, Chi-square, and Mutual information maximization), whereas multivariate methods (for instance Minimum redundancy maximum relevance) examine the multivariate relations within attributes, and the scoring criterion is weighted average of attribute redundancy and relevancy. The attribute relevancy is an association of the attribute with the target variable, while the attribute redundancy is the extent of redundancy existing in a specific attribute concerning the set of already selected attributes [140]. This feature selection problem could be defined, as done by Brown et al. [141]. Where J be the scoring criterion (relevance index), Y is the class labels, X be the set of all features, X_k be the feature to be assessed, and S be the set of already selected features.

In the following paragraphs, we will concentrate in the description of some feature selection methods mentioned before.

3.3.1 Fisher score

Fisher score [142], filter-based feature selection method, which selects features such that the distance between the two classes is maximized and the within-class distance is minimized. The scoring criterion is defined as,

$$J_{Fisher}(X_k) = \frac{\sum_{m=1}^{2} n_m (\mu_{k,m} - \mu_k)^2}{\sum_{m=1}^{2} n_m \sigma^2{}_{k,m}}, \tag{3.20}$$

where μ_k is the overall mean of the feature X_k, n_m is the number of samples in *m*-th class, and $\mu_{k,m}$ and $\sigma^2_{k,m}$ is the mean and variance of the feature X_k on *m*-th class.

3.3.2 Relief

Relief [143], assumes *p* randomly sampled data instances and defines the scoring criterion as,

$$J_{Relief}\left(X_k\right)=\frac{1}{2}\sum_{t=1}^{p} d\left(X_{t,k}-X_{NM(x_t),k}\right)-d\left(X_{t,k}-X_{NH(x_t),k}\right) \tag{3.21}$$

where $X_{t,k}$ is the value of instance x_t on features X_k, $X_{NM(x_t),k}$ and $X_{NH(x_t),k}$ are the values on the *k*-th feature of the nearest point to x_t with the same and different class label respectively, and $d(.)$ denotes the distance.

3.3.3 T-test (T_Score)

T-test based feature selection evaluates a feature using a t-score, which is defined as,

$$J_{ttest}\left(X_k\right) = \frac{\mu_1-\mu_2}{\sqrt{\frac{\sigma_1^2-\sigma_1^2}{n_1-n_2}}}, \tag{3.22}$$

where μ_1, μ_2 and σ_1^2, σ_2^2 are the means and variances of the two classes on the feature X_k, whereas n_1 and n_2 correspond to the cardinality of the two classes.

3.3.4 Chi-square (Chi_Score)

Chi-square score for a feature with *r* different values is defined as,

$$J_{Chi-square}\left(X_k\right)=\sum_{i=1}^{r}\sum_{m=1}^{2}\frac{\left(n_{im}-\mu_{im}\right)^2}{\mu_{im}}, \tag{3.23}$$

where n_{im} is the number of samples with *i*-th feature value in *m*-th class and $\mu_{im} = \frac{n_{*m} n_{i*}}{N}$

Here, n_{i*} is the number of samples with *i*-th feature value, n_{*m} is the number of samples in class *m* and *N* is the number of samples.

3.3.5 Mutual information maximization (MIM)

Mutual information maximization [144] uses information theory to measure the relevance of a feature. The scoring criterion is defined as mutual information between a feature and a class label. It is given as,

$$J_{\min}(X_k) = I(X_k; Y). \tag{3.24}$$

3.3.6 Minimum redundancy Maximum relevance (MrMr)

Minimum redundancy maximum relevance (MrMr) [145], tries to evaluate feature using relevancy-redundancy trade-off. Here the configurable parameter β is set as the cardinality of the set of selected features. Hence, the scoring criterion is defined as,

$$J_{mrmr}(X_k) = I\left(X_k; Y\right) - \frac{1}{|S|} \sum_{X_j \in S} I\left(X_k; X_j\right). \tag{3.25}$$

CHAPTER 4

BACKGROUND ON DEEP LEARNING

The primary goal of this chapter is to understand the basic building blocks of deep learning. It is essential to get fundamental knowledge of a basic working principle of the deep learning, which subsequently helps in understanding the deep learning-based segmentation algorithms described in Chapter 5, and their experimental results explained in Chapter 6. The main architectures of deep neural networks are discussed in Section 4.3 followed by their sub-sections. The primary motivation of using deep learning is mentioned in Section 4.2, before this, a fundamental concept of traditional feed-forward ANNs is given in Section 4.1. Network training is explained in Section 4.5 and, as regularization improves the performance of the network training, explained in Section 4.4.

4.1 Basic Concept of Feed-Forward Artificial Neural Networks (ANNs)

Artificial neural networks (ANNs) also known as neural networks (NNs), which are computing systems imprecisely inspired by the biological neural system of the human brain [78]. Neural networks consist of single neurons unit, which creates the neural networks by their connections with each other [79]. The fundamental concept of ANNs is to extract linear relationship in derived input features, and then the target is modeled by mapping these derived input features by non-linear activation function [80]. Figure 4.1 illustrates the basic structure of a single neuron. Here, x_1, x_2, and x_3 represent *inputs*, +1 denotes *bias* (b), and w_1, w_2, and w_3 indicate *weights*. Finally, $H(x)$ is an *output*, which is obtained by the sum of weighted inputs transferred through activation function, and mathematically can be expressed as:

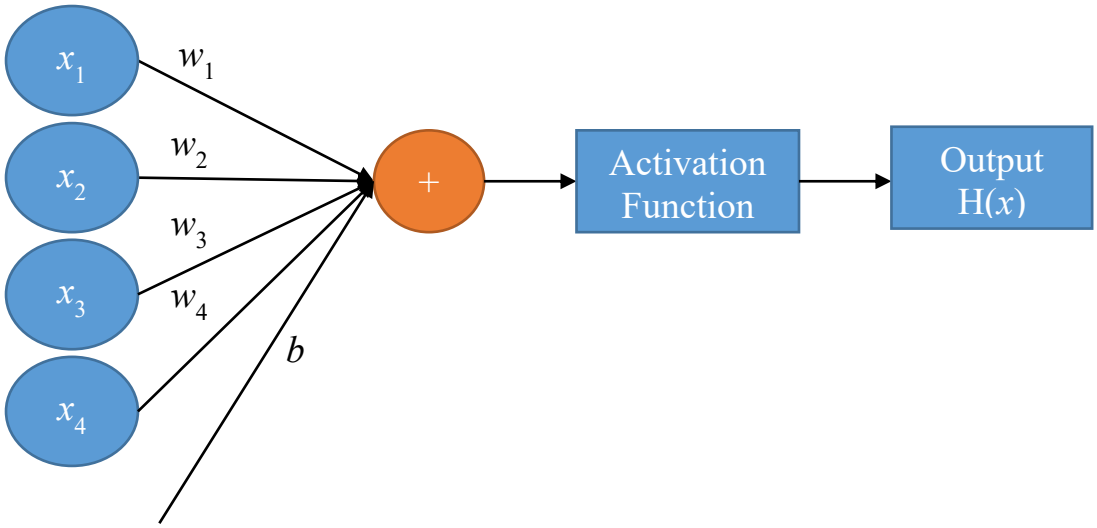


Figure 4.1: An illustration of a unit neuron with four inputs.

$$H(x) = g\left(\sum_{i=1}^{4} x_i w_i + b\right) \tag{4.1}$$

where g(·) represents a nonlinear activation function. Traditionally, neural networks were modeled using hyperbolic tangent function *Tanh* and *sigmoid* as non-linear activation functions but recently rectified linear unit (*ReLU*) function is more efficient computationally [81]. These activation functions are described in section 4.3.4.

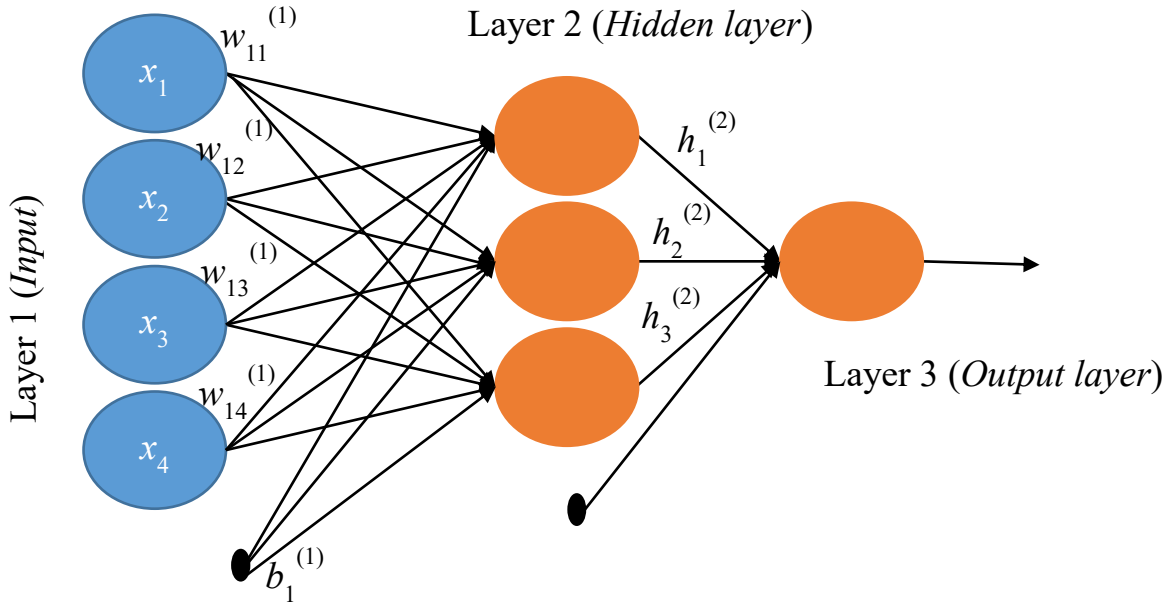


Figure 4.2: An example of feed-forward or multilayer perceptron

According to equation (4.1), each x_i is weighted by multiplying its value with every w_i, individually. After that, the weighted inputs are summed up along with *bias* (i.e., *offset*). The final output is obtained by transferring their linear combination through the nonlinear activation function g(·) [79].

Generally, ANNs consist of tri chronological layers as shown in Figure 4.2, where layer 1 and 3 are input and output layers, and middle layer 2 is a hidden layer, respectively. There are many hidden layers can be integrated depending on complexity in decisions, where the previous layer is fully connected to the next one. In other words, each neuron in every previous layer is connected with each neuron of the next layer as shown in figure 4.2. These types of networks are known as feed-forward neural networks or also known as multilayer perceptron, in which inputs are always passed in the forward direction without any closed directed loop [82]. Furthermore, the network requires a backpropagation process which adjusts and updates the weights w_i in the training phase to learn a general rule from the given training data. In section 4.5.1, backpropagation process is explained.

4.2 Deep Learning

Deep learning [83] has recently emerged as a successful and popular methodology in computer vision, speech recognition and natural language processing. Deep learning is also known as deep convolutional neural networks; abbreviated as CNNs or ConvNets. CNNs have successfully attained many breakthroughs and state-of-the-art performance in various computer vision applications, including image classification and recognition [84-85], semantic segmentation [86-88], stereo matching [89], and object detection [90-91]. Convolutional neural networks (CNNs) have a very close resemblance to traditional neural networks (NNs). Similarly, CNNs are made-up of a group of neurons to have learnable weights and biases. As ordinary NNs performs dot product operation as discussed in section 4.1 and from equation (4.1). Similarly, CNNs utilizes a linear operation namely, convolution (where inputs are convolved with weights of different kernel sizes) followed by a nonlinear activation function. Besides, they also contain loss function such as cross-entropy based Softmax loss or SVM Hinge loss on the last layer (i.e., fully connected (FC) layer). Thus, all the approaches that are

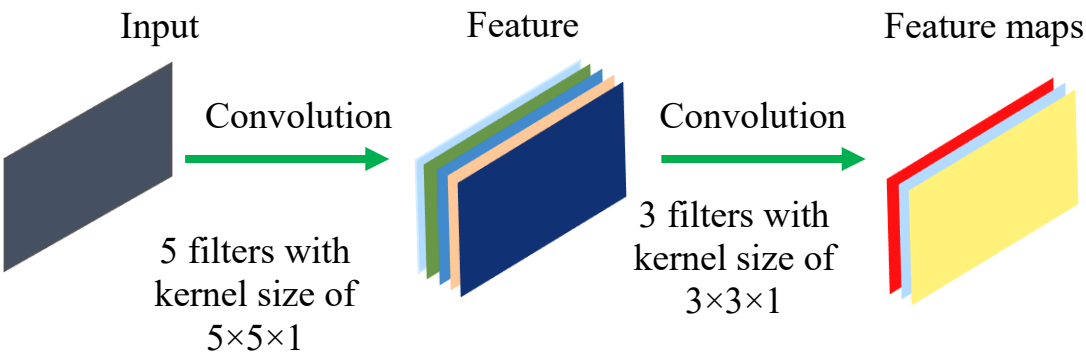


Figure 4.3: An example of convolutional networks by applying different size of filters but smaller than the input size. Each thin cuboid represents one feature channel that is convolution output obtained by applying one filter on the preceding input.

developed for learning regular Neural Networks parameters still apply to the CNNs [92]. However, both networks contain the same properties but CNNs still expresses a single differentiable score function: as CNNs takes input as raw images and processes the input on one end to class scores at the other.

The main question arises that if these both networks have similar properties where every developed method have been learned for Neural Networks can be utilized in CNNs, then why there exists a need to learn another type of network like CNNs? An explicit assumption made by CNNs, which allow us to incorporate certain modifications into the network architecture; making the network computationally faster and reduces the number of parameters.

In short, CNNs takes advantage of three crucial considerations those may help to improve a machine learning system, such as 1) sparse interactions, 2) parameter sharing, and 3) equivariant representations [83]. These three properties make CNNs more efficient, to be able to reduce the number of parameters in the whole network dramatically. Moreover, convolution offers several ways to deal with inputs of variable size.

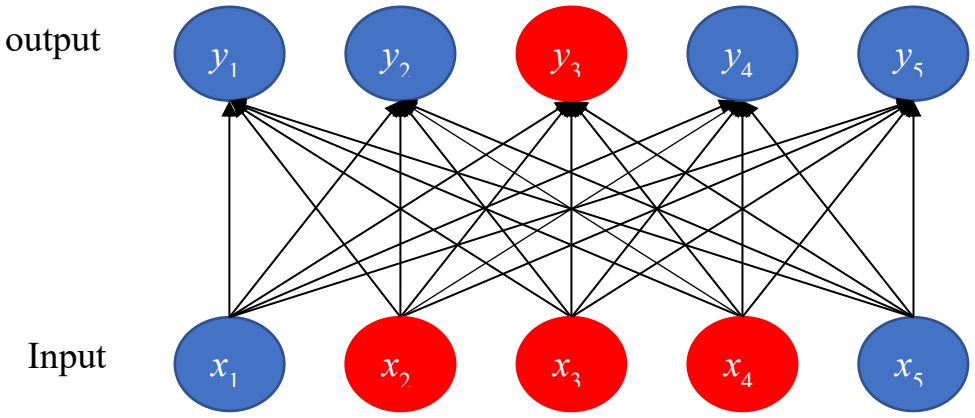


Figure 4.4: An example of ordinary NNs or fully connected neural networks. The top row (i.e., output) is molded by matrix multiplication with full connectivity. The black arrows point out the effect of input units over the output units, and it demonstrates that all units along with red circles in the bottom row affect the output y3. The figure is redrawn from [83].

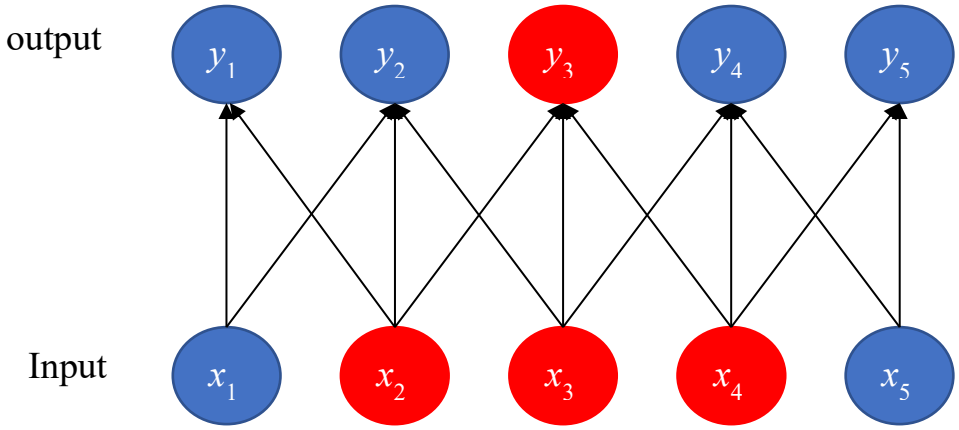


Figure 4.5: An example of convolution operations. The top row referred to the output which is achieved by applying a filter of kernel size 3 to the input, i.e. bottom row. The black arrows point out the effect of input units over the output units. The output y3 is affected by red circles in the bottom row and are called as the receptive field of the output y3 whereas blue circles in the bottom row indicate no effect on the output units. The figure is redrawn from [83].

In conventional neural network layers where every output unit interacts with every input unit, as shown in Figure 4.2. Whereas, CNNs provides sparse interactions

between output and input units. This sparse connectivity is achieved by incorporating different kernel sizes that should be smaller than the input size, as shown in Figure 4.3. By Considering an example of image processing where images are high dimensional vectors, and each image might have thousands or millions of pixels which require a massive amount of parameters to characterize the network. In this scenario, if ordinary NNs receive 256 × 256 images as input then each neuron in the hidden layers will have 64K connections; this lead the network to have a significant number of weights, and it also requires much memory.

On the contrary, CNNs uses a kernel that occupies the small and restricted area to extract meaningful features, like edges features, etc. This strategy in CNNs leverage several benefits over ordinary NNs. First, fewer parameters are required to store; in this way memory requirement of the model is dramatically reduced. An illustration is depicted in figures 3.4 and 3.5 where a comparative difference between CNNs and ordinary NNs in terms of connectivity, required parameters and the receptive field difference of one unit in each network is shown respectively. Secondly, it also demonstrates that output computation typically requires fewer numerical computations, which may often be quite significant in real applications.

Moreover, CNNs offers parameter sharing where the same value of the weight applied to one input is tied to the value of weight applied elsewhere. In contrast to this, the traditional neural network uses each element of the weight matrix exactly only once to compute the output of one layer. CNNs employ each associate kernel at every pixel of the input image. However, the kernel may not be applied at pixel around the boundary of the input image, but there are some CNNs architecture designs, which incorporate ‘zero padding' (explained in section 4.3.1), can deal with pixels around the boundary. This parameter sharing property in CNNs ensures that the network learns only one set of parameters shared by all location rather than learning a set of parameters for each location, separately. In this way, the memory storage requirements of the model are further reduced. Hence, the convolution operation is immensely more efficient than dense matrix multiplication in terms of memory requirements and the number of parameters to be learned.

Furthermore, by adopting this parameter sharing causes the convolution layer to maintain equivariance property to translation. For example, convolution layer generates

a 2D features map for the 2D input image. If the object in the input is moved with a particular amount or direction, then the same amount in the output will be moved by the feature representation of the input.

4.3 Convolutional Neural Networks (CNNs) Architecture

Recall from Section 4.2, CNNs have advantages to extracting information from the raw input images, and they constrain the architecture in a more functional way than traditional NNs. Originally, a CNNs architecture was designed for 2D input data (i.e., 2D color image with three red (R), green (G), and blue (B) channels). Therefore, traditional CNNs are 2D in nature. In this Section, we will enlighten the basic structure of 2D CNNs, and then we will discuss their modification to 3D in chapter 4. Typically, CNN was designed for image classification, where predicted output was considered as a single class label as shown in figure 4.6. Here network takes an input image and predicts image class; it is either dog, cat or bird. CNN can also be used for detection or segmentation purposes, where the predicted output contains object location information. More simply, a single class label is assigned to every pixel associated with the desired object in the output image. Figure 4.7 shows an example of semantic segmentation where a bird is a given input and pixels related to the bird is associated with the predicted class label, and these pixels are marked with green while unrelated pixels are colored with black.

Usually, a CNN comprises the number of convolution layers, alternated between many pooling layers. A typical CNN contains a set of filters or kernels, which are convolved with input to generate a number of feature maps equal to a number of filters. After that, the nonlinear activation (i.e., ReLU) function is applied to the generated feature maps to introduce nonlinearity in them. Some CNN integrate batch normalization (BN) before applying a non-linear activation function that makes training easy for deep networks.

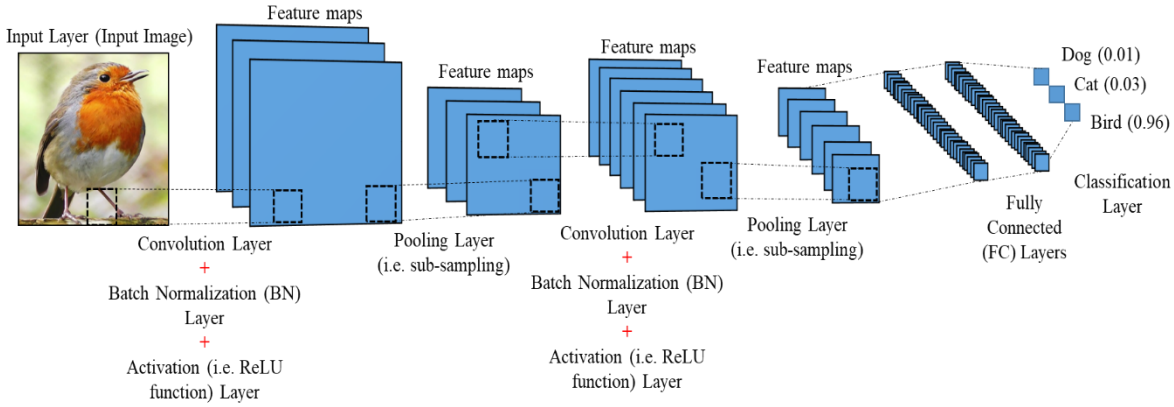


Figure 4.7: An example of a convolutional neural network's architecture for image classification.

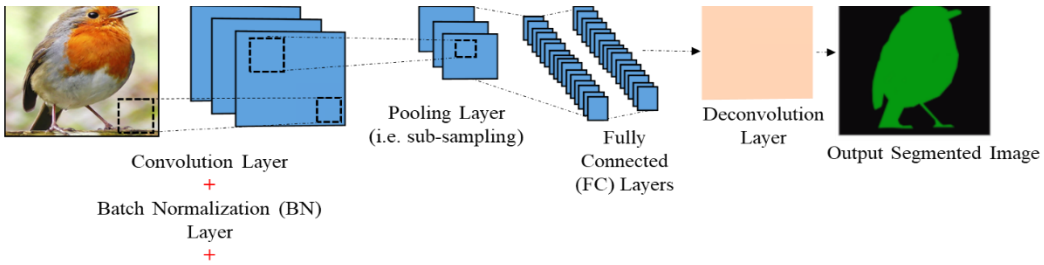


Figure 4.6: An example of a convolutional neural network's architecture for semantic segmentation.

Figure 4.6 shows a traditional CNNs architecture, which is comprised of multiple layers defining several operations. Each of which is elucidated in the following subsections: Input layer in Section 4.3.1, convolution layers in Section 4.3.2, pooling layers or subsampling layer in Section 4.3.3, activation layers in Section 4.3.4, batch normalization (BN) layers in Section 4.3.5, and fully connected layers in Section 4.3.6. Additionally, deconvolution layer is described in Section 4.3.7, which is used in segmentation to up-sample the sub-sampled features by pooling layer in the network. Finally, the classification layer, which produce output in class probabilities, is explained in Section 4.3.8.

4.3.1 Input Layer

The input layer is actually not considered as a network layer, which is not used for learning or training purpose. It is a visible layer, which generalizes the input data, like pre-processing for normalizing the data or converting the data into the required format.

4.3.2 Convolution Layer

A mathematical representation of images are in matrices containing color information in the form of RGB color codes (i.e., natural images), and intensity texture information, etc., (i.e., medical images). Therefore, an image has volumetric size $h \times w \times d$, where color channel depth $d = 3$. Whereas, in medical images like computer tomography (CT) or magnetic resonance imaging (MRI), the depth d is corresponding to the number of channels. Convolutional layers are fundamental layers in CNNs. Generally, a convolutional layer contains a set of kernels or filters. These kernels are convolved with the given input to extract features and generate one feature map corresponding to every kernel. Let K be the kernel with size $K_x \times K_y \times d$; x and y represent rows (i.e., height) and columns (i.e., width) of K, and d is its depth. The height and width of the kernel are smaller than image height and width. The kernel with its given height and width convolves with the whole input as shown in figure 4.6 and 3.7 (i.e., dotted line circles correspond to the kernel, which slides over the input) and their numerical illustration is shown in figure 4.8. Besides, the size of the kernel corresponds to a receptive field on the input as shown in figure 4.8; therefore one may call kernel as the receptive field on the input.

Typically, CNNs performs convolution operation as the sum of the element-wise multiplication of the kernel and the input image (from the first layer) or input feature maps (from second and onward layers). While creating a convolution layer, one can modify the number and size of kernels, the size of the stride, and the presence or absence of padding.

Stride[1]: The term stride in convolution layer is referred to a number of pixels by which the kernel shifts at a time. In the CNNs, the output feature maps produced by each convolutional layer is consequently decreased in every next layer depending on the size of the kernel and stride. Equation (4.2), illustrates the relationship between

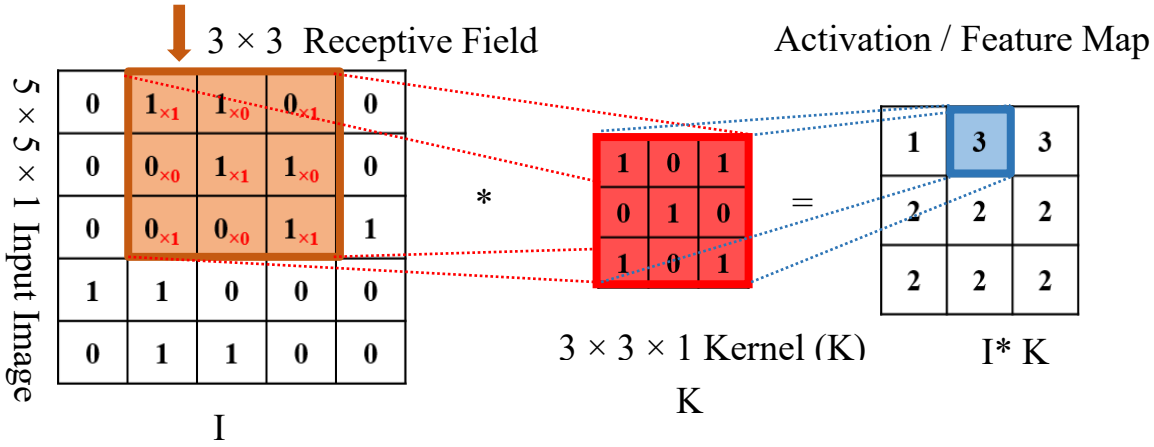


Figure 4.8: An illustrative example with one color channel, i.e. gray level image. This example shows how a convolution operation is performed on an input image I with kernel K using a unit stride. The kernel weights are referred to as the parameters to be trained.

output feature maps (h) size and the input (I), after convolution with the kernel (K) and stride (s).

$$h_x = \frac{I_x - K_x}{s} + 1, \qquad h_y = \frac{I_y - K_y}{s} + 1, \tag{4.2}$$

where h_x and h_y represent rows and column of output feature maps produced by the convolutional layer. Equation (4.2) shows that the convolutional layer with larger stride values produces smaller sized output feature maps. Figure 4.9 shows an example where given input of size 5 × 5 which is convolved with the kernel of size 3 × 3 and stride of 2 × 2 sized. Recall from equation (4.2); the convolution operation produced an output size 2 × 2.

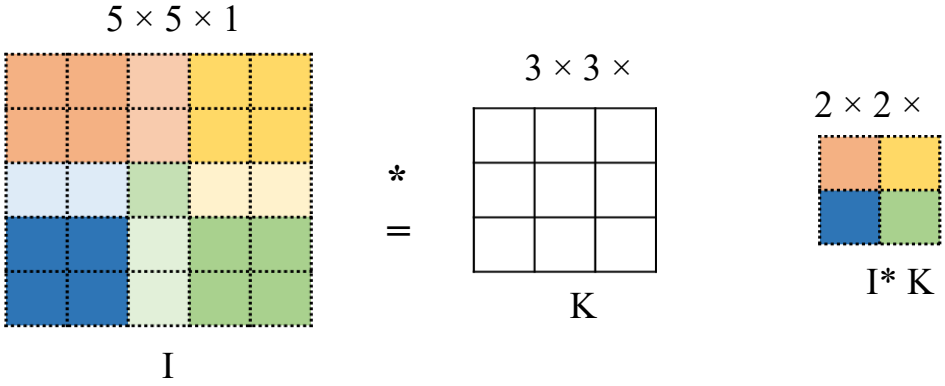


Figure 4.9: An example of convolution on an input image I with kernel K using a non-unit stride of size 2 x2.

[1]https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

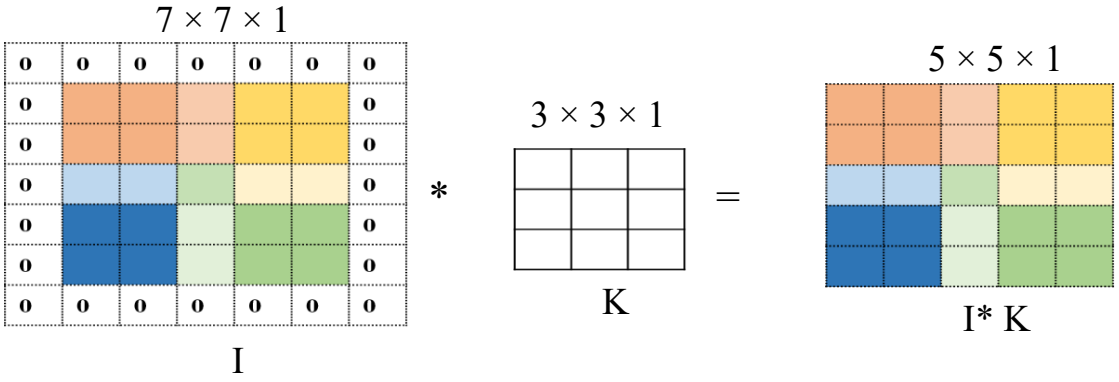


Figure 4.10: An example of using padding with unit stride.

Padding[1]: In order to get less overlapping between the receptive fields, we usually use the stride of larger size. This also produces smaller resulting output feature maps since the kernel is shifted over potential pixels. If someone wants to maintain the dimensionality of the output with respect to the input, they can use padding around the input (i.e., edge) either with zeros or randomly chosen image's values, as shown in figure 4.10. Padding is usually utilized in deep CNN architectures to maintain the size of the feature maps, elsewise they would reduce at every next layer, which is unwanted as it may increase the error rate of the network.

4.3.3 Batch Normalization (BN)

Batch normalization (BN) [93] is an adaptive re-parametrization technique which optimizes deeper CNNs networks by making their training easier. Training deep neural networks is intricate since each layer's inputs are affected during training as the parameters of the preceding layers change with saturating nonlinearities. Taking into account this fact, a careful parameter initialization and lower learning rate are required. On the other hand, this makes the network slower and also makes the network disreputably hard to train models. This phenomenon is referred to as internal covariate shift [93]. In these situations, it is tough to select a suitable learning rate, e.g., for stochastic gradient descent (SGD) (described in Section 4.5.3) updates the network's parameters in each layer simultaneously. Therefore, any layer in the network cannot learn independently. In [93], batch normalization algorithm was proposed to address

[1]https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

the above problem. BN increases the stability of a neural network by normalizing each layer input (i.e., the output of the previous layer). BN follows simple normalization procedure where the layer input is subtracted from batch mean and further their subtraction is divided by batch standard deviation; as described in the following paragraph.

Since the normalization is applied independently to every input layer. Let $\mathcal{B}$ is mini-batch of size *m* and *x* is an input of any layer, consequently $\mathcal{B}$ contains *m* values of this layer input activation, i.e . $\boldsymbol{\mathcal{B}} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots, \boldsymbol{x_m}\}$, the batch normalization procedure is illustrated in the following steps:

Step 1: Calculate mini-batch mean and variance, such that:

$$\mu_{\mathcal{B}} = \frac{1}{m}\sum_{i=1}^{m} x_i \quad , \tag{4.3}$$

$$\sigma_{\mathcal{B}}^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}}), \tag{4.4}$$

where $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}^2$ indicate mini-batch mean and variance, respectively.

Step 2: Calculate normalized values $\bar{x}$ for *x* such as,

For each $i = 1,2,\ldots m$

$$\bar{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \varepsilon}}, \tag{4.5}$$

where ε is a minimal positive scalar value to be added in the denominator to avoid getting an undefined result.

Step 3: After that, these normalized values are linearly transformed, such that:

$$y_i = \gamma(\bar{x}_i) + \beta \tag{4.6}$$

Subsequently, batch normalization incorporates these two trainable parameters (i.e., γ and β) to each activation, so the normalized output $\bar{x}_i$ is multiplied by a "standard deviation" parameter (γ) and add a "mean" parameter (β). As the normalizing value of each activation in the network reduces the network's expressive power. Therefore, in order to maintain the network's expressive power, the normalized values are linearly

transformed with these two trainable parameters. The new parametrization of the network is much more natural to be trained with SGD as it makes the SGD do de-normalization by varying only these two parameters or weights for every activation rather than decreasing the stability of the network by varying all the weights.

In CNNs, the same normalizing parameters γ and β are applied at every location in a feature map to guarantee that the feature map statistically is the same across all the spatial locations [83].

4.3.4 Non-linear Activation Layer

Usually, neural networks (NNs) generates output neuron can take on enormous values. When this output with such large values is fed as an input to the next layer without any modifications, then the next layer will transform these large values into even more significant. This larger value makes the process computationally intricate. The activation functions play a vital role in squeezing neural networks' output to be within certain bounds (e.g., between 0 and 1). The activation function can be two types; linear and non-linear activation functions.

Linear activation function: It is an ordinary linear function, i.e., $f(x) = x$, which linearly transforms input to output without any amendments, shown in figure 4.11.

Non-linear activation function: In contrast to the linear activation function, non-linear functions are utilized to separate the data that is not linearly separable. Furthermore, NNs employed complex functions, where non-linear activation functions allow them to estimate randomly complex functions. Thus, the absence of non-linearity in activation function, all layers of the network are equivalent to a single layer neural network. Considering this fact, there are some types of non-linear activation functions such as *sigmoid*, hyperbolic tangent function *Tanh*, and rectified linear unit (ReLU), etc. have been implemented. These functions are discussed as follows:

A. Sigmoid: This function is called as a logistic activation function. It squeezes the output to be bounded between 0 and 1. It converts large negative numbers to 0 and large positive numbers to 1. This function is represented mathematically in equation (4.7) and graphically in figure 4.11 (b).

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{4.7}$$

However, this function is easy to understand but has some drawbacks, which are mentioned as follows:

It vanishes the gradient: As the function is flat near 0 and 1, therefore during backpropagation gradient in neurons which are saturated in these regions is almost zero. As each gradient is updated in the backpropagation process where the number of factors

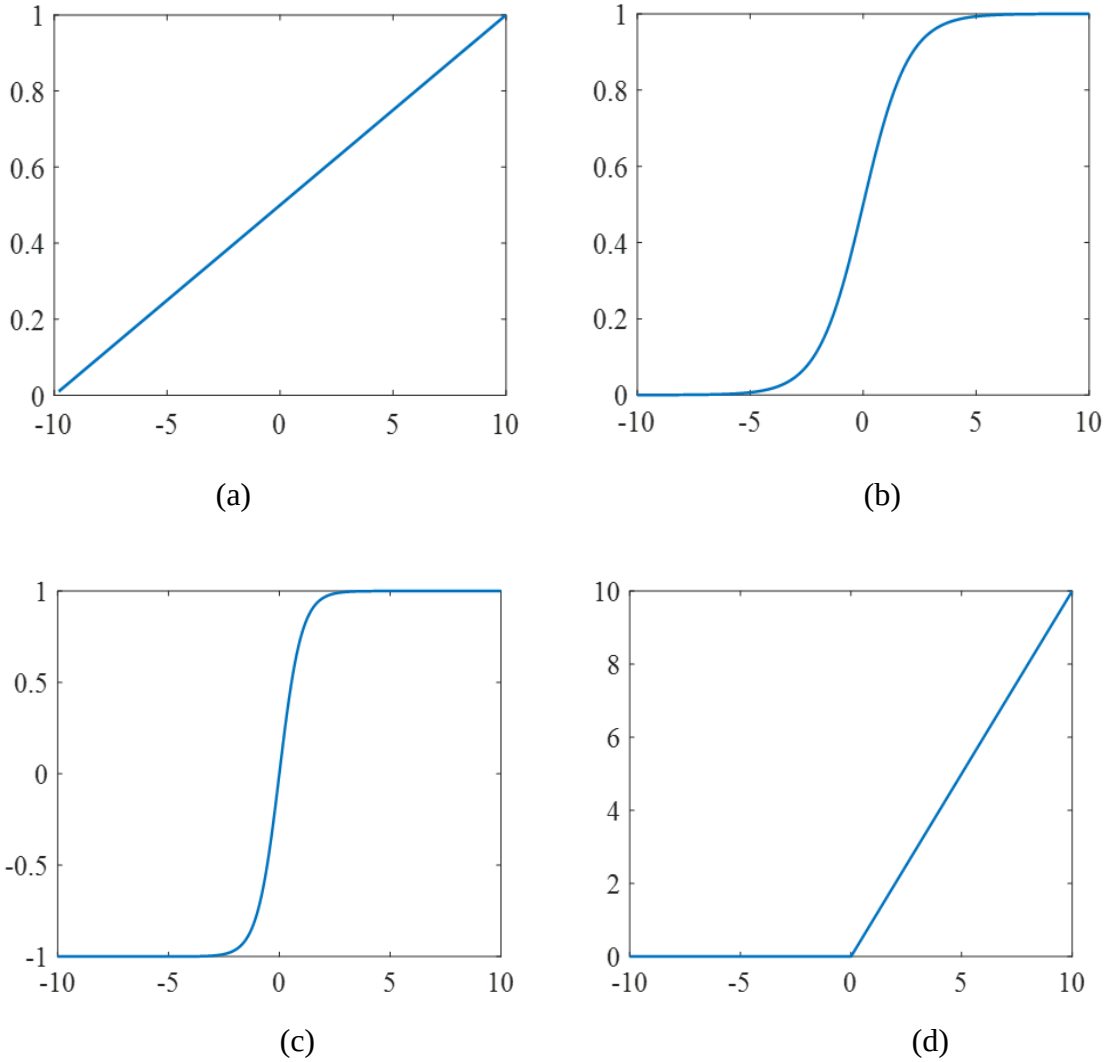


Figure 4.11: Activation functions: (a) Linear activation function, (b) nonlinear sigmoid function, (c) nonlinear Tanh function, and (d) ReLU.

is multiplied. These factors are derivatives of activation function neurons, weights, and

biases, etc. If you multiply the bunch of terms which are very small, i.e. less than 1 or nearly zero then resulting gradient will be very small, and if multiply the bunch of terms which are greater than 1, the resulting gradient will be infinity. In other words, if there is a deeper network comprises on several layers than gradient will be reduced in every next layer and the end, it may have vanished. Thus, the network is not able to perform backpropagation further anymore. Hence, gradient should be near to or equal to 1 ideally.

Not zero centered: The output of the sigmoid function is not centered which may make optimization tougher. *Computationally expensive*: As sigmoid incorporates *exp* (·) function, which makes it computationally expensive.

B. Hyperbolic Tangent Function (*Tanh*): *Tanh* is similar to *sigmoid*, which squeezes the values in the output but bounds them into the range between -1 and 1. Its mathematical representation is given in equation (4.8), and the graphical representation is depicted in figure 4.11 (c). In contrast to *sigmoid*, *Tanh* provides resulting output values; those are zero-centered since the scope is between -1 and 1. Considering this, *Tanh* produces positive inputs considered as positive (i.e., values near to 1), negative inputs corresponds to negative, and while zero input values mapped near zero. However, practically *Tanh* is more preferable over *sigmoid* as it provides zero-centered output values, but it also suffers from the vanishing gradient problem.

$$Tanh(x) = \frac{2}{1+e^{-x}} - 1 \tag{4.8}$$

C. Rectified Linear Unit (*ReLU*): To address the main vanishing gradient problem in above two activation functions, *ReLU* was proposed [81]. *ReLU* is expressed mathematically in equation (4.9) and graphically shown in figure 4.11 (d). From its mathematical representation, it contains a simple expression, which demonstrates that when the input $x > 0$ the output is x, elsewhere the output is zero. In other word, *ReLU* has gradient 1 (i.e. if $x = 1$) when output > 0, and zero otherwise. Also, *ReLU* is computationally simple and efficient than *sigmoid* and *tanh* as it does not contain any exponential function like them. Thus, *ReLU* significantly increases the convergence speed of the stochastic gradient descent (SGD). Nowadays, most of the deep learning

algorithms are using *ReLU* as non-linear activation function in their network architectures.

$$R(x) = max(0, x) \tag{4.9}$$

4.3.5 Pooling Layer

Pooling layer is referred to as a down-sampling layer, is incorporated periodically in-between successive convolution layers to reduce the spatial dimensions progressively excluding depth of the learned convolutional layer output. The most commonly used pooling operation is max pooling where the convolution layer output is down-sampled by taking its maximum value within the applied kernel. Except max pooling, there are several types of pooling layer like average pooling (where the output is down-sampled by taking only mean value within the kernel), and L_2-norm pooling, etc. In max pooling, specifically, both kernel and stride have similar sizes, i.e., 2 × 2. Figure 4.12 (a) and (b) exemplify the operation of max pooling operation. In figure 4.12 (a), the pooling operation slides the kernel of size 2 × 2 over a single slice of 4 × 4 dimensions, and the max value is chosen from each kernel, resulting output has reduced

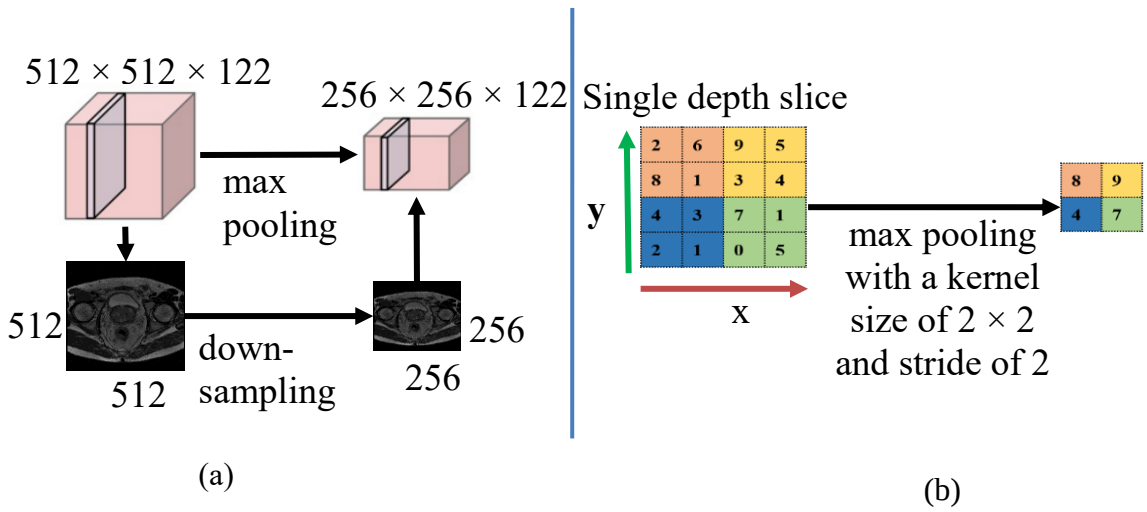


Figure 4.12: An example of max pooling operation; (a) multiple slice depth, (a) numerical illustration using single-slice depth.

the dimension of 2 × 2.

What are the advantages of using pooling layer? Firstly, it reduces spatial dimension, therefore less spatial dimension means less parameter are required to learn and also fewer parameters decreases the risk of getting over-fit. Secondly, pooling operation introduces some translation invariance in the output [94]. Translation invariance means that the system produces precisely the same response, regardless of how its input is shifted.

4.3.6 Fully Connected (FC) Layer

Fully connected (FC) layer is usually incorporated at the end of the network as shown in figure 4.6. Basically, FC takes an input volume from the previous convolutional layer or pooling layer and convert the volumetric output into *N*-dimensional vector and ensures that all activations are received. The main difference between the convolutional layer and FC layer is that neurons in the convolutional layer only connected to a local region of the input where the neurons share parameters. Whereas in FC layer, neurons are fully connected to all activations in the preceding layer likewise traditional NNs (described in section 4.1). Nevertheless, both layers have the same functionality as neurons in both layers still calculate dot products. Therefore, these both layers are interconvertible through reshaping process.

Since neurons in the FC are fully connected to all activations from the previous layer, their full connections result in spatial information loss. This is objectionable particularly in segmentation problem, where spatial information is crucial to learn. Therefore, some recent research in medical image segmentation replaced FC layer with a convolutional layer to have a kernel of 1×1 sized at the end of network architecture [47, 49].

4.3.7 Deconvolution Layer

As in Section 4.3.5, we describe pooling layer, which is used to reduce the number of network parameters by decreasing the spatial dimension of the convolutional layer output. In order to recover the spatial dimension of the output (i.e., especially in segmentation problem), un-pooling; a reverse operation of pooling layer is incorporated

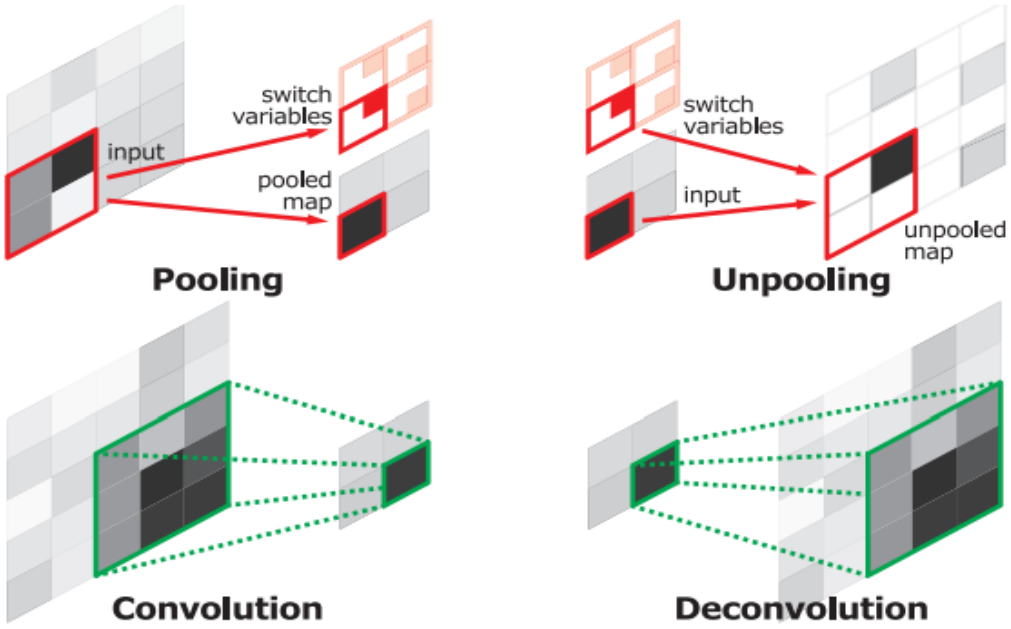


Figure 4.13: Working principle of deconvolution, the figure is taken from [88].

at the end of the network [88] to enlarge the pooled input. However, the un-pooling produces sparse output as shown in figure 4.13. Considering this problem in un-pooling, deconvolution layer is introduced in [86]. In contrast to the un-pooling layer, Deconvolution layer produces dense output as shown in figure 4.13. In brief, convolutional layers map several activations in a receptive field to a single activation whereas deconvolution layers perform vice versa.

4.3.8 Classification Layer

This is the last layer of the network. In this layer, softmax function is used to predict the un-normalized probability of mutually exclusive classes by calculating cross-entropy loss (discussed in Section 4.5.2). Let is a specific input corresponds to a particular class, c, then the softmax function predicts its probability value as,

$$prob\left(y = c \middle| \bar{x} = x_m\right) = \frac{e^{SR_c}}{\sum_n e^{SR_c}} \tag{4.10}$$

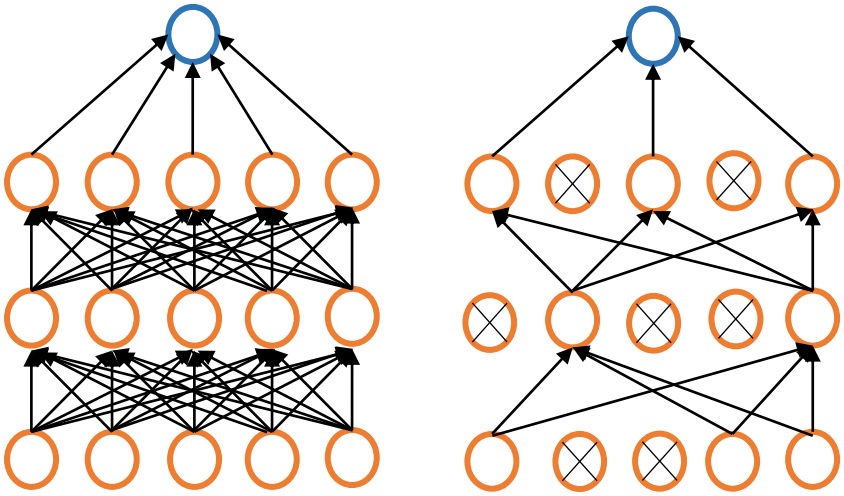


Figure 4.14: An example of dropout regularization. (a) is a standard network, (b) after dropout operation. The figure is redrawn from [96].

where *SR* is the score achieved for the specific class from the preceding layer of the network.

4.4 Dropout Layer

Overfitting is a major problem in neural network training. Generally, the network loses its ability of generalization with the over-fitted model on the training data. Mainly, deep networks are subject to learn noise along with fine pattern in training data set [95], and lead the network to learn a large number of weights with few train samples which may lead over-fitting and high variance. To address this problem, a simple method, dropout, to prevent the overfitting was proposed in [96]. Basically, the dropout process randomly drops the nodes and its connections from the network as shown in figure 4.14. By doing this, the network has weights which are less fitted to the data and consequently decreases the difference in performance between training data and validation data. Note that dropout layers are only incorporated throughout the training phase, not during testing or validation phase.

4.5 Network Training

The main aim of network training is to estimate the best fit for the weights throughout the network. Network learning/training process requires training data along with its ground truth. During training, the loss is computed in each iteration between predicted output and given ground truth at the last layer. Then the output is back propagated to the network for updating the weights for further loss minimization. This back propagation process is carried out either one-to-one or in batches. As the training of deep neural networks is an expensive process; therefore optimization algorithms have been developed. The optimization algorithm finds the best parameters that minimize loss. In this section, we will discuss the importance of backpropagation, cross entropy loss and the most commonly used optimization algorithm, stochastic gradient descent (SGD), which is used by many deep learning algorithms.

4.5.1 Backpropagation

Backpropagation is a process that is used to update or adjust the weights for a better generalization of the trained model. As a specific combination of weights that can be effective in minimization of loss function or cost function is a solution of the network optimization problem. The backpropagation is a process of calculating gradients. The network requires these gradients to calculate the weights. Then gradient-based optimization algorithm, i.e., SGD, uses backpropagation to calculate the gradient of the loss or cost function, which adjust or update the weights of the networks [96-97]. The backpropagation process computes the gradient of the error function at each iteration, thus optimizing the loss function at all iterations.

Initially, random weights are assigned for network training. These random weights do not make any relationship between the ground truth and the output; thus they cannot produce a meaningful prediction. For the better prediction, the weights should be updated or adjusted in a way there should be minimum error/loss between the ground truth and the predicted output class. Usually, weights are updated in the network in two computational phases, the forward pass and the backward pass.

Forward pass: It is a simple feedforward network that we discussed in section 4.1. Simply in CNN, an input image is given to network where feature maps are calculated

then these feature maps are fed to next layer on. This process continues until the last output layer.

Backward pass: In backward pass, weights are adjusted and updated by backpropagation process. The backpropagation is carried out in epochs where multiple epochs are employed in network training, and one epoch comprises several parts, like 1. *Loss function*: in the forward pass where predefined loss *L* is incorporated to minimize the error between the input and the output. The main aim to adapt the weights that can reduce the value of loss function, which is achieved by computing the derivatives of loss function with respect to its weights. 2. *Backward pass*: through backpropagation process, in each iteration the weights those contribute the most to the loss, are adjusted for decreasing the total loss. 3. *Weight update*: As the loss, function gradient is calculated in a negative direction, which makes a problem for backpropagation to calculate the loss function gradient with respect to the network weights. In this regard, it is essential to calculate the partial derivative $\frac{\partial L}{dW}$ in the backward pass to minimize the value of loss function.

4.5.2 Loss Function

The set of predicted scores needs to be optimized by adjusting the values of parameters learned in the network (weight filter or bias). The loss function defines quantitatively which set of parameters are ideal. For the softmax classifier, the loss function corresponds to cross-entropy loss for each vector of class scores SR:

$$\boldsymbol{L_n = -\log\left(\frac{e^{SR_n}}{\sum_n e^{SR_n}}\right).} \tag{4.11}$$

While cross entropy loss which is mostly used by CNN is given as,

$$\boldsymbol{L(p,q) = -\sum_{m=1}^{n} p(x_m)\, log q(x_m),} \tag{4.12}$$

where *q* denotes softmax function, and *p* is a prediction. The final loss is given as,

$$\boldsymbol{L = \sum_n^N L_n + \lambda R(W),} \tag{4.13}$$

where *L* is a total loss, and λ is scalar value multiplied with regularization term *R*. The total loss is minimized by optimization algorithm SGD.

4.5.3 Stochastic Gradient Descent (SGD)

Stochastic gradient descent (SGD) is a network optimization algorithm, which updates or adjusts the previous weights in the network by utilizing a linear combination of these preceding weight update V_t and the negative gradient $\nabla L(W)$. Let's α is the learning rate, and μ is momentum. The scalar value of α changes the negative gradient $\nabla L(W)$, and scalar value of α changes the prior update V_t. SGD uses these weights of α and β on preceding weight update V_t, and calculates the new updated values V_t+1, such that:

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t). \tag{4.14}$$

where α and μ are called network learning hyperparameters. These updated values consequently update weights W_t+1 at iteration t + 1, such that:

$$W_{t+1} = W_t + V_{t+1} \tag{4.15}$$

4.5.4 CNN Hyperparameters

Network hyperparameters are variables, which are defined before the training process and their proper selection is an essential part of developing CNN architecture. Different methods exist in choosing the Hyperparameters:

1. *Manual*: the values are chosen manually by an expert user.
2. *Search algorithms*: a random search or a grid search or algorithm is recommended, which identifies appropriate ranges for the network hyperparameters. Thereafter, the training process is then carried out by incorporating all combinations of parameters made available in these ranges.
3. *"Hyper" Optimization*: Here, a need exists to design an automated method, which can select hyperparameters those can optimize the performance of the model with better generalization according to the task. There are three commonly followed methods to optimize the hyper-parameters:
    a) Batch Gradient Descent: The cost function gradient is calculated over the entire dataset.

b) Mini-Batch Gradient Descent: A subset of the training dataset, the mini-batch, is fed into the network. Each mini-batch leads to updates.

c) Stochastic Gradient Descent: Parameters updates are made for each training example.

Many researchers have done the best practices by considering the following factors, as listed below:

Learning Rate: The learning rate might be understood as the rate at which the gradient updates to the parameters occur in the gradient direction. When learning rate value is too small, the model takes a long time to converge. Model diverges if the learning rate is too high. To guarantee optimal learning, an initial learning rate should be defined appropriately.

Weight Initialization: The local minimum reached by the training algorithm is highly dependent on the initialization of the weights matrix. Weights are initialized to vary in a random zero mean distribution, while bias can be set to 0.

CHAPTER 5

3D COLORECTAL TUMOR SEGMENTATION

In this chapter a brief introduction of 3D deep learning-based algorithms is given, they are successfully applied in different medical image segmentation applications; these are 3D fully connected convolutional neural networks (3D FCNNs) [48], *3D U-net* [45] *and 3D DenseVoxNet* [49]. *In Section 5.1, limitations of traditional level set algorithms and 2D CNNs are briefly elaborated. Sections 5.2, 5.3, 5.4 describe 3D FCNNs, 3D U-net and DenseVoxNet, respectively. We utilized them in colorectal tumor segmentation from 3D MRI. Based on their pros and cons, we proposed a novel algorithm, 3D Multiscale Densely connected neural network (3D MSDenseNet), explained in Section 5.5. Finally, 3D level-set algorithm is discussed in Section 5.6, which is used to refine the final output of each network.*

5.1 Limitations of Level Set And 2D CNN

Lately, level-set based segmentation algorithms have been widely used and become preferable algorithms for medical image segmentation [98-99]. Level-set approaches perform segmentation based on energy minimization problem by integrating different type of regularization (smoothing terms) and priors [100]. However, level-set based segmentation methods are more preferable in segmentation problems as they provide segmentation function with a tendency to change topological properties, but they required an appropriate contour initialization to obtain better segmentation results. Also, level-set based approaches are progressively deficient due to their simple appearance model [100]. In very recent, convolution neural networks (CNNs) based deep learning methods have been successfully employed in medical imaging, especially for segmentation and detection purpose [101-102]. In deep learning based methods, features for complex structures and patterns are erudite from well-defined large training

data sets. After that, these trained features are utilized for prediction. Unlike level-set based methods, deep learning can learn appearance models automatically from the extensive training data. Nevertheless, deep learning based approaches are not able to provide an explicit way of integrating prior shape and regularization [103].

Moreover, medical image segmentation is a task more chaotic than natural image segmentation. Firstly, the patient data is extremely diversified. In other words, the pattern of the same pathology varies among patients. Secondly, small and incomplete medical data sets make CNNs training more prone to overfitting [101]. Despite this, recent proposed CNN architectures demonstrated better performance than other machine learning based algorithms for medical image segmentation [50]. Thirdly, medical images such as MRI (Magnetic Resonance Imaging) or CT (Computed Tomography) scans are often in 3D volumetric form while the existing CNNs are in 2D nature. These 2D CNNs are applied slice-by-slice sequentially [44], thereby disregarding spatial information in the third dimension. An alternative solution was proposed in [42], where spatial information is enhanced by aggregating axial, sagittal and coronal planes in a one-to-one association, respectively. In [42], input slices are treated independently considering each orthogonal plane separately where the convolutional kernel is used only for two orthogonal planes when it is not shared with the third one. Although these 2D CNN-based methods demonstrated vast improvement in segmentation accuracy [50], the inherent 2D nature of the kernels limits their application when using volumetric spatial information. Hence, this solution is not able to utilize volumetric spatial information completely. Furthermore, 3D CNNs have been discouraged due to computational complexity and memory requirement [102]. Considering above problem in the 2D CNN for volumetric data, 3D CNN based algorithms [102] have been recently presented where 3D kernels are used instead of 2D, which extract spatial information across all three volumetric dimensions. Unlike other 2D CNNs based methods, they use the 3D kernel, which shares spatial information across all three dimensions. In [101], 3D-FCNNs provide efficient performance in segmenting brain lesions. In [102], 3D-FCNNs based on baseline CNN architecture is proposed for subcortical segmentation. 3D U-net [45], the first method was proposed for 3D volumetric biomedical images. After that, recently, DenseVoxNet [49] as 3D CNN, was proposed for 3D volumetric cardiovascular segmentation.

5.2 3D Fully Connected Convolutional Neural Networks (3D FCNNs)

In this study, 3D-FCNNs [48], based on CNNs baseline model by incorporating the concept from [102], is used for segmenting colorectal tumor. The architecture of the 3D-FCNNs is shown in Fig. 4.1, which we proposed with preliminary results [48]. Unlike traditional CNNs where the dimension of output maps are spatially reduced by pooling layer of stride 2, the 3D FCNNs architecture contains only convolutional layers with unit stride. As this network excludes pooling layers, therefore they did not use any up-sampling component (i.e., deconvolutional layer). At first convolutional layer, MRI volume as an input is taken, and then the input is convolved with 3D convolutional filters (i.e., kernels) to yield feature volume. In consequent layers, the input is taken as feature volumes of previous layers. Suppose F_k^{l-1} is k^{th} feature volume of (l-1) layer then m^{th} output feature volume of l layer is given as,

$$F_m^l(x,y,z) = H\left(\sum_k\left(\sum_{m,n,t} F_k^{l-1}(x-m,y-n,z-t)\otimes W_{ki}^l(m.n.t)\right)+b_l^k\right), \qquad (5.1)$$

where w_{ki}^l denotes the 3D filters with a kernel size of ($m\times n\times t$) is convolved element-wise over feature volume of each preceding layer, F_k^{l-1} $\otimes$ represents a 3D convolutional operation, b_l^k denotes bias and $H(\cdot)$ is non-linear activation function. Here, newly introduced non-linear activation function namely Parametric Rectified Linear Unit (PReLU) [109], is used as a replacement of the famous Rectified Linear Unit. PReLU function is given as,

$$H(F_i) = \max(0, F_i) + \alpha_i \cdot \min(0, F_i), \qquad (5.2)$$

where F_i represents input, $H(F_i)$ is output, and α_i is a trainable parameter which is required to learn to control the negative part of F_i, while α_i is almost zero in ReLU. Consequently, PReLU can adjust rectifiers to their input, thus by improving the network's accuracy with nearly zero additional computational cost and also reducing overfitting risk. This PReLU is applied on each layer except last layer (i.e., softmax layer). The proposed network contains two fully connected layers to retain spatial information further and learn complex patterns extracted in preceding layers. Finally,

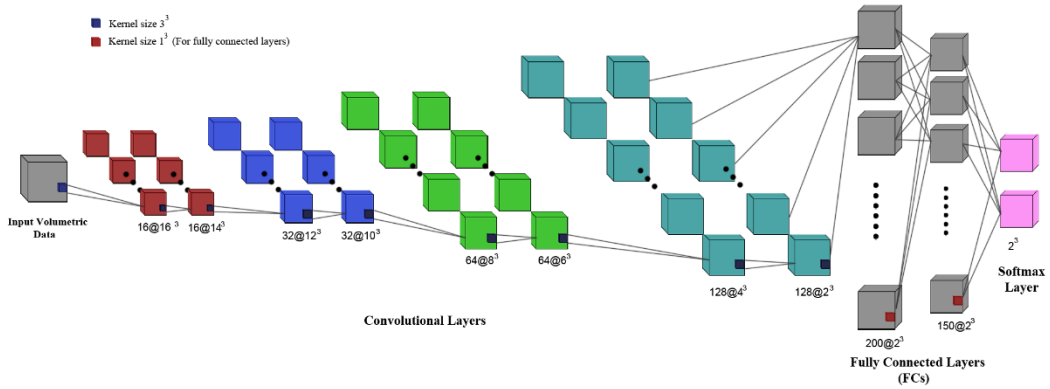


Figure 5.1: 3D FCNNs network architecture [13].

in end all the neurons are convened in *n* class feature maps where the normalized probability values are computed using *n* class feature maps by softmax function, such that:

$$probability\_map_n = \frac{\exp(F_l^n)}{\sum_{i=1}^{n} \exp(F_l^i)} \tag{5.3}$$

In this work, authors presented deeper architecture with small kernel size, 3×3×3, where each convolutional layer is once repeated with same kernel size. The size of the feature volume depends on kernel size (i.e., the *size of feature volume = kernel size* - 1). Therefore, feature volume produced each layer is smaller by 2 voxels than their input volumes as shown in figure 5.1. In [48], the 3D-FCNNs presents 8 convolutional layers with a number of filters followed by each convolutional layer are as 16, 16, 32, 32, 64, 64, 128, and 128, with a small kernel size of 3 × 3 × 3. Finally, two fully connected (FCs) layers with a kernel size of 1 × 1 × 1, contained 200 and 150 hidden neurons, individually, are incorporated. These FCs layers are followed by final softmax layer or classification layer, which produces probability maps.

5.3 3D-Unet

At very first U-Net [104], one of more prominent 2D CNN based algorithm was proposed for biomedical image segmentation. The U-Net has outperformed in medical image segmentation and won a number of competitions. Traditional 2D U-Net is depicted in figure 5.2. The U-net contains two different paths contraction (i.e., encoder) and expansion (i.e., decoder) as shown in figure 5.1. Each path has an architecture based on convolutional neural networks. The contraction path is also known as analysis path and utilizes chronological application of convolution and pooling layers where the spatial dimension of each convolutional layer output is reduced by pooling layer periodically. The expansion path, also known as synthesis or up-sampling path; where the original spatial dimension of the outputs from the contraction path is recovered. The expansion path contains deconvolution layers to produce full resolution segmentation output. Furthermore, the output feature maps from the contraction path are concatenated to the same resolution layer of expansion path through skip connection. These skip connections between the feature maps of layers of both paths having the same resolution, give fine details of the segmented object.

In [45], a 3D version of U-net was proposed for 3D volumetric data by

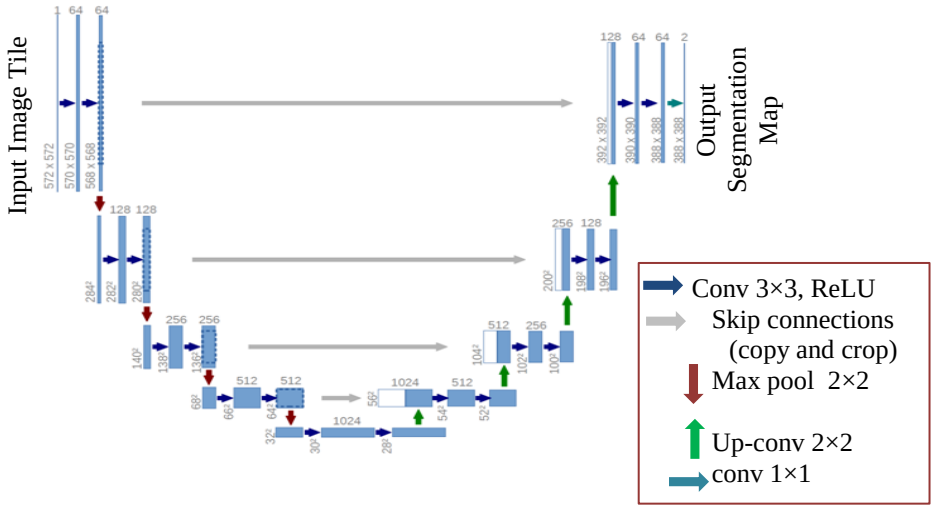


Figure 5.2: Network architecture of 2D U-net [104].

incorporating 3D operation. Particularly, the 2D kernel was replaced with 3D kernel corresponding to each layer (i.e., 3D convolutional layers, 3D max pooling and 3D

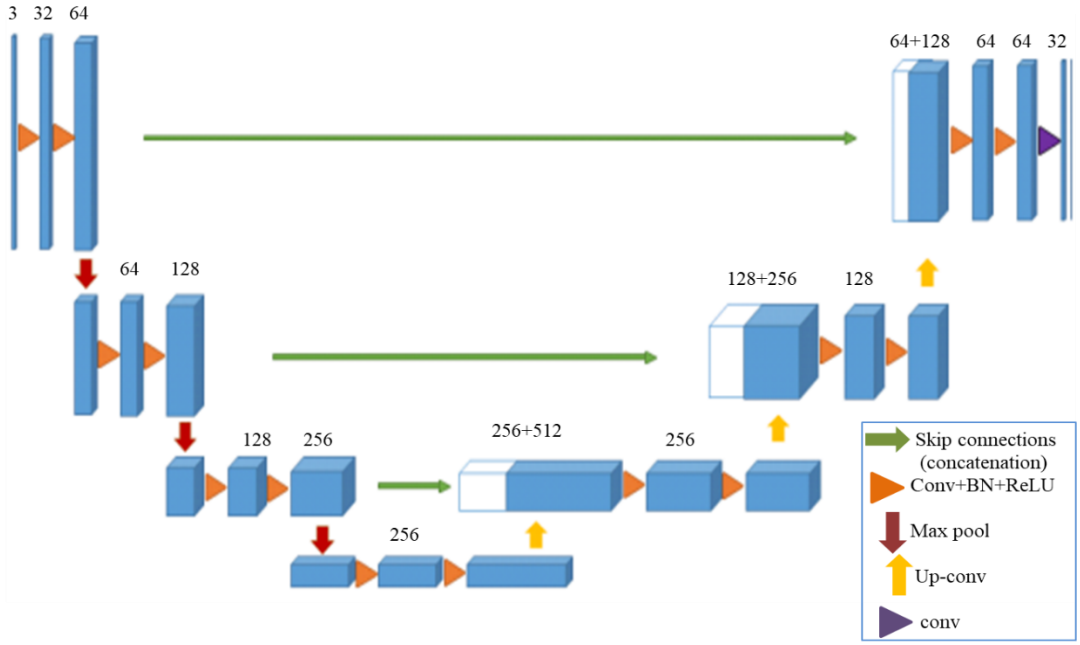


Figure 5.3: 3D U-net network architecture [45].

deconvolutional layers). Additionally, 3D U-net integrates batch normalization [101] for faster convergence. Figure 5.3 exemplifies the 3D U-net architecture. Similar to traditional 2D U-net, the 3D U-net has both contraction and expansion paths. In the contraction path, each layer contains convolutions of 3 × 3 × 3 sized kernels followed by batch normalization (BN) and a rectified linear unit (ReLU), and subsequently a max pooling of 2 × 2 × 2 sized kernels with strides of two in each dimension. In the expansion path, each deconvolutional layer contains the kernel of 2 × 2 × 2 sized with strides of two in each dimension, trailed by two convolutions of 3 × 3 × 3 sized kernel each followed by batch normalization (BN) and a rectified linear unit (ReLU). Similarly, in 3D U-net, skip connections from layers of the same resolution in the contraction path provide the significant high-resolution features to the expansion path. Finally, the last layer convolution layer of 1 × 1 × 1 sized kernel decreases the number

of output channels to the number of labels. The 3D U-net architecture has 19 million total parameters.

5.4 DenseVoxNet

Generally, in feed-forward CNN or ConvNet, the output of the l^{th} layer is represented as, X_l, which is obtained by mapping non-linear transformation H_l from the output of the preceding layer X_{l-1}, such that:

$$X_l = H_l(X_{l-1}) \tag{5.4}$$

where H_l is composed of convolution or pooling operation followed by non-linear activation function such as rectified linear Unit (ReLU) or Batch Normalization-ReLU (BN-ReLU) etc. Recent work in computer vision problem has addressed that as deeper the network with more layers as more accuracy with better learning [105]. However, the performance of deeply modelled network tends to decrease, and its training accuracy is saturated with the network depth increasing due to vanishing/exploding gradient [130]. Later, [104] solved this vanishing gradient problem in the deep network by incorporating skip-connection, which propagates output features from layers of same resolution in the contraction path to the output features from the layers in the expansion path. Nevertheless, this skip-connection allows the gradient to flow directly from the low-resolution path to high-resolution path, which makes training easy, but this generally produces enormous feature channels in every layer and lead network to adjust a large number of parameters during training. To overcome this problem, Huang et al. [105] introduce a densely connected network (DenseNet). The DenseNet extends the concept of skip connections by constructing a direct connection (as shown in figure 5.4) from every layer to its corresponding previous layers to ensure maximum gradient flow between layers. In DenseNet, feature maps produced by the preceding layer were concatenated as an input to the advanced layer, thus providing a direct connection from any layer to subsequent layer, such that:

$$\boldsymbol{X_l = H_l([X_{l-1}, X_{l-2}, X_{l-3}, \ldots, X_0])} \tag{5.5}$$

where $[\cdots]$ represents the concatenation operation. In [105], DenseNet has emerged as an accurate and efficient method for natural image classification. Yu et al. [49]

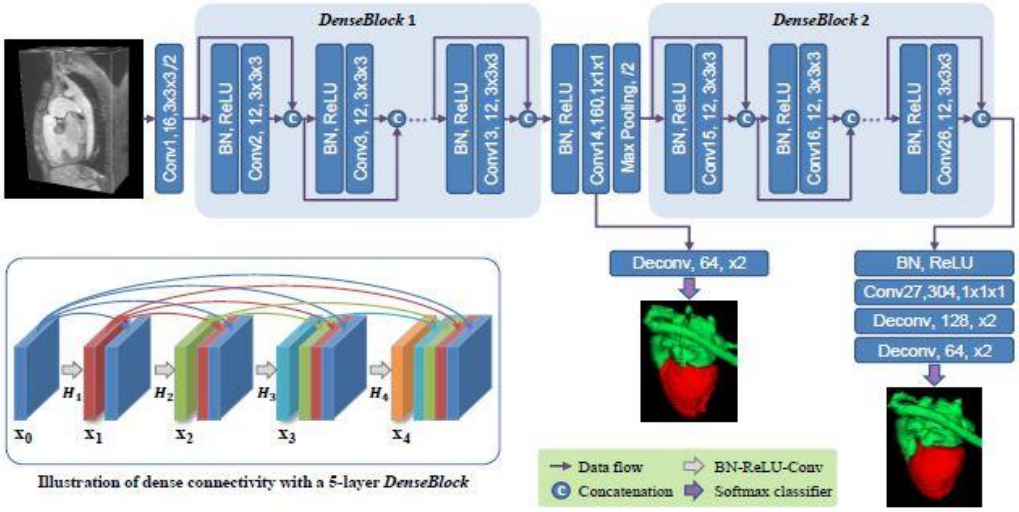


Figure 5.4: DenseVoxNet architecture, the figure is taken from [49].

proposed densely-connected volumetric convolutional neural network (DenseVoxNet) for volumetric cardiac segmentation which is an extended 3D version of DenseNet [105]. DenseVoxNet architecture is depicted in figure 5.4. The DenseVoxNet utilizes two dense blocks followed by pooling layers. The first block learns high-level feature maps and second block learn low-level feature maps followed by pooling layer, which reduces the resolution of the learned high-level feature maps in the first block. Finally, high-resolution feature maps are restored by incorporating deconvolution layers.

Figure 5.4 shows the DenseVoxNet architecture that takes on fully convolutional network architecture and adopts both down-sampling and up-sampling components to lead the network for end-to-end learning. The network architecture prefixed the first dense block with a convolution layer of 16 output channels with a stride of two to learn primitive features. In this prefixed convolution layer, the input feature map size is down-sampled by the stride of two for better efficiency of memory space, and it also increases the receptive field to enclose more information for better prediction. Afterward, the down-sampling components were further divided into two dense blocks, as discussed earlier. Each dense block has consisted on 12 densely connected transformation layers where every transformation layer has consecutively consisted on a batch normalization (BN), and ReLU followed by convolution of $3 \times 3 \times 3$ sized kernels with feature growth rate, k, of 12. A transition layer is used to connect these

dense blocks. The transition layer is composed of a BN, a ReLU followed by con convolution of 1 ×1 × 1 sized kernel and 2 ×2 × 2 max-pooling layers.

In order to recover the shape details from coarser (i.e., down-sampled) feature maps, the network contains 3D up-sampling block. The 3D up-sampling block has consisted on a BN, a ReLU, followed by a convolution of 1 ×1 × 1 sized kernel and 2 ×2 × 2 deconvolutional (Deconv) layers to make segmentation prediction map to have equal size of the input data. Finally, the up-sampling block is then trailed by a convolution of 1 ×1 × 1 sized kernel and soft-max layers to produce the final label map of the segmentation. Furthermore, DenseVoxNet incorporates a dropout layer with a dropout rate of 0.2 in its network after each convolution layer to increase the robustness in the network against overfitting problem. The DenseVoxNet is a method that is more preferable over 3D U-net as the DenseVoxNet has lesser total parameters approximately 1.8 million than 3D U-net with 1.9 million.

5.5 Proposed Method (3D MSDenseNet)

In DenseVoxNet, early layers of the first block learn fine-scale features (i.e., high-level features) based on a small receptive field while coarse-scale features (i.e., low-level features) are learned by later layers of the second block with a larger receptive field. In short, fine-scale and coarse-scale features are learned in early and later layers, respectively, which limit the network to learn multi-scale contextual information throughout the network and may lead the network in poor performance [106].

Considering multi-scale contextual learning problem in DenseVoxNet, a novel method to overcome the above challenges in 3D volumetric segmentation is presented. We propose 3D multi-scale densely connected convolutional neural network (3D-MSDenseNet), a 3D volumetric network that is an extension recently proposed 2D Multi-scale dense networks (MSDNet) for natural image classification [107]. summary, we have employed 3D-MSDenseNet to colorectal tumor segmentation problem with the following contributions:

1. Multiscale training scheme with parallel 3D densely interconnected convolutional layers for two-dimensional depth and coarser scales are utilized where low, and high-level features are generated from each scales individually.

A diagonal propagation layout is incorporated to decouple the depth features with the coarser features from the first layer and on, thus maintaining local and global contextual information throughout the network to improve segmentation results efficiently.

2. Proposed network is based on volume-to-volume learning and interference, which eradicate computation redundancy.

3. The proposed method is validated on colorectal tumor segmentation from 3D MR images and have achieved better and comparable segmentation results with the state-of-the-art. The proposed method may be applied in other related applications.

In 3D-MSDenseNet, we have two interconnected levels, depth level and scaled level for simultaneous computation of high and low-level features, respectively. Let consider X_0^1 is an original input volume and feature volume produced by layer l at scale s represented as, x_l^s. Considering two scales in the network (i.e., s_1 and s_2), we represent depth level (horizontal path) and scaled level as s_1 and s_2 individually, as shown in figure 5.5. The first layer is an inimitable layer where the feature map of very first convolution layer is divided into respective scale s_2 via pooling of stride of power 2. The high-resolution feature maps (x_l^1) in a horizontal path (s_1) produced at subsequent layers (l >1) are densely connected by following (Huang et al. [105]). While output feature maps of subsequent layers in a vertical path (i.e., coarser scale, s_2) are results of a concatenation of transformed features maps from previous layers in s_2 and down-sampled features maps from previous layers of s_1, propagated as a diagonal way, as shown in figure 5.5. In this way, output features of coarser scale s_2 at layer l in our proposed network expressed as:

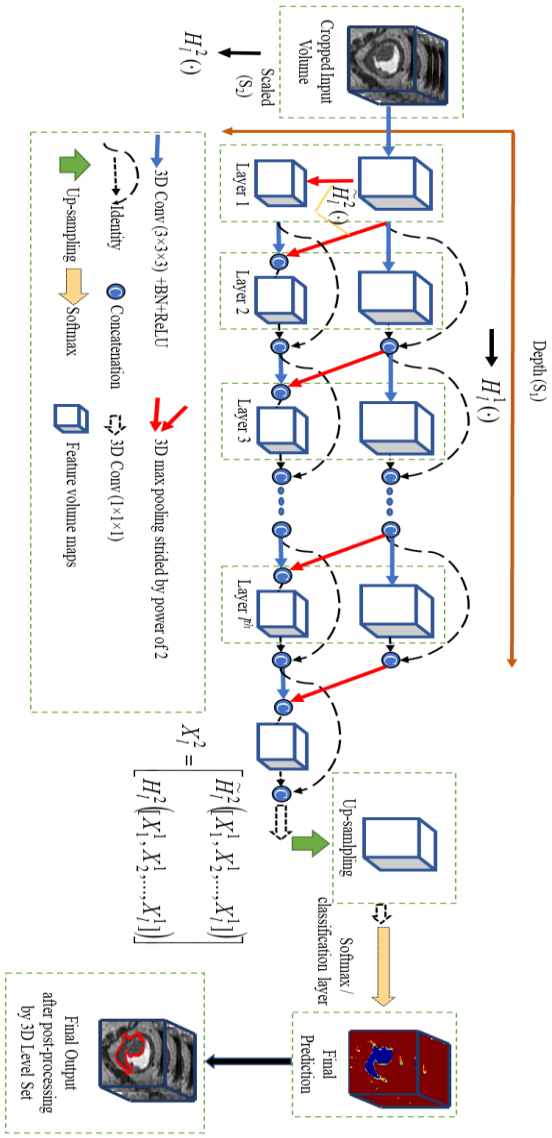


Figure 5.5: Block diagram of proposed network architecture.

$$X_l^2 = \begin{bmatrix} \tilde{H}_l^2([X_1^1, X_2^1, ..., X_l^1]) \\ H_l^2([X_1^1, X_2^1, ..., X_l^1]) \end{bmatrix}, \tag{5.6}$$

where $[\cdots]$ denotes concatenation operator, $\tilde{H}_l^2(\cdot)$ represents to feature maps from finer scale s_1 which are transformed by pooling layer of stride of power 2, diagonally (as shown in figure 5.5), and $H_l^2(\cdot)$ indicates to feature maps from coarser scale s_2 transformed by regular convolution. Here, $\tilde{H}_l^2(\cdot)$ and $H_l^2(\cdot)$ have the same size of feature

maps. In our network, classifier only utilizes the feature maps from the coarser scale at layer *l* for final prediction.

Network Architecture: Our network architecture is composed of dual parallel paths, depth and scaled path, as illustrated in figure 5.5, which achieves 3D end-to-end training by adopting the nature of the fully convolutional network. The depth path consists of eight transformation layers, and the scaled path consists of nine transformation layers. In each path, every transformation layer is composed of a BN, a ReLU followed by 3×3×3 convolution (Conv), by following the similar fashion of DenseVoxNet. Furthermore, 3D up-sampling block is utilized likewise DenseVoxNet. Like DenseVoxNet, the proposed network uses dropout layer with a dropout rate of 0.2 after each Conv layer to increase the robustness in the network against overfitting problem. Our proposed method has total parameters approximately 0.7 million, which is much fewer than DenseVoxNet [49] with 1.8 million and 3D U-net [45] with 19.0 million parameters. The implementation code of the proposed method in 3D-Caffe is publically available online at:
http://host.uniroma3.it/laboratori/sp4te/teaching/sp4bme/documents/codemsdn.zip

5.6 3D Level Set

The 3D level set is used in the testing phase as a post-preprocessor to refine the output of each method by integrating smoothing function and prior information. In this work, 3D level-set based on 3D geodesic active contour algorithm [108] is incorporated to refine the initial segmentation obtained by each method discussed above. 3D active contour regulates the tumor boundaries more precisely. The mathematical derivation of the geodesic active contour is well explained in [108] and [98]. This algorithm presents an association between active contours and the calculation of geodesic or nominal distance curves. This association gives stable boundary detection even in the presence of abundant gaps and variations of gradients. Suppose $\varphi(\mathbf{P}_l, t = 0)$ be level-set function with given initial surface at $t = 0$. Here, $\mathbf{P}_l$ is the probability map obtained from each method and is used as the initial surface to initialize 3D level-set. The level-set function is evolved to refine the tumor boundaries by partial differential equation [108], such that:

$$\frac{\partial \varphi}{\partial t} = \alpha \mathbf{X}(\mathbf{P}_l) \cdot \nabla \varphi - \beta \mathbf{Y}(\mathbf{P}_l) |\nabla \varphi| + \gamma \mathbf{Z}(\mathbf{P}_l) \kappa |\nabla \varphi|, \tag{5.7}$$

where $\mathbf{X}(\cdot)$ is a convection function, $\mathbf{Y}(\cdot)$ represents expansion or propagation function, and $\mathbf{Z}(\cdot)$ is a spatial modifier or smoothing function. And α, β, and γ is constant scalar quantities to make tradeoff among convection, propagation, and spatial modifier functions. As the initial zero-level surface is required by a level-set algorithm then the initial surface is propagated in a particular direction (inward, outward) with speed, this is controlled by the propagation function and smoothness of regions concerning mean curvature κ is controlled by spatial modifier function. The termination of this process depends on a convergence criterion or a maximum number of iteration. Here, we have set maximum iteration as 50.

CHAPTER 6

3D COLORECTAL TUMOR SEGMENTATION — EXPERIMENTAL RESULTS

In this chapter, experimental results obtained by each method (discussed in chapter 5), are presented. In Section 6.1, we present the details of our experimental volumetric data. Section 6.2 describes the networks' parameters setting for their training, and we describe the evaluation metrics that we used to assessed and compare the segmentation results produced by each method in Section 6.3. Experimental results are discussed in Section 6.4, and finally, the discussion is given in Section 6.5.

6.1 Experimental Data Sets

All algorithms described in the previous chapter are validated and compared on T2-weighted 3D Colorectal MRI. Data were collected from two hospitals; Department of Radiological Sciences, Oncology, and Pathology, University La Sapienza, AOU Sant'Andrea, Via di Grottarossa 1035, 00189 Rome, Italy and Department of Radiological Sciences, University of Pisa, Via Savi 10, 56126 Pisa, Italy. The overall dataset consisted on 43 volumes T2-weighted MRI, and each MRI volume has consisted on several slices which are varied in number across subjects as 69 ~122 and have dimension as 512 × 512 × (69 ~122). Their voxels with voxel spacing also varying from 0.46 × 0.46 × 0.5 to 0.6 × 0.6 × 1.2 mm/voxel across each subject. As the data have a slight slice gap, thus we did not incorporate any spatial resampling. Cross-validation was performed in 100 rounds by partitioning the dataset into 30 volumes for training and 13 for testing. The colorectal MR volumes were acquired in a sagittal view on a 3.0 Tesla scanner without contrast agent. All MRI volumes went for pre-processing where they were normalized so that they have zero mean and unit variance. We cropped all the volumes with a size of 195 × 114 × 61.

Furthermore, during training, the data were augmented with random rotations of 90^0, 180^0 and 270^0 in the sagittal plane to enlarge the training data. Also, the colorectal tumor was manually segmented in all volumes by two medical experts using ITK-snap software [109-110]. These manual delineations of tumor s from each volume were then used as ground truth labels to train the network and validate the network in the test phase.

6.2 Networks Training Procedure

All the networks, 3D FCNNs, 3D U-net[1], DenseVoxNet[2], and proposed network 3D MSDenseNet[3], as their network architectures are described in previous Chapter 5, were originally implemented in Caffe library [111]. For a fair comparison, we have applied the same training procedure, which is almost utilized by 3D U-net, DenseVoxNet.

Firstly, we randomly initialized weights with a zero-mean Gaussian distribution by setting values as, μ = 0, σ = 0.01. The stochastic gradient descent (SGD) algorithm (described in Chapter 4, Section 4.5.3) is used to realize the network optimization. We set the meta-parameters for the SGD algorithm to update the weights as; batch size = 4, weight decay = 0.0005 and momentum was set as 0.05. We set an initial learning rate as 0.05 and divide by 10 every 50 epochs. Similar learning rate policy in DenseVoxNet, i.e., "poly," was adopted for all methods. The "poly" learning rate policy changes the learning rate over each iteration by following a polynomial decay, where the learning rate is multiplied by the term[4] $\left(1-\frac{iteration}{\max imum\_iterations}\right)^{power}$ [112]. Here the term power was set as 0.9, and 40000 maximum iterations were set. Moreover, considering limited GPU memory, the training volumes were cropped randomly with sub-volumes of 32 × 32 × 32 sized as an input to the network and the major voting strategy [113] was incorporated to obtain final segmentation results from the predictions of the overlapped sub-volumes. Finally, the softmax with cross-entropy loss was used to measure the loss between network predicted output and the ground truth labels.

[1]https://lmb.informatik.uni-freiburg.de/resources/opensource/unet.en.html

[2]https://github.com/yulequan/HeartSeg

[3]http://host.uniroma3.it/laboratori/sp4te/teaching/sp4bme/documents/codemsdn.zip

[4]https://github.com/BVLC/caffe/wiki/Solver-Prototxt

6.3 Evaluation Metrics

In this study, three evaluation metrics were used to validate and compare the proposed algorithm; namely Dice similarity coefficient (DSC) [114], Recall rate, and Average symmetric surface distance (ASD) [115]. These metrics are briefly explained as follows:

6.3.1 Dice Similarity Coefficient (DSC)

The Dice similarity coefficient is widely explored performance metric in medical image segmentation. It is also known as overlap index. It computes a general overlap similarity rate between the given ground truth label and the predicted segmentation output by a segmentation method. DSC is expressed as,

$$DSC(S_p, S_g) = \frac{2TP}{FP + 2TP + FN} = \frac{2|S_p \cap S_g|}{|S_p| + |S_g|}, \tag{6.1}$$

where S_p and S_g are predicted segmentation output and ground truth label, respectively. FP, TP, and FN indicate false positive, true positive and false negative, individually. DSC gives the score between 0 and 1, where 1 gives the best prediction and indicates that the predicted segmentation output is identical to the ground truth.

6.3.2 Recall Rate

The recall is also referred to as a true positive rate (TPR) or sensitivity. We utilized this term as a voxel-wise recall rate to assess the recall performance of different algorithms. This metric performance measures misclassified and correctly classified tumor-related voxels. It is mathematically expressed as,

$$Recall = \frac{TP}{TP + FN} = \frac{|S_p \cap S_g|}{|S_g|}. \tag{6.2}$$

It also gives a value between 0 and 1. A higher value indicates the better the prediction.

6.3.3 Average Symmetric Surface Distance (ASD)

Average symmetric surface distance (ASD) measures an average distance between the sets of boundary voxels of the predicted segmentation and the ground truth, mathematically given as,

$$ASD(S_p, S_g) = \frac{1}{|S_p| + |S_g|} \times \left(\sum_{p_k \epsilon S_p} d(p_k, S_g) + \sum_{p_g \epsilon S_g} d(p_g, S_p) \right) \quad (5.3)$$

where p_k and p_g represent k^{th} voxel from S_p and S_g sets, respectively. d denotes the point to set distance and defined as $d(p_k, S_g) = \sum_{p_g \epsilon S_g} \| p_k - p_g \|$, here $\|\cdot\|$ is Euclidean distance. Lower values of ASD indicate higher closeness between two sets, hence, better segmentation and vice versa.

6.4 Experimental Results

In this section, we experimentally evaluate the efficacy of multiscale end-to-end training scheme of our proposed method, where parallel 3D densely interconnected convolutional layers for two-dimensional depth and coarser scales paths are incorporated (as described in Chapter 5, Section 5.5). In this work, the proposed network is assessed on 3D colorectal MRI data. For more comprehensive analysis and comparison of segmentation results, each dataset was divided into ground truth masks (i.e., manual segmentation is done by medical experts), training and validation subsets. Quantitative and qualitative evaluations and comparisons with baseline networks are stated for the segmentation of the colorectal tumor. First, we analyzed and compared the learning process of each method, described in Section 6.4.1. Secondly, we assessed the efficiency of each algorithm qualitatively, and Section 6.4.2 presents a comparison of qualitative results. Finally, we evaluated the segmentation results yielded by each algorithm quantitatively, using evaluation metrics as described below in Section 6.4.3.

6.4.1 Learning Curves

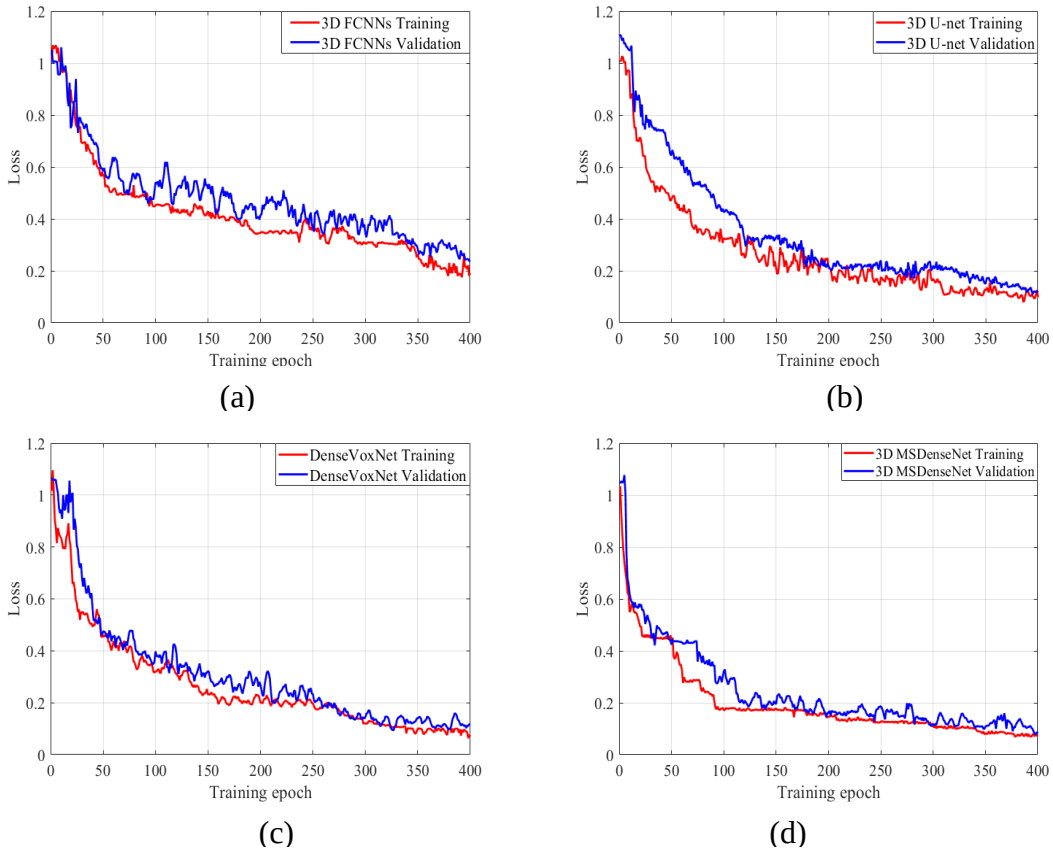


Figure 6.1: Comparison of learning curves of state-of-art methods. (a), (b), (c) and (d) are learning curves correspond to 3D FCNNs, 3D U-net, DenseVoxNet, and proposed method 3D MSDenseNet, respectively.

Learning process of each method is illustrated in figure 6.1, where loss versus training and loss versus validation is compared, individually, among baseline methods. Figure 6.1, demonstrates that each method does not observe severe overfitting as their validation loss decreases along with decrement in training loss consistently because each method has adopted 3D fully convolutional architecture where error backpropagation is carried on per-voxel-wise strategy instead of patch-based training scheme [116]. In other words, every single voxel is utilized as an independent training sample, which enlarges the training database dramatically and thus reducing the over-fitting risk. In contrast to this, traditional patch-based training scheme [116], needs dense prediction (i.e., many patches are required) for each voxel in 3D volumetric data; thus this redundant patches computation for every voxel makes the network computationally complex and impractical for volumetric segmentation.

By comparing the learning curves of 3D FCNNs (figure 6.1 (a)), 3D U-net (figure 6.1 (b)), and DenseVoxNet (figure 6.1 (c)), the 3D U-net and DenseVoxNet converge much faster with minimum error rate than the 3D FCNNs. This demonstrates that the 3D U-net and DenseVoxNet successfully overcome gradients vanishing/exploding problems through reuse of the features of earlier layers to later layers as discussed in chapter 5, section 5.3 and 5.4, respectively. On the other hand, it is also shown that there is no any significant difference between learning curves of the 3D U-net and DenseVoxNet, however, in the beginning; the DenseVoxNet attains a steady drop of validation loss. It further proves that reusing of features from subsequent layers to every next layer by DenseVoxNet, which propagates maximum gradients than those skip connections utilized by 3D U-net, which propagates output features from layers of the same resolution in the contraction path to the output features from the layers in the expansion path. Furthermore, figure 6.1 (d) shows that the proposed method, which incorporates a multiscale dense training scheme, has a minimum loss rate than all. It reveals that the multiscale training scheme in our proposed method optimizes and speeds up the network training procedure, thus proposed method has the fastest convergence with the lowest loss rate than all.

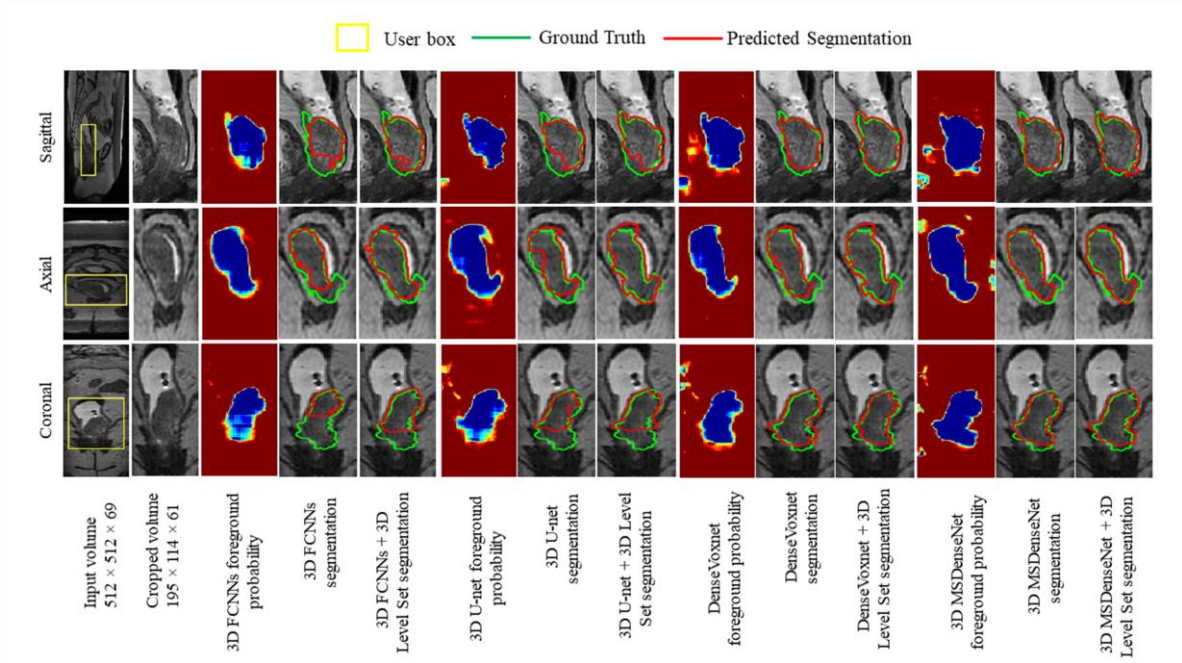
6.4.2 Qualitative Results

In this section, we presented the qualitative results to assess the effectiveness of each segmentation method qualitatively on the colorectal tumor segmentation task. Figure 5.2 (a) gives a visual comparison of colorectal tumor segmentation results achieved from each method. In figure 6.2 (a), from left to right; the first two columns are the raw MRI input volume and its cropped volume, rest every three columns are related to segmentation results produced by each method where each column represents predicted foreground probability, initial colorectal segmentation results and then refined segmentation results by 3D level set, correspondingly. Moreover, the segmentation results produced by each method are outlined with a red marker and overlapped with true ground truth, which is outlined with a green marker. In figure 6.2 (b), we overlapped the segmented 3D mask with the true ground truth 3D mask to

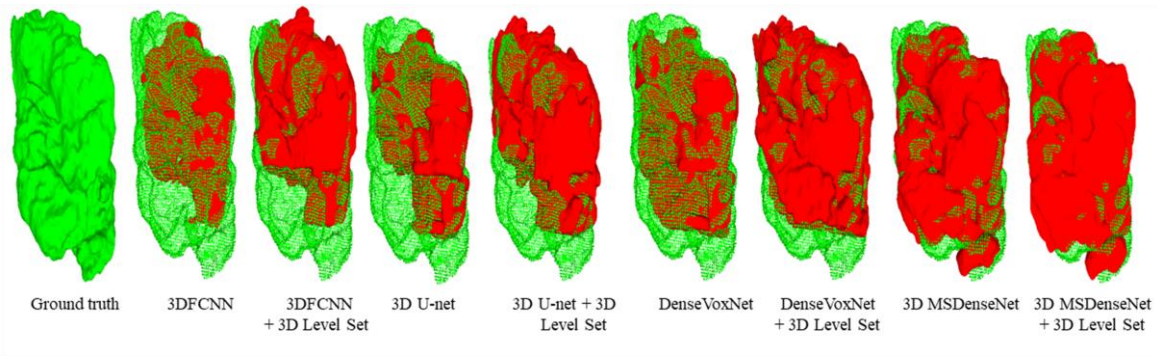
observe visually false negative rate in the segmentation results. It is observed that the proposed method 3D MSDenseNet outperforms than all with less false negative rate followed by DenseVoxNet, 3D U-net, and 3DFCNNs. It is also noteworthy that segmentation results of each method improve by incorporating 3D level set.

6.4.3 Quantitative Results

Table 6.1 presents the quantitative results of colorectal tumor segmentation produced by each method. The quantitative results are achieved by computing the mean and standard deviation of each performance metric for 13 test volumes. We initially compared the results obtained by each method without post-processing of the 3D level set, and we call them baseline methods. Afterward, we presented a comparison in among them by incorporating 3D level set as a post-processor to refine the boundaries of the segmented results of these baseline algorithms. In this way, we have got total eight settings and we name them as; 3D FCNNs, 3D U-net, DenseVoxNet, 3D MSDenseNet, 3D FCNNs + 3D Level Set, 3D U-net + 3D Level Set, DenseVoxNet + 3D Level Set, 3D MSDenseNet + 3D Level Set, respectively. From Table 6.1, it reveals that the 3D FCNNs has the lowest performance in all metrics followed by 3D U-net and DenseVoxNet. Whereas, the proposed method has maintained his performance by achieving the highest value of the dice similarity coefficient (DSC), recall, and the lowest value of ASD. When comparing the methods after post-processing where every method has effectively improved their performance in all. Where 3D FCNNs + 3D Level Set has improved DSC and recall as 16.44% and 15.23%, individually, and it reduced ASD to 3.0029 from 4.2613 mm. Similarly, 3D U-net + 3D Level Set and DenseVoxNet + 3D Level Set have attained improvements in (DSC and recall) as; (5% and 5.97%), and (4.99% and 4.29%), correspondingly. They both also have got a noteworthy reduction in ASD, as 3D U-net + 3D Level Set and DenseVoxNet + 3D Level Set reduce ASD (to 2.9 from 3.02) and (to 2.52 from



(a)



(b)

Figure 6.2: Qualitative comparison of colorectal tumor segmentation results produced by each method. In (a), from left to right columns are the raw MRI input volume, cropped volume, first three columns are corresponded to predicted probability by 3DFCNNs, segmentation results by 3D FCNNs (red), and 3D FCNNs + 3D Level Set (red) overlapped with true ground truth (green), correspondingly. Similarly Second, third and fourth three columns are related to predicted probability, segmentation results by rest of methods; 3D U-net (red), 3D U-net + 3D Level Set (red), DenseVoxNet (red), DenseVoxNet + 3D Level Set (red), 3D MSDensenet (red), and 3D MSDensenet + 3D Level Set (red), respectively. In (b), we have overlapped the 3D masks segmented by each method with the ground truth 3D mask. In (b), from left to right are ground truth 3D mask, overlapping of segmented 3D mask by 3D FCNNs (red), 3D FCNNs + 3D Level Set (red), 3D U-net (red), 3D U-net + 3D Level Set (red), DenseVoxNet (red), DenseVoxNet + 3D Level Set (red), 3D MSDensenet (red), and 3D MSDensenet + 3D Level Set (red) with the ground truth 3D mask (green points). The green points which are not covered by the segmentation results (red) of each method are referred as false negatives.

Table 6.1: Quantitative comparison of colorectal tumor segmentation results

Methods	Performance Metrics		
	DSC	**Recall**	**ASD [*mm*]**
3D FCNNs [48]	0.65 ± 0.012	0.69 ± 0.1	4.3 ± 3.2
3D U-net [45]	0.72 ± 0.013	0.75 ± 0.03	3.02 ± 3.0
DenseVoxNet [49]	0.783 ± 0.015	0.81 ± 0.02	2.7 ± 2.9
3D MSDenseNet (Proposed Method)	**0.84 ± 0.02**	**0.85 ± 0.02**	2.6 ± 2.8
3D FCNNs + 3D Level Set [48]	0.76 ± 0.02	0.79 ± 0.02	3.0 ± 3.0
3D U-net + 3D Level Set	0.82 ± 0.02	0.84 ± 0.02	2.9 ± 2.7
DenseVoxNet + 3D Level Set	0.83 ± 0.012	0.84 ± 0.02	**2.52 ± 2.8**
3D MSDenseNet + 3D Level Set (Proposed Method)	**0.86 ± 0.02**	**0.87 ± 0.02**	2.54 ± 2.4

2.7), respectively. While 3D MSDenseNet + 3D Level Set has got progress in DSC and recall as 2.13% and 2.42%, separately, and it reduces ASD to 2.54 from 2.64. Nevertheless, the 3D MSDenseNet + 3D Level Set method could not attain a significant improvement by utilizing the post-processing step but still outperforms than all. Taking into account both qualitative and quantitative results, it has been observed that the addition of the 3D level set as a post-processor with each method improves their segmentation results.

6.5 Discussion and Conclusion

In this work, we tested our initially proposed method 3D FCNNs + 3D Level Set [48], and two prominent and widely explored volumetric segmentation algorithms, namely 3D U-net [45], and 3D DenseVoxNet [49] for volumetric segmentation of colorectal tumor from T2-weight abdominal MRI. Furthermore, we extended their ability by incorporating 3D level set in their original implementations for the colorectal tumor segmentation task. Based on their pros and cons, we proposed a novel algorithm; 3D Multiscale Densely connected neural network (3D-MSDenseNet). In medical image segmentation, there are many studies have been carried out for developing several techniques, mostly based on geometrical methods to address the hurdles and challenges

in the chaotic medical image segmentation, including statistical shape models, graph cuts, level set and so on [117]. Recently, level-set based segmentation algorithms are commonly explored approaches for medical image segmentation. Generally, level set based algorithms utilize the energy minimization problem by incorporating different regularization (smoothing terms) and priors depending on segmentation tasks. The level set based segmentation algorithms can vary topological properties of segmentation function [118], which makes the level set more preferable in segmentation problems. However, they always require an initial appropriate contour initialization to segment the desired object. This initial contour initialization requires expert user intervention in the medical image segmentation. Also, medical imaging has disordered intensity distribution and also varies from one imaging modality to another, and even varies slice to slice in a volume of the same modality. Therefore, their segmentation is impeded to generalize using these traditional approaches based on statistical models of the intensity distribution. In other words, level-set based approaches are progressively deficient due to their simple appearance model [100], which may limit its transferability and generalization capability to learn the chaotic intensity distribution in medical images. Currently, convolution neural networks (CNNs) based deep learning methods have been successfully employed in medical imaging, especially for image classification, detection and segmentation purposes. Usually, the deep learning based approaches learn a model by extracting features deeply from intricate structures and patterns from well-defined large training data sets where the trained model is used for prediction. Unlike level-set based methods, deep learning can learn appearance models automatically from the extensive training data, which improves its transferability and generalization ability. Nevertheless, deep learning based approaches are not able to provide an explicit way of integrating prior shape and regularization. Therefore, considering the merits and demerits of both level set and deep learning contrariwise, we incorporated 3D level set in each method that we used for our task.

Moreover, traditional CNNs are 2D in nature and were designed primarily for 2D natural images. Whereas, medical images like MRI or CT are in 3D form. Generally, these 2D CNNs with 2D kernels have been used for medical image segmentation where volumetric segmentation was performed in a slice by slice sequentially. These 2D kernels are not able to use volumetric spatial information completely by sharing spatial

information among the three planes simultaneously. Taking into account this problem with tradition CNN, 3D U-net and DenseVoxNet provide a 3D CNN architecture, which utilizes 3D kernel which simultaneously shares spatial information among three planes.

Another challenge with 3D CNN is controlling the network optimization when the network goes deeper. Deeper networks are more prone to get the risk of over-fitting due to the vanishing of gradients in advance layers. This has been proven in this work. From the segmentation results produced by 3D FCNNs, someone can see in figure 6.2 that how the patterns/gradients have been lessened. In order to preserve the gradients in the next layers when the network goes deeper, 3D U-net and DenseVoxNet reused the features from early to the next layer. In this way, 3D U-net overcomes this vanishing gradient problem in a deep network by incorporating skip-connection, which propagates output features from layers of the same resolution in the contraction path to the output features from the layers in the expansion path.

Nevertheless, this skip-connection allows the gradient to flow directly from the low-resolution path to high-resolution path, which makes training easy, but this generally produces large feature channels in every layer and lead network to adjust a large number of parameters during training. To overcome this problem, the DenseVoxNet extends the concept of skip connections by constructing a direct connection from every layer to its corresponding previous layers to ensure maximum gradient flow between layers. In simple words, feature maps produced by the preceding layer were concatenated as an input to the advanced layer, thus providing a direct connection from any layer to the subsequent layer. Our results have proven that direct connection strategy by DenseVoxNet provides better segmentation than skip connection strategy by 3D U-net. However, DenseVoxNet has a deficit as network learns high-level feature and low-level features in early and later layers, individually, which limit the network to learn multi-scale contextual information throughout the network and may lead the network in poor performance. Our proposed network provides a multi-scale dense training scheme where high resolution and low-resolution features are learned simultaneously, thus maintaining maximum gradients throughout the network. Our experimental analysis reveals that reusing of features through multi-scale dense connectivity produces effective colorectal tumor segmentation.

Nevertheless, our proposed method obtained better performance in the colorectal tumor segmentation, but it has a limitation too. As it is seen from Table 6.1 that our proposed algorithm has high standard deviations in DSC and recall as compared to other methods. It shows that the proposed method is not able to confront with a variation of contrast in the cancerous region and variation of the slice gap along z-axis among data sets. The better normalization and super-resolution method may be required to solve this problem, and more training samples might be helpful.

CHAPTER 7

CLASSIFICATION OF RESPONDERS AND NON-RESPONDERS TUMORS – EXPERIMENTAL RESULTS

This chapter presents experimental results related to our third contribution; assessment of predictive model using radiomics features in three different cases; Case 1: Predictive model using only handcrafted radiomics features, Case 2: Predictive model using only deep radiomic features, and Case 3: Predictive model using a combination of handcrafted and deep features, respectively. Our study focuses on analyzing the above three cases to obtain an optimal radiomics based biomarker in tumor response to colorectal therapy. Since high accuracy, efficiency and reliability are crucial factors in the obtained predictive and prognostic models, which totally depend on the success of radiomics based clinical biomarkers. Thus, in order to examine the effectiveness of radiomics based features in obtaining an accurate predictive model; it is necessary to validate and compare different machine learning models utilizing all possible radiomics features. For this purpose, in this thesis, the most widely explored supervised machine learning based classifiers were employed. In addition, radiomics have high space dimensionality problem like any high-throughput data-mining field. In this regard, we have assessed the performance of six different feature selection algorithms, which can improve the performance of radiomics based predictive models in different ways.

7.1 Material and Methods

In this work, we have assessed seven classification methods and six feature selection methods for radiomics based prediction of tumor response to neoadjuvant chemoradiotherapy (CRT) in colorectal cancer. In our study, a selection of these methods was based on their popularity and wide exploration in the literature.

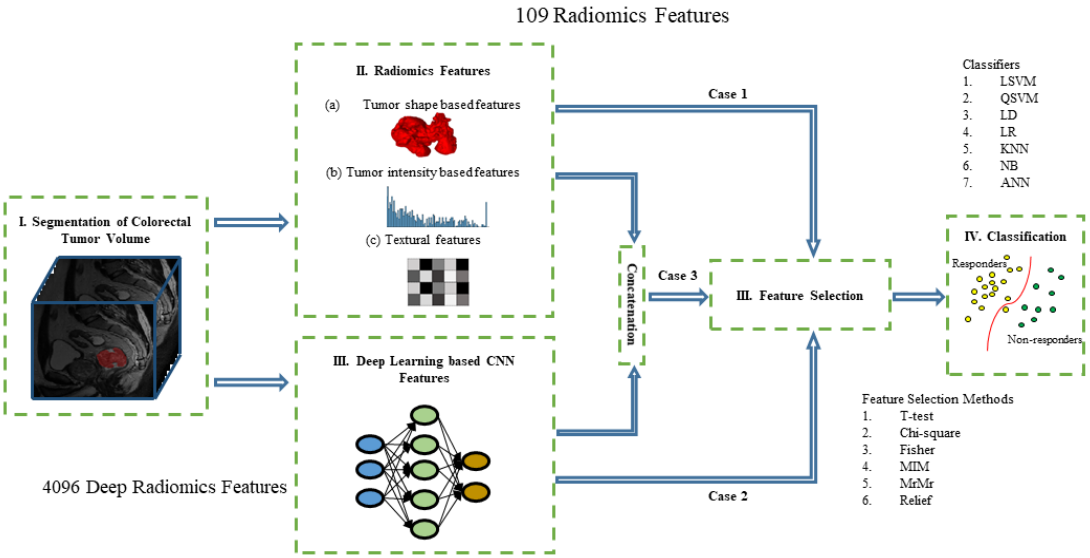


Figure 7.1: Proposed methodology for classification of responders and non-responders

Additionally, their implementation is publicly available online (http://www.scikit-learn.org), we have set configuration in these methods with their general and widely testified hyper-parameters [139], thus by following unbiased assessment for these approaches. A detail description of each classifier and feature selection algorithm is given in previous chapter 3.

An overview of the proposed methodology is depicted in figure 7.1. We used 3D MRI data acquired from two different hospitals (Department of Radiological Sciences, University of Pisa, Via Savi 10, 56126 Pisa, Italy and Department of Radiological Sciences, Oncology and Pathology, University La Sapienza, AOU Sant'Andrea, Via di Grottarossa 1035, 00189 Rome, Italy) along with their manual segmentation of tumor volumes. A detail description of data sets is discussed in section 7.1.1. Two different types of radiomics features were extracted from our data: traditional handcrafted radiomics features and deep radiomics features. A total of 109 handcrafted radiomic features were extracted from each MRI volume in this study. These handcrafted radiomics features are divided into further three groups, including 1) Tumor shape based, 2) Tumor intensity based, and 3) Textural features. The tumor shape-based features are referring to 3D shape representation of the tumor. The intensity-based features are first-order textural features where first-order statistical distribution of the voxel intensities within the tumor are calculated. The textural features are second-order statistical textural features where the second-order spatial distribution of the voxel

intensities is computed. These handcrafted radiomics features are explained in section 7.1.2. Furthermore, 4096 deep radiomics features for each patient are computed using transfer learning from the pre-trained convolutional neural network (CNN_S), as explained in section 7.1.3.

Consequently, this study examines a predictive model using radiomics features in three different cases as shown in figure 7.1; *Case* 1: Predictive model using only handcrafted features, *Case* 2: Predictive model using only deep radiomics features, and *Case* 3: Predictive model using a combination of handcrafted and deep radiomics features, respectively. A detail experimental analysis using the above three cases is mentioned in section 7.2. Considering radiomics features using any case from the above three cases; feature selection and predictive modeling can produce an optimal radiomics based biomarker in clinical oncology.

7.1.1 Experimental Data sets

Study Population: The retrospective study has involved 43 patients, 27 males, and 16 females. All patients have locally advanced rectal cancer, with an average age of diagnosis of 67 (52 – 81 years old range).

The inclusion criteria were:

- Rectal cancer diagnosis;
- Local advanced rectal cancer: T3, T4 or any T with N+;
- Patients undergoing preoperative radiochemotherapy;
- Anatomopathological examination available after surgery;
- MRI both pre- and post-radiochemotherapy and good quality imaging;
- The absence of other neoplastic diseases

Every patient underwent to staging analysis before radiochemotherapy, which includes digital rectal examination, blood tests, chest X-ray examination, and colonoscopy, abdominal computed tomography with contrast, endorectal ultrasound and magnetic resonance imaging with phased-array surface coils.

MRI acquired data: The overall data set consisted of 43 volumes T2-weighted MRI and each MRI volume consists of several slices which are varied in number across subjects as 69 ~122 and have dimensions as 512 × 512 × (69 ~122). Their voxel spacing

also varying from 0.46 × 0.46 × 0.5 to 0.6 × 0.6 × 1.2 mm/voxel across each subject. As the data have a slight slice gap, thus we did not incorporate any spatial resampling.

Anatomopathological Examination: The surgery samples have been evaluated by an anatomopathologist both macro- and microscopically after fixed with formalin and stained mesorectal fascia with ink. The microscopic examination has been performed with hematoxylin and eosin stain. All the suspect areas have been accurately observed, including the mesorectal and inferior mesenteric lymph nodes. In some cases, particularly difficult to evaluate, immunohistochemical analysis has been performed too.

Every histological report described:

- Histotype
- Tumor grade: G1 if the tumor is well differentiated, G2 if the tumor is moderately differentiated, G3 if the tumor is poorly differentiated, G4 if the tumor is undifferentiated
- Tumor invasion limited to the rectal wall, beyond the rectal wall into mesorectum, etc.
- Neoplastic growth pattern
- Vascular invasion
- Perineural invasion
- Tumor budding (the presence of tiny detached clusters and cords of tumor cells embedded in desmoplastic stroma at the leading edge of the invasive front of the tumor): absent/present
- Peritumoral lymphocyte infiltration
- Intratumoral lymphocyte infiltration
- Surgical margin status: absent/present tumor invasion;
- Number of lymph nodes involved
- Quirke's graded assessment of completeness of mesorectal excision: 1-poor, 2-moderate, 3-good;
- Circumferential Resection Margin (CRM), the distance between tumor cells and CRM, it is considered positive if the distance is less than 1 mm
- ypTNM according to American Joint Committee on Cancer

- TRG of Dworak: TRG0 = no regression, TRG1 = Dominant tumor with fibrosis and/or vasculopathy, TRG2 = Significant fibrosis with groups of tumor cells (easy to find), TRG3 = Dominant fibrosis or mucin with very few tumor cells (difficult to find microscopically), TRG4 = No tumor cells, only fibrotic mass (total regression)

There are many different TRG score systems, and actually, a gold standard has not been defined. Each system classifies specimens according to the increasing or decreasing percentage of fibrosis in three to five groups. Generally, a percentage of fibrosis greater than 85% is representative of a complete response that is TRG 4 [147]. A higher overall survival, as well as disease-free survival, has been demonstrated in complete responders [148-149]. Particularly, we have considered this last parameter in order to divide patients into two different categories: patients with a response TRG4 are being considered complete responders, the other one (with a chemoradiotherapy response TRG0, TRG1, TRG2, and TRG3) are being considered partial /non-responders. Consequently, among 43 patients, we have 23 patients observed as complete responders, and 20 observed as non-responders.

7.1.2 Handcrafted Radiomics Features

Handcrafted features were extracted from the region of interest (ROI) in three different groups: first-order histogram based features, shaped based, and second order textural features. These features were extracted in python using publicly online pyradiomics software (https://pyradiomics.readthedocs.io/en/latest/), [28]. A detail of these features is given in Appendix A.

7.1.3 Deep Radiomics Features

Recently, deep learning [83] has arisen as a successful and widely explored methodology in computer vision and attained many breakthroughs and state-of-the-art performance in various computer vision applications, including image classification and recognition [84-85], semantic segmentation [86-88], stereo matching [89], object detection [90-91], etc. Deep learning is also known as deep convolutional neural

Table 7.1: CNN_S Architecture [141]

Conv1	96 × 7 × 7, stride of 2, padding 0, ×3 max pooling, LRN
Conv2	256 × 5 × 5, stride of 1, padding 1, ×2 max pooling
Conv3	512 × 5 × 5, stride of 1, padding 1
Conv4	512 × 5 × 5, stride of 1, padding 1
Conv5	512 × 5 × 5, stride of 1, padding 1, ×3 max pooling
Full 6	4096, dropout
Full7	4096, dropout
Full8	1000 softmax

networks, abbreviated as CNNs or ConvNets. We have already explained the details of the advantages and basic building blocks of CNN in previous chapter 4. Due to the fact, the performance of deep learning based algorithms is dependent on data density. As large data sets improve the performance of deep learning algorithms. Nonetheless, in contrast to natural images, the data sets in the medical imaging domain are insufficient to obtain an effective performance in the deep learning approaches.

Despite this fact, transfer learning is frequently employed in computer vision to tackle with a small data set problem [143]. In the machine-learning domain, transfer learning is a way where previously learned knowledge from one domain, is applied to a new but different domain in a similar way to the previous domain [140]. More precisely, the new problem is solved by previously learned model or experience. Transfer learning can be employed in the radiomics field for extraction of abundance deep radiomics features from the hidden layers of CNN. These deep features decode more nonconcrete information from medical images. These deep features may deliver more prognostic patterns than traditional handcrafted radiomics features. According to our best knowledge through literature review, there is some work at small-scale has been proposed where deep radiomics features are evaluated in comparison with traditional handcrafted radiomics features [26, 35].

In our work, we extracted deep features via transfer learning by applying the pre-trained CNN_S model [141] on our data in forward propagation only. The CNN_S was

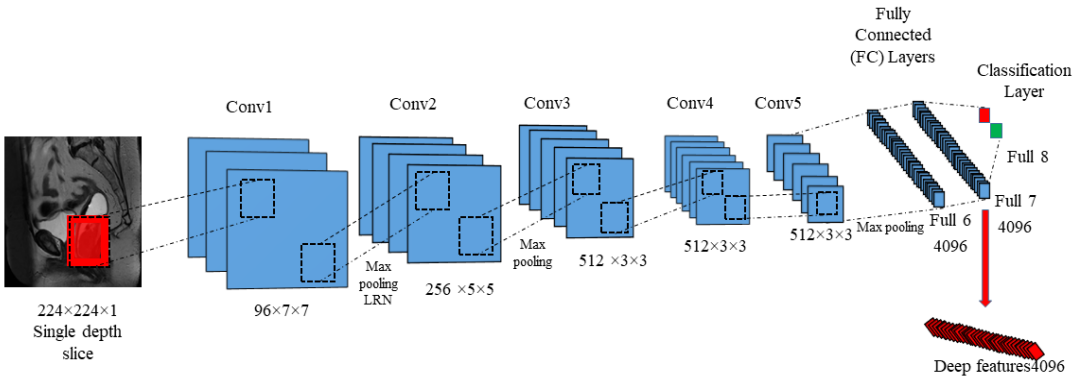


Figure 7.2: Extraction of deep radiomics features using pertained model of CNN_S.

trained on large ImageNet Large Scale Visual Recognition Challenge dataset 2012 (ILSVRC2012 [145]). ILSVRC database is comprised of more than fourteen million

images and more than thousands of object cliques. On the other hand, we have a comparatively a small number of medical data sets (i.e., 43 patients) that is insufficient to train such a CNN, which can learn millions of weights. The CNN_S architecture consists of total eight layers: five are convolution (Conv) layers, and two are fully connected (FC), and the last one is a softmax classification layer, as mentioned in Table 7.1 and figure 7.2. We provide segmented tumor as an input to CNN_S. For this purpose, we have selected out the slice from each segmented volume, which contains the tumor with the biggest area. As CNN_S requires an input of size 224 × 224 × 3 where 3 is channel depth, i.e., RGB. However, MR images are grayscale images; therefore, we modified the code by using only one channel, red (R), while keeping others off. Secondly, CNN_S requires normalized data; subsequently, the voxel intensities of MR volumes were normalized to the range [0 255]. Thirdly, the size of the segmented tumor varies in the range of average width and height, ~ 45 to 33 approximately, pixels concerning the actual resolution of the MRI scan. Therefore, we use a window of an approximate size of tumor size (i.e., if the segmented tumor has size 36 × 42, the window size should be 35 × 35, so that window covers all pixels related to the tumor). In this way, we cropped the tumor with its corresponding window size. In order to achieve the CNN_S input size requirement that is 224 × 224, the cropped tumor resized accordingly to 224 × 224 by applying bi-cubic interpolation. Finally, we extracted deep features from the second last layer (i.e., fully connected layer 7), as shown in figure 7.2. In a result, we have in total 4096 deep features. This experiment has been performed in a MATLAB toolbox, MatConvNet, [142].

7.1.4 Statistical Analysis

A. Evaluating predictive and prognostic performance of feature selection algorithms and classifiers

In this study, we used filtered based feature selection algorithms, namely T-test (T-score), Chi-square (Chi-score), Fisher, Mutual Information Maximization (MIM), Minimum Redundancy Maximum Relevance (MrMr) and Relief. We used supervised machine learning based classifiers, namely Support Vector Machine (SVM) with linear kernel (LSVM), SVM with kernel 'poly' (QSVM), Logistic Regression (LR), Linear Discriminant (LD), Naïve Bayes (NB), K-Nearest Neighbor (KNN), Artificial Neural Network (ANN). The training of these classifiers was carried out on by 100 repetitions where the data were randomly divided into 75% training data and 25% test data in each repetition. The trained model was evaluated on split test data in term of area under ROC curve (AUC).

The best size of selected features is very important to assess and compare each classifier performance with respect to each feature selection algorithm. In this regard, considering three factors, size of selected features, feature selection algorithm, and classifier model; a three-dimensional parameter search grid is formed. Thus, we incremented a number of selected features for each feature selection algorithm with an incremental factor of 5 till 25; like in different subsets, i.e., 5, 10, 15, 20, and 25. Consequently, the performance of each classifier is assessed using each subset in terms of area under ROC curves (AUC). As 100 repetitions were used; therefore, we took the mean of 100 AUCs for each classifier with respect to each subset of every feature selection algorithm.

B. Finding the potential size of the selected features and the effect of classifier performance regarding the feature selection algorithm

In order to find the best size of selected features, and their effect corresponding to its feature selection algorithm on classifier performance; we use multi-factor ANOVA on computed AUCs. First, we compare the different size of selected features of all feature selection algorithms to find the best size of selected features, which gives the

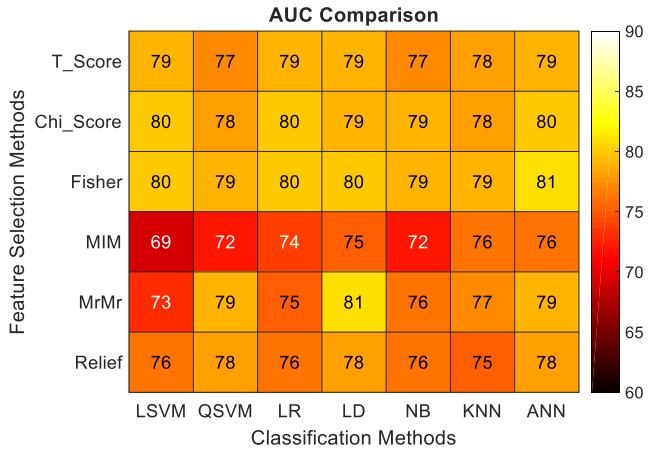
highest AUC. Then we compare interactions between classifiers and feature selection algorithms, to get the best combination of feature selection and classification algorithms.

7.2 Experimental Results

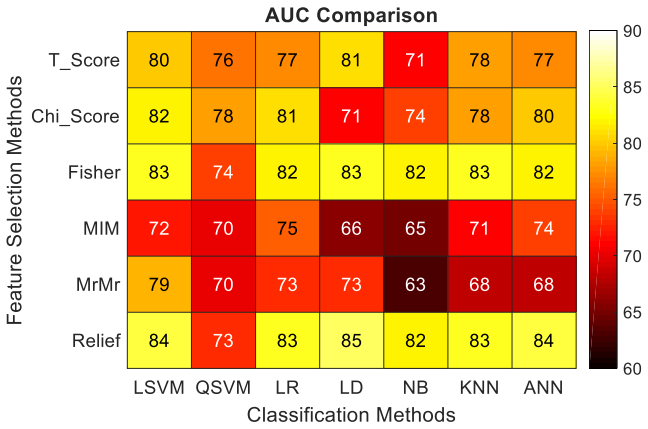
In order to assess the performance of classifiers utilizing predictive radiomics features as prognostic biomarkers, we extracted features from our data in different three cases. Such as, a total of 109 handcrafted radiomics features were extracted in *case 1*; in *case 2*, a total of 4096 deep radiomics features extracted via transfer learning and; in *case 3*, we analyzed predictive model using a combination of handcrafted radiomics features and deep radiomics features. In *case 3*, these features were combined by concatenating the selected features from each case.

7.2.1 Case 1: Analysis with only Handcrafted Radiomics Features

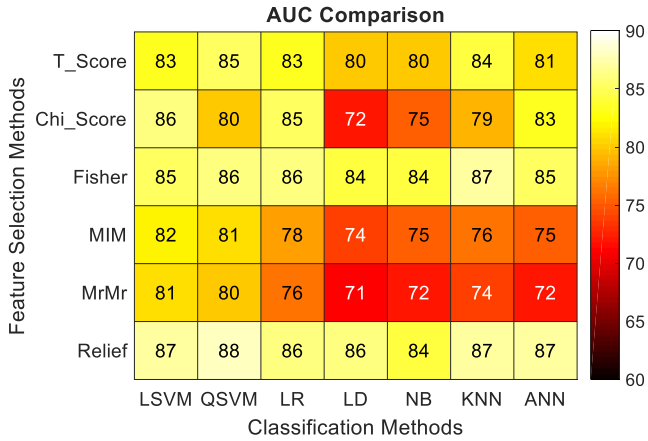
In this study, we assessed six feature selection algorithms and seven classification methods in term of AUC by utilizing handcrafted features as the predictive biomarkers. Figure 7.3 (a), gives a heatmap representation of obtained mean AUCs (in %) by different feature selection algorithms, depicted in rows; and different classification algorithms, depicted in columns. Accordingly, each feature selection algorithm contains seven mean AUC values (in %) with respect to seven different classifiers; similarly, each classifier contains six different mean AUC values (in %) with respect to six different feature selection algorithms. Furthermore, for more simplification, their obtained AUCs (mean ± Std) are depicted in Table 7.2, and Table 7.3. Table 7.2 presents the performance of each classification method in term of AUC (mean ± Std) with respect to all features selection for each case, and Table 7.3 presents the performance of each feature selection in term of AUC (mean ± Std) with respect to all classification method for each case. For case 1, Table 7.2 demonstrates that classification methods, ANN and LD stood first by obtaining the best mean AUC values (mean ± Std), as 0.79 ± 0.02 and 0.79 ± 0.023, individually. Whereas, NB and LSVM stood the last by obtaining mean AUCs, as 0.76 ± 0.025 and 0.76 ± 0.44, respectively.



(a) Case 1: Analysis with only Handcrafted Radiomics Features



(b) Case 2: Analysis with only Deep Radiomics Features



(c) Case 3: Analysis with Combination of both Handcrafted and Deep Radiomics Features

Figure 7.3: Heatmap representing the mean AUCs (in %) for each case with size of selected features = 5; feature selection algorithms (in rows) and, in columns for classification methods (in columns).

Table 7.2: Mean values of AUCs obtained by different classifier in each case.

Algorithms	Case 1	Case 2	Case 3
LSVM	0.76 ± 0.44	**0.8 ± 0.042**	**0.84 ± 0.025**
QSVM	0.77 ± 0.024	0.73 ± 0.03	0.83 ± 0.035
LR	0.77 ± 0.025	0.79 ± 0.04	0.82 ± 0.044
LD	0.79 ± 0.023	0.76 ± 0.07	0.78 ± 0.064
NB	0.76 ± 0.025	0.73 ± 0.08	0.78 ± 0.051
KNN	0.77 ± 0.014	0.77 ± 0.06	0.81 ± 0.055
ANN	**0.79 ± 0.016**	0.77 ± 0.06	0.81 ± 0.06

Table 7.3: Mean values of AUCs obtained by different feature selection methods in each case.

Feature Selection Algorithms	AUC (mean ± Std)		
	Case 1	Case 2	Case 3
T_Score	0.78 ± 0.01	0.77 ± 0.03	0.82 ± 0.018
Chi_Score	0.79 ± 0.012	0.77 ± 0.04	0.80 ± 0.05
Fisher	**0.8 ± 0.01**	0.81 ± 0.03	0.85 ± 0.012
MIM	0.74 ± 0.02	0.7 ± 0.04	0.77 ± 0.031
MrMr	0.77 ± 0.3	0.71 ± 0.05	0.75 ± 0.04
Relief	0.77 ± 0.01	**0.82 ± 0.04**	**0.87 ± 0.013**

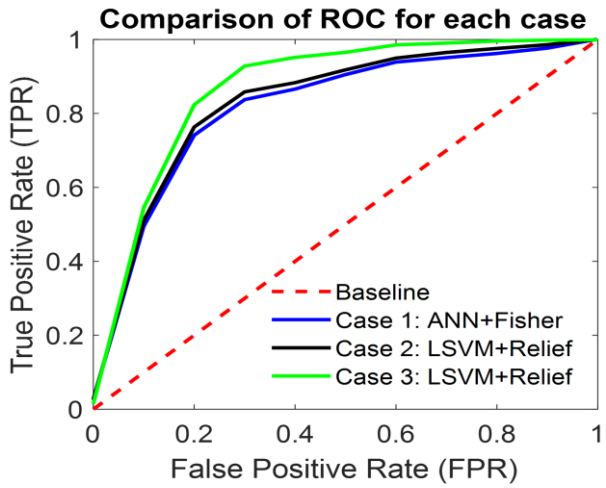
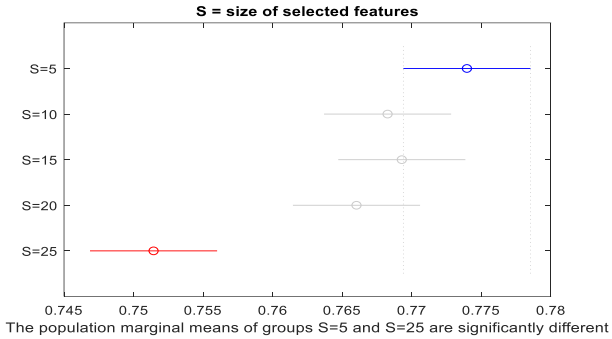
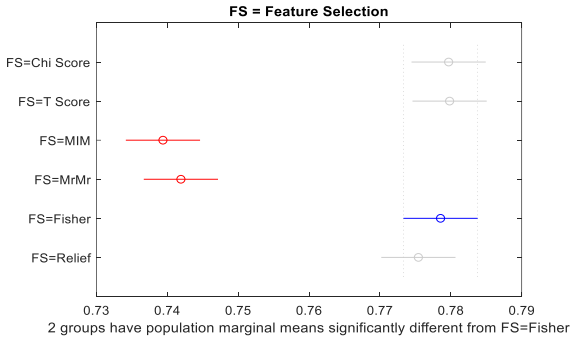


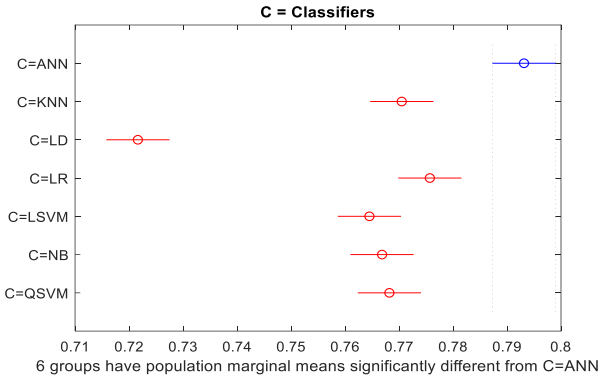
Figure 7.4: Comparison of ROC for the best combination of classifier and FS for each case.


S = size of selected features
S=5
S=10
S=15
S=20
S=25
0.745
0.75
0.755
0.76
0.765
0.77
0.775
0.78
The population marginal means of groups S=5 and S=25 are significantly different

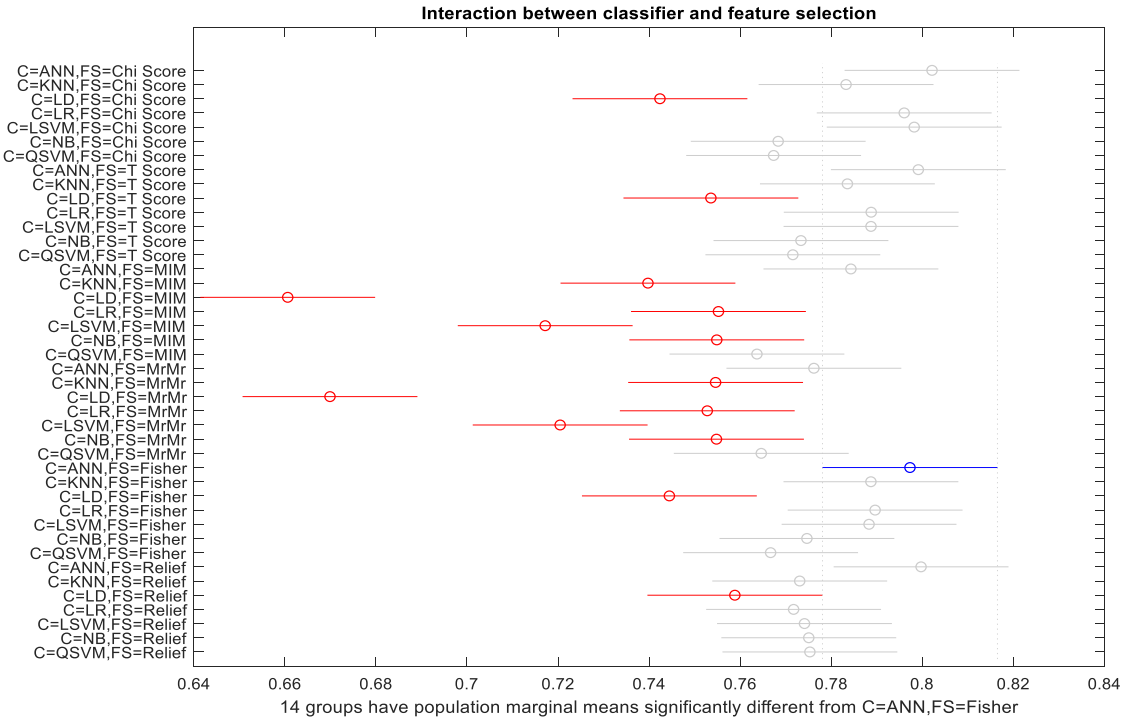
(a)


FS = Feature Selection
FS=Chi Score
FS=T Score
FS=MIM
FS=MrMr
FS=Fisher
FS=Relief
0.73
0.74
0.75
0.76
0.77
0.78
0.79
2 groups have population marginal means significantly different from FS=Fisher

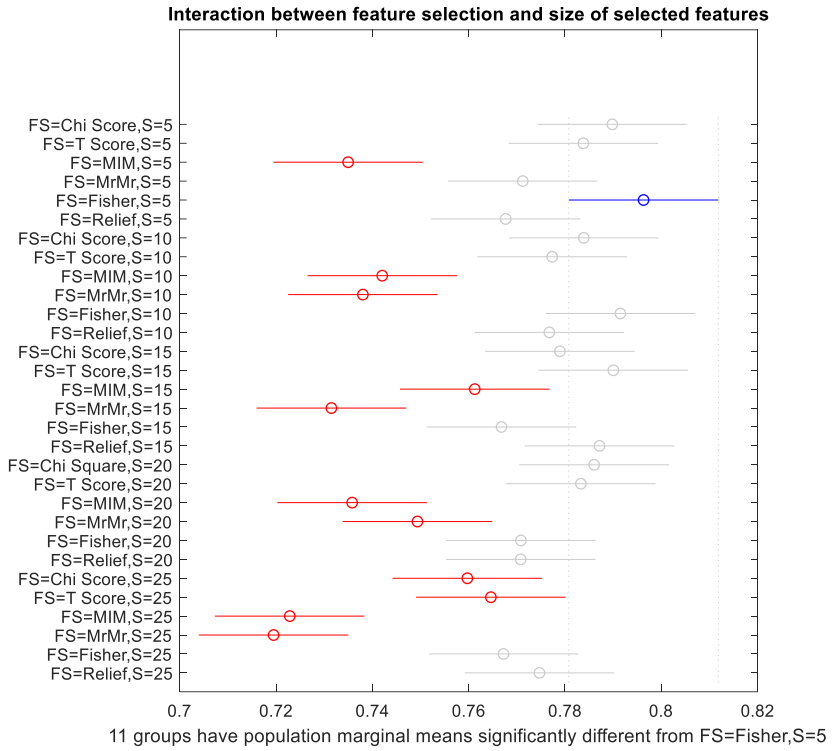
(b)


C = Classifiers
C=ANN
C=KNN
C=LD
C=LR
C=LSVM
C=NB
C=QSVM
0.71
0.72
0.73
0.74
0.75
0.76
0.77
0.78
0.79
0.8
6 groups have population marginal means significantly different from C=ANN

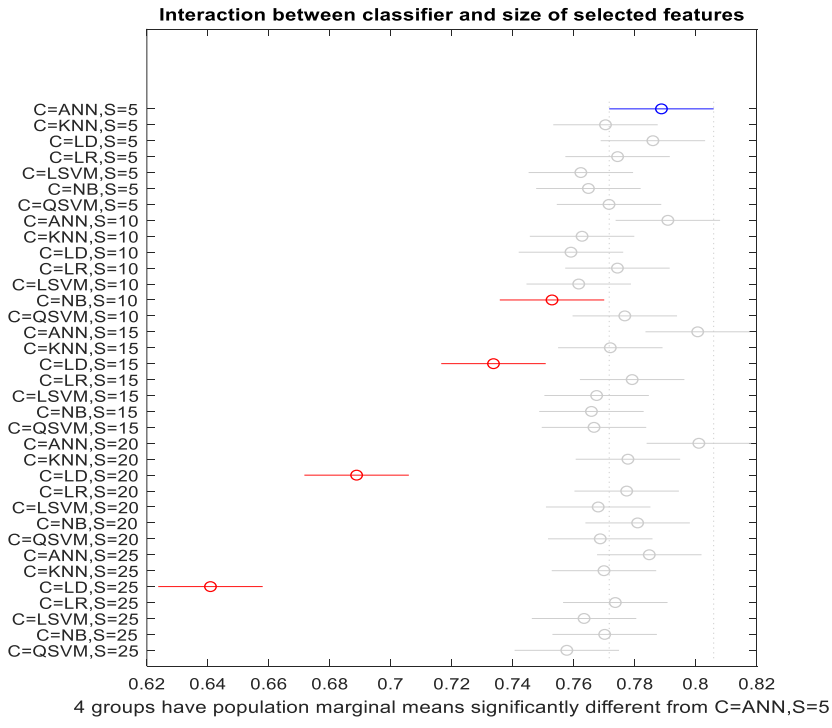
(c)


Interaction between classifier and feature selection
C=ANN,FS=Chi Score
C=KNN,FS=Chi Score
C=LD,FS=Chi Score
C=LR,FS=Chi Score
C=LSVM,FS=Chi Score
C=NB,FS=Chi Score
C=QSVM,FS=Chi Score
C=ANN,FS=T Score
C=KNN,FS=T Score
C=LD,FS=T Score
C=LR,FS=T Score
C=LSVM,FS=T Score
C=NB,FS=T Score
C=QSVM,FS=T Score
C=ANN,FS=MIM
C=KNN,FS=MIM
C=LD,FS=MIM
C=LR,FS=MIM
C=LSVM,FS=MIM
C=NB,FS=MIM
C=QSVM,FS=MIM
C=ANN,FS=MrMr
C=KNN,FS=MrMr
C=LD,FS=MrMr
C=LR,FS=MrMr
C=LSVM,FS=MrMr
C=NB,FS=MrMr
C=QSVM,FS=MrMr
C=ANN,FS=Fisher
C=KNN,FS=Fisher
C=LD,FS=Fisher
C=LR,FS=Fisher
C=LSVM,FS=Fisher
C=NB,FS=Fisher
C=QSVM,FS=Fisher
C=ANN,FS=Relief
C=KNN,FS=Relief
C=LD,FS=Relief
C=LR,FS=Relief
C=LSVM,FS=Relief
C=NB,FS=Relief
C=QSVM,FS=Relief
0.64
0.66
0.68
0.7
0.72
0.74
0.76
0.78
0.8
0.82
0.84
14 groups have population marginal means significantly different from C=ANN,FS=Fisher

(d)



(e)



(f)

Figure 7.5: Case 1: Significance comparison for the obtained AUCs; (a) comparison among size of selected features (s), (b) comparison among feature selection algorithms (FS), (c) comparison among classification algorithms (C), (d) comparison of interaction between feature selection and classification algorithms, (e) comparison of interaction between feature selection and size of selected features, (f) comparison of interaction between classification algorithms and size of selected features.

In regard to feature selection algorithms, Fisher followed by Chi_Score, and T_Score have the better predictive performance (i.e. AUCs (mean ± Std); 0.8 ± 0.001, 0.79 ± 0.012, and 0.78 ± 0.01 respectively) compared to MrMr followed by Relief and MIM, as they have AUC; 0.77 ± 0.3, 0.77 ± 0.01 and 0.74 ± 0.02, individually. In this study, our focus was to obtain the best predictive model with the most relevant with the minimum rank of selected features. The above results we obtained using the most relevant prognostic features with a rank of 5.

Likewise, we repeated our analysis by increasing the rank of selected features as 10, 15, 20 and 25. Results related to these ranks are presented in Appendix B.

Furthermore, multifactor ANOVA is used to analyze each experimental factor (i.e., size of selected features, classification algorithm and feature selection algorithm) independently as well as their interaction with each other. Figure 7.5, a, b and c show an independent analysis of each experimental factor. Figure 7.5 (a) shows that the size of selected features with top-rank of 5 comparatively produces the better prognostic performance. Similarly, figure 7.5 (b) shows that Chi_Score, T_Score, and Fisher are better featured selection methods than MIM and MrMr. In the case of classification algorithms as an individual comparison, figure 7.4 (c) shows that the ANN has achieved better prognostic performance followed by LR and KNN those have merely same prognostic performance as ANN has. Now considering the interaction of the above experimental factors with each other. Our analysis regarding the interaction between the classification algorithms and feature selection algorithms shows that the ANN classifier with four different feature selection algorithms (*viz*., Chi_Score, T_Score, Fisher and Relief) produces the best prognostic model, as shown in figure 7.5 (d). In case of interaction between classification algorithms and different sizes of selected features, it is noted that all classifiers except the LD perform well with all sizes of selected features; the performance of the LD classifier tends to decrease with increasing the size of selected features, as shown in figure 7.5 (f). Furthermore, it is noteworthy that the ANN has the best performance than all in combination with all size of selected features. Finally, the interaction between the feature selection algorithm and the size of the selected features were analyzed. Figure 7.5 (e) shows that the feature selection

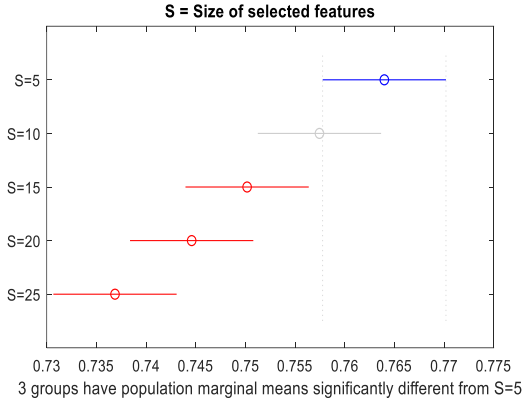
algorithm Fisher has the highest prognostic performance than all; particularly, in combination with the size of selected features with the minimum ranks of 5 and 10.

In case 1, we have found that the combination of the ANN classifier and the Fisher as feature selection algorithm with the minimum (i.e., 5) top-rank of the size of selected features outperforms as being the best predictive model in our study. Those top-five ranked features selected by the Fisher are: *Volume* (shape-based feature), *Energy* (Haralick's Second order GLCM), *Entropy* (Haralick's Second order GLCM), *Long Run Emphasis* (*LRE*) (Second-order GLRLM), *Size Zone Non-Uniformity Normalized* (*SZNN*) (GLSZM), *Cluster shade* (Haralick's Second order GLCM).

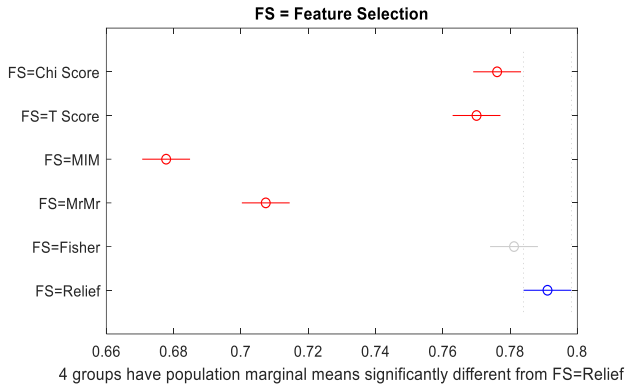
7.2.2 Case 2: Analysis with only Deep Radiomics Features

In case 2, we have extracted a total of 4096 deep radiomics features as explained in Section 7.1.3. In this case, the results were obtained utilizing deep radiomics features as the prognostic biomarkers by following an analogous strategy of statistical analysis as in case 1. Figure 7.3 (b) gives a heatmap representation of attaining mean AUCs (in %) by different future selection algorithms (rows) and different classification algorithms (columns). Like in case 1, their obtained mean AUCs are depicted in Table 7.2 and Table 7.3, respectively. Table 7.2 shows that the classification methods, LSVM, and LR stood first by obtaining the best mean AUC values (mean ± Std), as 0.8 ± 0.042 and 0.79 ± 0.04, respectively. Whereas, the NB stands the last by obtaining the lowest mean AUCs, as 0.73 ± 0.06. Table 7.3 shows feature selection algorithms' performances where Relief and Fisher have achieved better predictive performance (i.e., AUCs (mean ± Std); 0.82 ± 0.04, and 0.81 ± 0.03, respectively) compared to Chi_Score and T_Score (AUC; 0.77 ± 0.04 and 0.77 ± 0.03, respectively). Whereas, MIM and MrMr have the lowest performances, as 0.7 ± 0.04 and 0.71 ± 0.05, correspondingly. Like in case 1, the above results are obtained by selecting the top 5 ranked features from each feature selection algorithm. Similarly, we repeated our analysis by increasing the rank of selected features as 10, 15, 20 and 25. The results related to these selected ranks are presented in Appendix C.

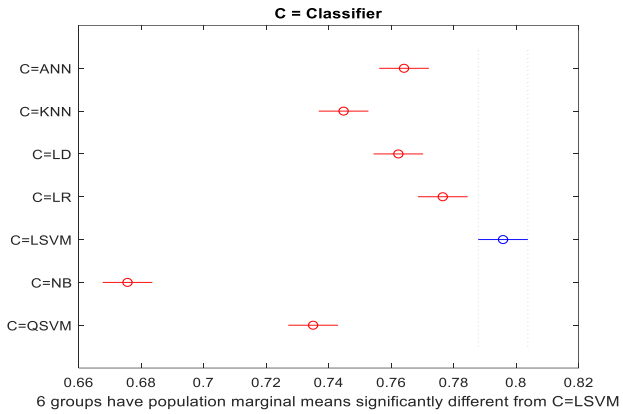
Likewise, multi-factor ANOVA was also used here to analyze the behavior of those three experimental factors (i.e., size of selected features, classifier, and feature selection


S = Size of selected features
S=5
S=10
S=15
S=20
S=25
0.73 0.735 0.74 0.745 0.75 0.755 0.76 0.765 0.77 0.775
3 groups have population marginal means significantly different from S=5

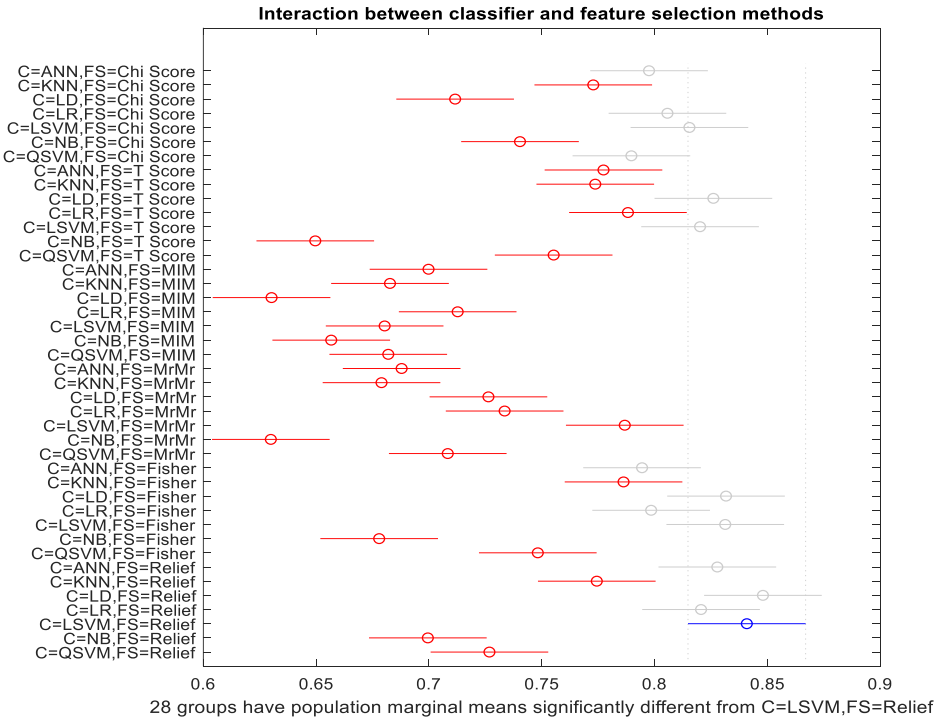
(a)


FS = Feature Selection
FS=Chi Score
FS=T Score
FS=MIM
FS=MrMr
FS=Fisher
FS=Relief
0.66 0.68 0.7 0.72 0.74 0.76 0.78 0.8
4 groups have population marginal means significantly different from FS=Relief

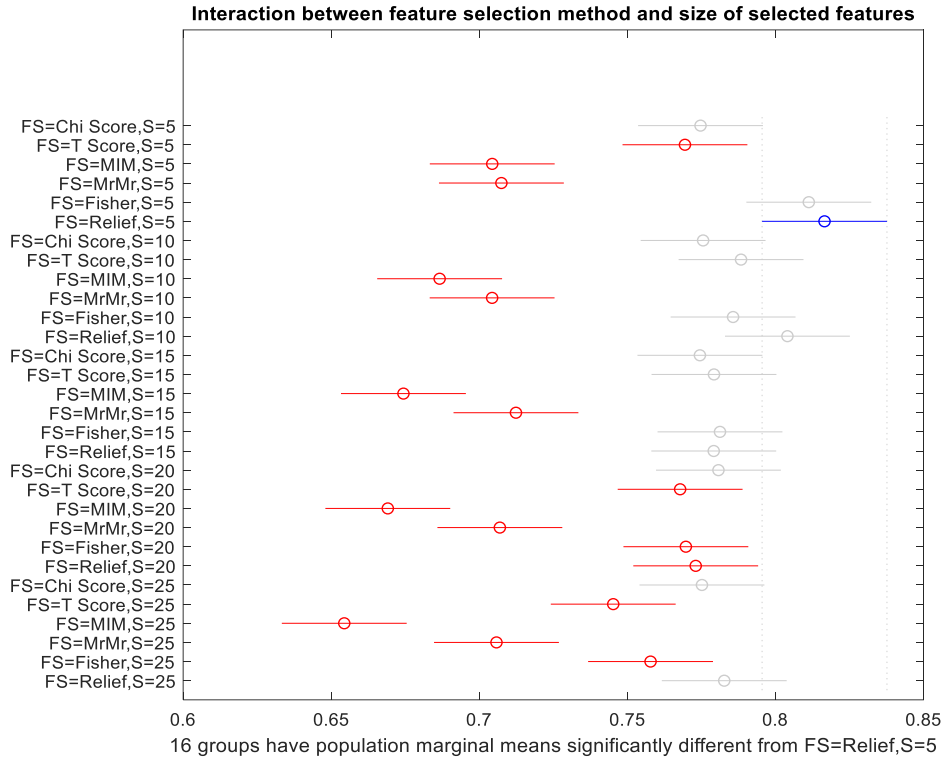
(b)


C = Classifier
C=ANN
C=KNN
C=LD
C=LR
C=LSVM
C=NB
C=QSVM
0.66 0.68 0.7 0.72 0.74 0.76 0.78 0.8 0.82
6 groups have population marginal means significantly different from C=LSVM

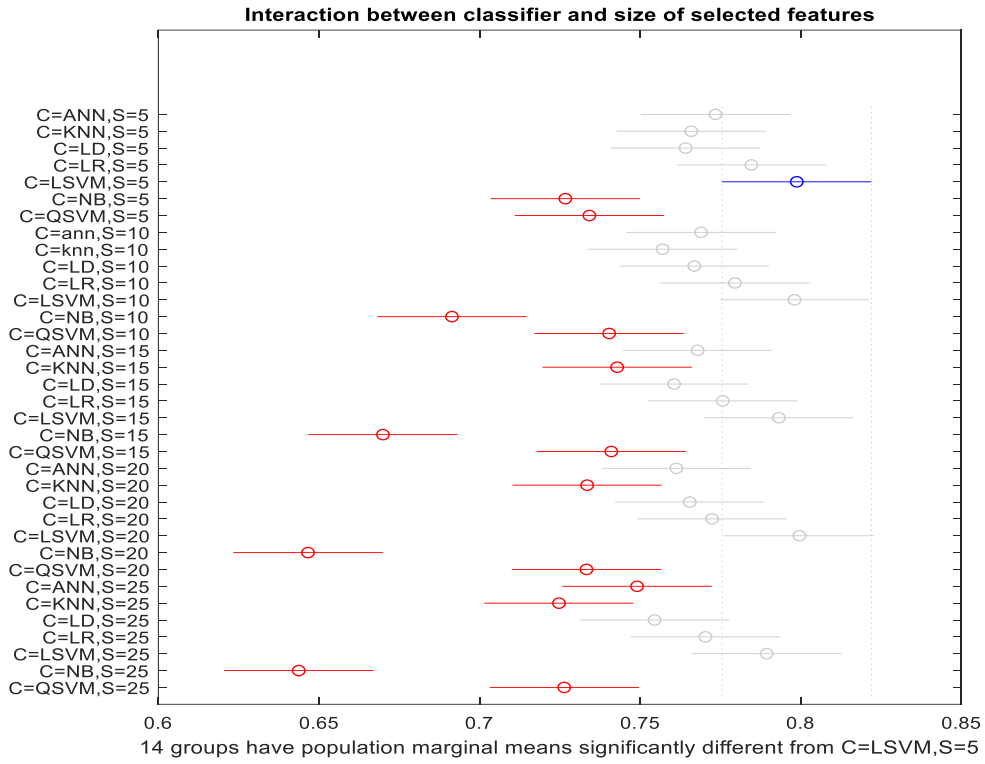
(c)


Interaction between classifier and feature selection methods
C=ANN,FS=Chi Score
C=KNN,FS=Chi Score
C=LD,FS=Chi Score
C=LR,FS=Chi Score
C=LSVM,FS=Chi Score
C=NB,FS=Chi Score
C=QSVM,FS=Chi Score
C=ANN,FS=T Score
C=KNN,FS=T Score
C=LD,FS=T Score
C=LR,FS=T Score
C=LSVM,FS=T Score
C=NB,FS=T Score
C=QSVM,FS=T Score
C=ANN,FS=MIM
C=KNN,FS=MIM
C=LD,FS=MIM
C=LR,FS=MIM
C=LSVM,FS=MIM
C=NB,FS=MIM
C=QSVM,FS=MIM
C=ANN,FS=MrMr
C=KNN,FS=MrMr
C=LD,FS=MrMr
C=LR,FS=MrMr
C=LSVM,FS=MrMr
C=NB,FS=MrMr
C=QSVM,FS=MrMr
C=ANN,FS=Fisher
C=KNN,FS=Fisher
C=LD,FS=Fisher
C=LR,FS=Fisher
C=LSVM,FS=Fisher
C=NB,FS=Fisher
C=QSVM,FS=Fisher
C=ANN,FS=Relief
C=KNN,FS=Relief
C=LD,FS=Relief
C=LR,FS=Relief
C=LSVM,FS=Relief
C=NB,FS=Relief
C=QSVM,FS=Relief
0.6 0.65 0.7 0.75 0.8 0.85 0.9
28 groups have population marginal means significantly different from C=LSVM,FS=Relief

(d)



(e)



(f)

Figure 7.6: Case 2: Significance comparison for obtain AUCs; (a) comparison among size of selected features (s), (b) comparison among feature selection algorithms (FS), (c) comparison among classification algorithms (C), (d) comparison of interaction between feature selection and classification algorithms, (e) comparison of interaction between feature selection and size of selected features, (f) comparison of interaction between classification algorithms and size of selected features.

algorithm). Figures 7.6 (a), (b) and (c), show an individual comparison of each factor and figures 7.6 (d), (e) and (f) show the comparison of their interaction to each other. Similar to case 1, we also found that the size of selected features as 5 top-ranked features produces the best prognostic performance, as shown in figure 7.6 (a). In this case, the LSVM classifier shows the highest prognostic performance than all, as shown in figure 7.8 (b). The Relief based feature selection algorithm has achieved the best performance than other four algorithms (*viz*., Chi_Score, T_Score, MIM, and MrMr) whereas the Fisher has merely the same performance as the Relief has. Considering an interaction between the classification methods and the different rank of selected features, figure 7.6 (f) shows all classifiers behaves nearly and merely same with respect to all ranks of selected features except the NB; the performance of NB decreases as the size of selected feature increases. Figure 7.6 (d) shows that the combination of LSVM with Relief, Fisher and Chi_Score and the combination of LD with Relief, Fisher and T_Score provide a better predictive model. It is clearly shown that the Relief based feature selection algorithm provides a better combination with almost all classifiers to produce the best predictive model. Finally, by analyzing an interaction between the size of selected features and feature selection algorithms; it has been observed that the Relief and the Fisher based algorithms have achieved the best performance with the size of selected features of top-ranked 5.

In conclusion, predicting prognostic model by using deep features; the classifier, LSVM, in combination with Relief or Fisher based feature selection algorithms with the size of selected features as the top-ranked 5, could be used as a good predictive model.

7.2.3 Case 3: Analysis with a Combination of both Handcrafted and Deep Radiomics Features

In the above two cases, it was observed that 5 top-ranked selected features could produce a better prognostic model. Considering this, in this case, we concatenated 5 top-ranked selected features from each case (i.e., case 1 and case 2). Consequently, we similarly analyzed these concatenated features as we did before in case 1 and 2. Similarly, figure 7.3 (c) represents the mean AUCs (in %) obtained by different feature

selection algorithms (column) and classification algorithms (row). Table 7.2 (column 3) shows mean AUCs values of each classifier with respect to all feature selection algorithms and Table 7.3 (column 3) shows mean AUCs values of each feature selection algorithm with respect to all classifiers. From Table 7.2, it can be seen that the performance of all classifiers except the LD, is improved by utilizing the concatenation of these two types of radiomics features. However, there is no significant improvement in LD classifier's performance by doing this. Similarly, a remarkable improvement was noted in the performance of all feature selection algorithms except MrMr. In addition, we also repeated the analysis by concatenating top-ranked features as 10, 15, 20 and 25; their results are depicted in Appendix D.

Table 7.3 reveals that the Relief and the Fisher based feature selection algorithms have achieved the best prognostic performance in combination with all classifiers, as they have mean AUCs (i.e., 0.87 ± 0.013 and 0.85 ± 0.011, respectively). Whereas classifiers, the LSVM followed by QSVM and LR have attained the best prognostic performance, as they obtained mean AUCs (0.84 ± 0.025, 0.83 ± 0.035, and 0.82 ± 0.045, respectively). In conclusion, the best performance in term of mean AUC with respect to all feature selection methods has been achieved by the LSVM and similarly among feature selection methods; the Relief has attained the best performance in term of mean AUC with respect to all classification methods. Those top-five ranked features selected by the Relief are *Skewness* (First-order statistic feature), *Sphericity* (shape-based feature), *Energy* (Haralick's Second order GLCM), *Entropy* (Haralick's Second order GLCM), and *Cluster shade* (Haralick's Second order GLCM).

Considering the above three cases with the best size of selected features (i.e., top-ranked 5), the best prognostic model corresponding to case 1, case and case 3 is produced by the combination of ANN + Fisher, LSVM + Relief, and LSVM + Relief, respectively. Their comparison in term of ROC curves obtained as a mean over 100 runs, is depicted in figure 7.4. Figure 7.4, shows that the best ROC curve is attained in case 3 while the other two cases have nearly the same ROC curves.

7.3 Discussion and Conclusion

Early diagnosis and accurate staging of colorectal cancer is very crucial in the oncologic patients' management, predominantly in personalized treatment strategies. Currently, magnetic resonance imaging (MRI) is the most widely explored and preferable imaging modality in the loco-regional staging of colorectal cancer [4-5]. Generally, medical diagnosis of suspected cancer is carried out in terms of different medical tests like a biopsy or medical diagnostic imaging. Nevertheless, the biopsy can provide an informed diagnosis, but it is an invasive diagnostic technique and may not provide heterogeneity of the tumor entirely, which is essential in the evaluation of response to therapy in colorectal chemoradiotherapy (CRT). On the contrary, the diagnostic imaging such as MRI and computed tomography (CT) those are non-invasive diagnostic techniques and can provide essential information related tumor's characteristics, such as, tumor size and its overall shape, tumor heterogeneity and tumor growth over time; these advantages of the medical diagnostic imaging techniques make them more preferable than the biopsy. However, the role of the medical diagnostic imaging in the prognosis of suspected cancer is challenging where the radiologist with high expertise is required to locate/detect the suspected cancer in a large data set, which is a time-consuming process too.

Nowadays, radiomics [34-38], semiautomatic/automatic quantitative diagnostic technique that decodes the encoded information in large medical imaging datasets, quantitatively. Radiomics measure tumor heterogeneity for diagnosis of several cancer types non-invasively, thus by providing a prognostic or predictive model. Several studies have been carried out to create Radiomics based prognostic model for different clinical issues such as patient survival outcome [25, 38], treatment response [17-18], tumor grading [26-28], and more [34-37]. In our study, an accurate diagnosis and staging of colorectal cancer at early basis is the supreme interest where medical experts may decide the treatment plan that a patient should go for either therapy or surgical operation. In literature, multiple radiomics based features have been incorporated for different purposes; therefore, it is difficult to say that what radiomics features are useful in the assessment of colorectal cancer. Thus, the goal of this work is to find which of the radiomics feature are the most appropriate in the prediction of complete tumor response to neoadjuvant therapy and to assess the possible correlation among these

features. In this regard, we have extracted two types of radiomics features; handcrafted features (traditional features) and deep Radiomics features via transfer learning, inspired by [38]. For effective insight analyses of radiomics based prognostic model, it is necessary to assess and compare several feature selection algorithms and classification algorithms as being a prognostic model. Getting inspired by recent studies [144], we have analyzed six different filter-based feature selection algorithms and seven different widely explored classification algorithms. Nevertheless, we did not follow the exact study in [144] by selecting similar classifiers and feature selection algorithms. Due to the limited sample size in our study, we have chosen those classifiers and feature selection algorithms, which are fitted with our limited datasets. These classifiers tune with the parameters' as defined in [139], where 179 different classifiers' families were evaluated on 121 datasets corresponding different domains. Similarly, to the previous study in [139], we tuned those parameters in our training datasets only using cross-validation of 100 repetitions, thus by assessing each classification algorithm in an unbiased manner. Their implementation in Python is publicly available. The recent studies have proven that radiomics derived from multiparametric MRI can provide a better prediction of tumor response in colorectal cancer [150-151]. In [151], it is confirmed that the radiomics derived from T2-w MRI modality can produce the best prediction of the colorectal tumor response than other MRI modalities. In our study, we have derived radiomics features from T2-w MRI.

Furthermore, our analysis is conducted on three different predictive models considering three different cases. In case 1, where the predictive model base on only handcrafted radiomics features was analyzed. In this case, we found that ANN classifiers produced the highest prognostic performance with the majority of feature selection algorithms. Similarly, Fisher based feature selection algorithm was observed with the best prognostic performance with all classifiers. Furthermore, our study demonstrates that the top-ranked five selected features (viz., *Volume* (shape-based feature), *Energy* (Haralick's Second order GLCM), *Entropy* (Haralick's Second order GLCM), *Long Run Emphasis* (*LRE*) (Second-order GLRLM), *Size Zone Non-Uniformity Normalized* (*SZNN*) (GLSZM), *Cluster shade* (Haralick's Second order GLCM)) provides the best prognostic performance.

Regarding case 2, where deep radiomic features were used as biomarkers as being the predictive model. We analyzed that the classifier, LSVM, has the best predictive performance with all feature selection algorithms. Accordingly, Relief or Fisher based feature selection algorithms arose as a good predictive model. In both cases, using multi-factor ANOVA, we found that the top-ranked features as 5 for the majority of feature selection algorithms have achieved the best predictive performance than all other ranks (i.e., 10, 15, 20 and 25). Considering this, we concatenated the 5 top-ranked features from case 1 case 2, as we discussed in case 3. In case 3, we found that the integration of these both Radiomics features to increase the performance of the majority of predictive models. Moreover, the comparable performance was given by LSVM and QSVM with all feature selection methods and Fisher and Relief based feature selection algorithms gave a comparable prognostic performance with respect to all classifiers. The LSVM as a classifier and the Relief as a feature selection algorithm gave the best prognostic performance. Our analysis found the best prognostic model in ease case, and we compare them in term of ROC curves as shown in figure 7.4 where the best performance is given by LSVM in combination with Relief using the combine radiomics features from both case 1 and case 2. Whereas the best predictive models of case 1 (i.e., ANN + Fisher) and case 2 (i.e., LSVM + Relief) have approximately similar ROC curves. Taking into account the results obtained in the above three different cases, our study may be a reference to the use of different radiomics based biomarkers to evaluate different prognostic models in different cancer diagnosis applications.

CHAPTER 8

CONCLUSION

The primary goal of this study was to design a system envisioned to automatically segment colorectal tumor with reasonable accuracy and predict tumor response in colorectal cancer evaluation in 3D MRI. This study was based on two fold objectives. First was related to segmentation of colorectal cancer in 3D MRI, and second, was to characterize the colorectal tumor into two groups; complete responders (CR) and non-responders (NR) to therapy in colorectal cancer. These two studies were carried out in parallel, in finding solutions for under the set objectives. Accordingly, the general conclusion is given for each study, as follows:

8.1 General Conclusion

Study 1: The segmentation of the tumor is the first and crucial step in the characterization of the tumor, which is generally segmented manually. The manual segmentation of the colorectal tumor is time consuming, laborious and it requires high expertise. Consider this problem; this study analyzed different deep learning-based algorithms (*viz.*, 3D FCNNs, 3D Unet, DenseVoxNet) as baseline methods to automatically segment tumor with reasonable accuracy. Based on pros and cons of those deep learning-based baseline methods, in this research work, 3D MSDenseNet, a novel 3D fully convolutional network architecture is presented for the accurate colorectal tumor segmentation in T2-weighted MRI volumes. Our proposed network provides dense interconnectivity among the horizontal layers (depth) and vertical layers (scaled). In this way, finer (i.e., high-resolution features) and coarser (low-resolution features) are coupled in a two-dimensional array of horizontal and vertical layers, and thus, features of all resolutions are produced from the first layer on and maintained throughout the network. However, in other networks (i.e., traditional CNN, 3D U-net, or DenseVoxNet) coarse level features are generated with increasing network depth.

The experiment results show that the multi-scale scheme in our proposed method has achieved the best performance overall. In addition, we incorporated a 3D level set algorithm with each method, as a post-processor to refine the segmented prediction. It is also shown that the incorporation of the 3D level set increases the performances of deep learning based approaches. As a future direction, the proposed method will further testify on other medical volumetric segmentation tasks.

Study 2: The accurate diagnosis and staging of colorectal cancer at first basis is the supreme in the oncology where medical experts have to decide the treatment plan that a patient should go for either therapy or surgical operation. Regarding the above motivational statement, the clinical appropriateness of our study lies in analyzing and classifying colorectal cancer non-invasively using radiomics and offering a user-friendly tool to the clinician. For such a purpose, in this study, we analyzed three different types of radiomics features, case 1: handcrafted based radiomics, case 2: deep features based radiomics and case 3: their combination. This study used seven classifiers (viz., LSVM, QSVM, LD, LR, NB, KNN, ANN) and six feature selection algorithms (viz., Fisher, Chi-square, T-test, MrMr, MIM, and Relief) to design and analyze different predictive models using different radiomic features. We have compared the performances of predictive models in terms of mean AUC (i.e., AUC; mean ± std).

Considering case 1, where traditional 109 handcrafted features were used, it was found that 5 features (viz., *Volume* (shape-based feature), *Energy* (Haralick's Second order GLCM), *Entropy* (Haralick's Second order GLCM), *Long Run Emphasis* (*LRE*) (Second-order GLRLM), *Size Zone Non-Uniformity Normalized* (*SZNN*) (GLSZM), *Cluster shade* (Haralick's Second order GLCM)) among 109 features provide the best prognostic performance in the predicting of the colorectal tumor response to therapy in the pre-treatment 3D MRI. These five best features were selected by the feature selection algorithm, so-called Fisher. Our analysis shows that Fisher presented the best prognostic performance in combination with all classifiers, that was (AUC; 0.8 ± 0.01). Using selected top-ranked five features, ANN classifier provided the best prognostic performance in combination with all feature selection algorithms, that was (AUC; 0.79 ± 0.016). In case 1, the best prognostic model was obtained from the combination of ANN classifier and Fisher feature selection algorithms.

Considering case 2 where 4096 deep radiomic features were extracted from our data by applying a pre-trained CNN_s model via transfer learning as explained in Chapter 7, section 7.1.2. In the case 2, we found that top five ranked features which are selected by Relief based feature selection algorithm produced the best prognostic performance in combination with all classifiers, that was (AUC; 0.82 ± 0.04). Similarly, using the top-ranked selected five deep radiomics features, LSVM classifier presented the best prognostic performance in combination with all feature selection, that was (AUC; 0.8 ± 0.042). In case 2, the combination of the LSVM and Relief based feature selection methods presented the best predictive performance in our study.

Considering case 3 where top-ranked five selected features from the above each case (i.e., case 1 and case 2) are merged to obtain the prognostic model. In case 3, we have found that the top-ranked five selected features by Relief showed the best prognostic performance (AUC; 0.87 ± 0.013) in combination with all classifiers. Similarly, LSVM classifier has the best performance (AUC; 0.84 ± 0.025) in combination with all feature selection algorithms. In case 3, the best prognostic performance was produced by the combination of Relief based feature selection and LSVM classifier algorithms. In the case 3, the top-ranked five best-handcrafted radiomics features selected by Relief, are: *Skewness* (First-order statistic feature), *Sphericity* (shape-based feature), *Energy* (Haralick's Second order GLCM), *Entropy* (Haralick's Second order GLCM), and *Cluster shade* (Haralick's Second order GLCM).

Our results in comparison of different classifiers and feature selection algorithms revealed that the combination of these both types of radiomics based features could give a better prognostic model for the colorectal cancer evaluation in 3D MRI. It was also observed that the performance of only handcrafted and only deep features have no significant difference.

8.2 Future Research Work

Below are some research directions for the possible improvement in the work carried out in this Ph.D. thesis:

1. Although, the above both studies are linked these were carried out in parallel, independently, due to unavailability of the data on time. In our first study related to segmentation of colorectal tumor in 3D MRI, we have achieved promising results. In the future, it would be interesting to analyze our second study by replacing the manual segmentation by automatic segmentation obtained by the proposed method.

2. In the second study, case 2, the deep radiomics features were extracted via transfer learning where pre-trained CNN_S model was trained on natural images. In the future, it would be interesting to train the CNN_S model on MR images dataset and may be utilized accordingly in this study. For such a purpose, an abundant number of MRI datasets are required. Nonetheless, nowadays, there is a considerable advancement in the medical imaging data, and these data are publicly available at the online database. Furthermore, there are many data augmentation techniques (i.e., rotation, flipping, mirroring, etc.) can be incorporated to enlarge the training samples.

3. In the second study, case 2 where the cropped tumor was resized using bi-cubic interpolation to required input size of pre-trained CNN_S model. The required input size was 224 × 224 while the cropped tumors' sizes were in the range of average width and height, ~ 45 to 33 approximately, pixels with respect to the actual resolution of the MRI scan. In this way, we were required to super-resolved by a factor of ×5. We believe that employing a better super-resolution algorithm instead of traditional bi-cubic interpolation could produce a better prognostic performance.

BIBLIOGRAPHY

[1]. Ashiya, "Notes on the Structure and Functions of Large Intestine of Human Body," http://www.preservearticles.com/201105216897/notes-onthe-structure-and-functions-of-large-intestine-of-human-body.html, 2013.

[2]. M. H. Soomro, G. Giunta, A. Laghi, D. Caruso, M. Ciolina, C. De Marchis, S. Conforto, M. Schmid, "Haralick's Texture Analysis applied to colorectal T2-weighted MRI: A preliminary study of significance for cancer evolution," In Proc. of *13th IASTED* (*BioMed 2017*), pp. 16-19, 2017.

[3]. R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2017," *CA: A Cancer Journal for Clinicians*, vol. 67, pp. 7–30, 2017.

[4]. R. G. Beets-Tan, D. M. Lambregts, M. Maas, S. Bipat, B. Barbaro, F. Caseiro-Alves, L. Curvo-Semedo, H. M. Fenlon, M. J. Gollub, S. Gourtsoyianni, S. Halligan, C. Hoeffel, S.H. Kim, A. Laghi, A. Maier, S. R. Rafaelsen, J. Stoker, S. A. Taylor, M. R. Torkzad, L. Blomqvist, "Magnetic resonance imaging for the clinical management of rectal cancer patients: recommendations from the 2012 European Society of Gastrointestinal and Abdominal Radiology (ESGAR) consensus meeting," *Eur Radiol*, vol. 23, pp. 2522–2531, 2013.

[5]. A. Laghi, M. Ferri, Catalano C, Baeli I, Iannaccone R, Iafrate F, Ziparo V, R. Passariello, "Local staging of rectal cancer with MRI using a phased array body coil," *Abdom Imaging*, vol. 27(4), pp. 425–431, 2002.

[6]. MS Group, "Diagnostic accuracy of preoperative magnetic resonance imaging in predicting curative resection of rectal cancer: prospective observational study," *BMJ* , vol. 333, pag. 779, 2006.

[7]. J. J. van den Broek, F. S. van der Wolf, M. J. Lahaye, L. A. Heijnen, C. Meischl, M. A. Heitbrink, W. H. Schreurs, "Accuracy of MRI in restaging locally advanced rectal cancer after preoperative chemoradiation," *Dis Colon Rectum*, vol. 60, pp. 274–283, 2017.

[8]. M. Maas, D. M. Lambregts, M. J. Lahaye, G. L. Beets, W. Backes, R. F. Vliegen, M. Osinga-de Jong, J. E. Wildberger, R. G. Beets-Tan, "T-staging of rectal cancer: accuracy of 3.0 Tesla MRI compared with 1.5 Tesla," *Abdom Imaging*, vol. 37, pp. 475–481, 2012.

[9]. U. I. Attenberger, L. R. Pilz, J. N. Morelli, D. Hausmann, F. Doyon, R. Hofheinz, P. Kienle, S. Post, H. J. Michaely, S. O. Schoenberg, D. J. Dinter, "Multi-parametric MRI of rectal cancer—do quantitative functional MR measurements correlate with radiologic and pathologic tumor stages?," *Eur J Radiol,* vol. 83, pp. 1036–1043, 2014.

[10]. M. Hotker, L. Tarlinton, Y. Mazaheri, K. M. Woo, M. Gonen, L. B. Saltz, K. A. Goodman, J. Garcia-Aguilar, M. J. Gollub, "Multiparametric MRI in the assessment of response of rectal cancer to neoadjuvant chemoradiotherapy: a comparison of morphological, volumetric and functional MRI parameters," *Eur Radiol*, vol. 26, pp. 4303–4312, 2016.

[11]. L. Curvo-Semedo, D. M. Lambregts, M. Maas, G. L. Beets, F. Caseiro-Alves, R. G. Beets-Tan, "Diffusion-weighted MRI in rectal cancer: apparent diffusion coefficient as a potential noninvasive marker of tumor aggressiveness," *J Magn Reson Imaging JMRI,* vol. 35, pp. 1365–1371, 2012.

[12]. S. H. Jung, S. H. Heo, J. W. Kim, Y. Y. Jeong, S. S. Shin, M. G. Soung, H. R. Kim, H. K. Kang, "Predicting response to neoadjuvant chemoradiation therapy in locally

advanced rectal cancer: diffusion-weighted 3 Tesla MR imaging," *J Magn Reson Imaging JMRI,* vol. 35, pp. 110–116, 2012.

[13]. P. Q. Cai, Y. P. Wu, X. An, X. Qiu, L. H. Kong, G. C. Liu, C. M. Xie, Z. Z. Pan, P. H. Wu, P. R. Ding, "Simple measurements on diffusion weighted MR imaging for assessment of complete response to neoadjuvant chemoradiotherapy in locally advanced rectal cancer," *Eur Radiol,* vol. 24, pp. 2962–2970, 2014.

[14]. M. Intven, O. Reerink, M. E. Philippens, "Dynamic contrast enhanced MR imaging for rectal cancer response assessment after neo-adjuvant chemoradiation," *J Magn Reson Imaging JMRI,* vol 41, pp. 1646–1653, 2015.

[15]. M. J. Gollub, D. H. Gultekin, O. Akin, R. K. Do, J. L. 3rd Fuqua, M. Gonen, D. Kuk, M. Weiser, L. Saltz, D. Schrag, K. Goodman, P. Paty, J. Guillem, G. M. Nash, L. Temple, J. Shia, L. H. Schwartz, "Dynamic contrast enhanced-MRI for the detection of pathological complete response to neoadjuvant chemotherapy for locally advanced rectal cancer," *Eur Radiol,* vol. 22, pp. 821–831, 2012.

[16]. P. W. Hamilton, P. H. Bartels, D. Thompson, N. H. Anderson, R. Montironi, J. M. Sloan , "Automated location of dysplastic fields in colorectal histology using image texture analysis," *J Pathol,* vol. 182, pp. 68–75, 1997.

[17]. C. N. De Cecco, B. Ganeshan, M. Ciolina, M. Rengo, F. G. Meinel, D. Musio, F. De Felice, N. Raffetto, V. Tombolini, A. Laghim, "Texture analysis as imaging biomarker of tumoral response to neoadjuvant chemoradiotherapy in rectal cancer patients studied with 3-T magnetic resonance," *Invest Radiol* , vol. 50, pp. 239–245, 2015.

[18]. C. N. De Cecco, M. Ciolina, D. Caruso, M. Rengo, B. Ganeshan, F. G. Meinel, D. Musio, F. De Felice, V. Tombolini, A. Laghi, "Performance of diffusion-weighted imaging, perfusion imaging, and texture analysis in predicting tumoral response to neoadjuvant chemoradiotherapy in rectal cancer patients studied with 3T MR: initial experience," *Abdom Radiol (NY),* vol. 41, pp. 1728–1735, 2016.

[19]. G. Castellano, L. Bonilha, L. M. Li, F. Cendes, "Texture analysis of medical images," *Clin Radiol* , vol. 59, pag. 106, 2004.

[20]. Ahmed, P. Gibbs, M. Pickles, L. Turnbull, "Texture analysis in assessment and prediction of chemotherapy response in breast cancer," *J Magn Reson Imaging JMRI,* vol. 38, pp. 89–101, 2013.

[21]. R. M. Haralick, K. Shanmugam, I. Dinstein, "Textural features for image classification," *In: IEEE Xplore Document,* 2007.

[22]. S. Gourtsoyianni, G. Doumou, D. Prezzi, B. Taylor, J. J. Stirling, N. J. Taylor, M. Siddique, G. J. R. Cook, R. Glynne-Jones, V. Goh, "Primary rectal cancer: repeatability of global and local-regional MR imaging texture features," *Radiology,* 2017.

[23]. O. Jalil, A. Afaq, B. Ganeshan, U. B. Patel, D. Boone, R. Endozo, A. Groves, B. Sizer, T. Arulampalam, "Magnetic resonance based texture parameters as potential imaging biomarkers for predicting long-term survival in locally advanced rectal cancer treated by chemoradiotherapy," *Colorectal Dis* , vol. 19, pp. 349–362, 2017.

[24]. G. Nketiah, M. Elschot, E. Kim, J. R. Teruel, T. W. Scheenen, T. F. Bathen, K. M. Selnaes, "T2-weighted MRI-derived textural features reflect prostate cancer aggressiveness: preliminary results," *Eur Radiol,* vol. 27, pp. 3050–3059, 2017.

[25]. Oikonomou, F. Khalvati, et al., “Radiomics Analysis at PET/CT Contributes to Prognosis of Recurrence and Survival in Lung Cancer Treated with Stereotactic Body Radiotherapy,” *Scientific Reports*, vol. 8, 2018.

[26]. H. J. Aerts, E.R. Velazquez, et al., “Decoding Tumor Phenotype by Noninvasive Imaging Using a Quantitative Radiomics Approach,” *Nature Communications*, vol. 5, 2014.

[27]. Y. Zhang, A. Oikonomou, et al., “Radiomics-based Prognosis Analysis for Non-small Cell Lung Cancer,” *Scientific Reports*, vol. 7, 2017.

[28]. J. Griethuysen, A. Fedorov, et al. “Computational Radiomics System to Decode the Radiographic Phenotype,” *Cancer Research*, vol. 77, no. 21, pp. 104-107, 2017.

[29]. P. Lambin, E. Rios-velazquez, et al., “Radiomics: Extracting more Information from Medical Images using Advanced Feature Analysis,” *European Journal of Cancer*, vol. 48, no. 4, pp. 441-446, 2012.

[30]. V. Kumar, Y. Gu, et al., “Radiomics: The Process and the Challenges,” *Magnetic Resonance Imaging*, vol. 30, no. 9, pp. 1234-1248, 2012.

[31]. R. Thawani, M. McLane, et al., “Radiomics and Radiogenomics in Lung Cancer: A Review for the Clinician ,” *Lung cancer*, vol. 115, pp. 34-41, 2017.

[32]. J. Tian, D. Dong, et al., “Radiomics in Medical Imaging-Detection, Extraction and Segmentation,” *Artificial Intelligence in Decision Support Systems for Diagnosis in Medical Imaging*, vol. 140, pp. 267-333, 2018.

[33]. R. Gillies, P. Kinahan, et al., “Radiomics: Images Are More than Pictures, They Are Data,” *Radiology*, vol. 278, no. 2, pp. 563-577, 2016.

[34]. L. Oakden-Rayner, G. Carneiro, et al., “ Precision Radiology: Predicting Longevity Using Feature Engineering and Deep Learning Methods in a Radiomics Framework,” *Scientific Reports*, vol. 7, no. 1, 2017.

[35]. R. Paul, S. Hawkins, et al., “Deep Feature Transfer Learning in Combination with Traditional Features Predicts Survival Among Patients with Lung Adenocarcinoma,” *Tomography: a journal for imaging research*, vol. 2, no. 4, pp. 388-395, 2016.

[36]. L. Fu, J. Ma, et al., “Automatic Detection of Lung Nodules: False Positive Reduction Using Convolutional Neural Networks and Handcrafted Features,” In Proc. *SPIE*, vol. 10134, 2017.

[37]. S. Liu, H. Zhengr, et al., “Prostate Cancer Diagnosis Using Deep Learning with 3D Multiparametric MRI,” In Proc. *SPIE*, vol. 10134, 2017.

[38]. J. Lao, Y. Chen, et al., “A Deep Learning-Based Radiomics Model for Prediction of Survival in Glioblastoma Multiforme,” *Scientific Reports*, vol. 7, no. 1, 2017.

[39]. M.A. Gambacorta, C. Valentini, N. Dinapoli, L. Boldrini, N. Caria, M.C. Barba, G.C. Mattiucci, D. Pasini, B. Minsky, V. Valentini, “Clinical validation of atlas-based auto-segmentation of pelvic volumes and normal tissue in rectal tumors using auto-segmentation computed system,” *Actaoncologica*, vol. 52, pp. 1676-1681, 2013.

[40]. B. Irving, A. Cifor, B. W Papież, J. Franklin, E. M. Anderson, M. Brady, and J. A. Schnabel, "Automated colorectal tumor segmentation in dce-mri using supervoxel neighbourhood contrast characteristics," In *MICCAI*, pp. 609–616, Springer, 2014.

[41]. S. Trebeschi, J. J. M. van Griethuysen, D. M. J. Lambregts, M. J. Lahaye, C. Parmer, F. C. H. Bakers, N. Peters, R. G. H. Beets-Tan, H. Aerts, "Deep Learning for Fully-Automated Localization and Segmentation of Rectal Cancer on Multiparametric MR," *Scientific reports*, vol. 7, pag. 5301,2017.

[42]. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, "Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network," in *Medical Image Computing and Comput.-Assisted Intervention-MICCAI*, pp. 246–253, Springer, 2013.

[43]. M. Havaei, A. Davy, D. Warde-Farley, A . Biard, A . Courville, Y. Bengio, C. Pal, P. M. Jodoin, H. Larochelle, "Brain tumor segmentation with deep neural networks," *Med. Image Anal.*, vol. 35, pp. 18–31, 2017.

[44]. H.R. Roth, L. Lu, A. Farag, A. Sohn, R.M. Summers, "Spatial Aggregation of Holistically-Nested Networks for Automated Pancreas Segmentation," *Springer International Publishing Cham*, pp. 451–459, 2016.

[45]. Ö. Çiçek, A. Abdulkadir, S.S. Lienkamp, T. Brox, O. Ronneberger, "3D U-net: learning dense volumetric segmentation from sparse annotation," In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 424–432, Springer, 2016.

[46]. H. Chen, Q. Dou, L. Yu, J. Qin, P. A. Heng, "Voxresnet: Deep voxelwise residual networks for brain segmentation from 3d mr images," *NeuroImage*, 2017.

[47]. Q. Dou, L. Yu, H. Chen, Y. Jin, X. Yang, J. Qin, P.A. Heng, "3D deeply supervised network for automated segmentation of volumetric medical images," *Medical image analysis*, vol. 41, pp. 40-54, 2017.

[48]. M. H. Soomro, G. De Cola, S. Conforto, M. Schmid, G. Giunta, E. Guidi, E. Neri, D. Caruso, M. Ciolina, A. Laghi, "Automatic segmentation of colorectal cancer in 3D MRI by combining deep learning and 3D level-set algorithm-a preliminary study," In Proc. of *IEEE 4th Middle East Conference on Biomedical Engineering (MECBME)*, Tunis, pp. 198-203, 2018.

[49]. L. Yu, J. Z. Cheng, Q. Dou, X. Yang, H. Chen, J. Qin, P. A. Heng, "Automatic 3D cardiovascular MR segmentation with densely-connected volumetric convnets," In *MICCAI*, pp. 287-295, 2017.

[50]. B. H. Menze et al., "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)," *IEEE Transactions on Medical Imaging*, vol. 34, pp. 1993–2024, 2015.

[51]. Van de Velde CJ, Boelens PG, Borras JM et al. EURECCA colorectal: multidisciplinary management: European consensus conference colon & rectum. European Journal of Cancer, January 2014.

[52]. Shandra Bipat, MSc Afina S. Glas, MD, PhD Frederik J. M. Slors, MD, PhD Aeilko H. Zwinderman, PhD Patrick M. M. Bossuyt, PhD Jaap Stoker, MD, PhD Rectal Cancer: Local Staging and Assessment of Lymph Node Involvement with Endoluminal US, CT, and MR Imaging—A Meta-Analysis Radiology 2004.

[53]. Bingham SA, Day NE, Luben R, Ferrari P, Slimani N, Norat T, Clavel-Chapelon F et al. Dietary fibre in food and protection against colorectal cancer in the European Prospective Investigation into Cancer and Nutrition (EPIC): an observational study. The Lancet 2003 Vol 361: 1496-1501.

[54]. Norat T, Bingham S, Ferrari P, Slimani N, Jenab M, Mazuir M, et al. Meat, fish, and colorectal cancer risk: the European Prospective Investigation into cancer and nutrition. J Natl Cancer Inst. 2005; 97(12): 906-16.

[55]. Ferrari P, Jenab M, Norat T, Moskal A, Slimani N, Olsen A, et al. Lifetime and baseline alcohol intake and risk of colon and rectal cancers in the European prospective investigation into cancer and nutrition (EPIC). Int J Cancer. 2007; 121(9): 2065-72.

[56]. Giovannucci E. An updated review of the epidemiological evidence that cigarette smoking increases risk of colorectal cancer. Cancer Epidemiol Biomarkers Prev. 2001; 10(7): 725-31.

[57]. Shimizu H, Mack TM, Ross RK, Henderson BE. Cancer of the gastrointestinal tract among Japanese and white immigrants in Los Angeles County. J Natl Cancer Inst. 1987; 78(2): 223-8.

[58]. Baxter NN, Tepper JE, Durham SB, Rothenberger DA, Virnig BA. Increased risk of rectal cancer after prostate radiation: a population-based study. Gastroenterology. 2005; 128(4): 819-24.

[59]. Bernstein CN, Blanchard JF, Kliewer E, Wajda A. Cancer risk in patients with inflammatory bowel disease: a population-based study. Cancer. 2001; 91(4): 854-62.

[60]. Zampino MG, Labianca R, Beretta GD, Magni E, Gatta G, Leonardi MC, Chiappa A, Biffi R, de Braud F, Wils J. Rectal cancer. Critical Reviews in Oncology Hematology. 2009; 70(2): 160-82.

[61]. Fernandes G, Leme C, Ruiz-Cintra M, Pavarino E, Netinho J, Goloni-Bertoll E. Clinical and epidemiological evaluation of patients with sporadic colorectal cancer. Journal of Coloproctology (Rio J.) vol.34 no.4 Rio de Janeiro Oct./Dec. 2014.

[62]. Weitz J, Kienle P, Magener A, Koch M, Schrodel A et al. Detection of Disseminated Colorectal Cancer Cells in Lymph Nodes, Blood and Bone Marrow. American Association for Cancer Research, Vol. 5, 1830–1836, July 1999 Clinical Cancer Research.

[63]. Nougaret S, Reinhold C, Mikhael HW, et al. The Use of MR Imaging in Treatment Planning for Patients with Rectal Carcinoma: Have You Checked the "DISTANCE"? Radiology: Volume 268: Number 2—August 2013.

[64]. Bipat S, Glas AS, Slors FJ, Zwinderman AH, Bossuyt PM, Stoker J. Rectal cancer: local staging and assessment of lymph node involvement with endoluminal US, CT, and MR imaging—a meta-analysis. Radiology 2004; 232:773–783.

[65]. Engelen SM, Beets GL, Beets-Tan RG. Role of preoperative local and distant staging in rectal cancer. Onkologie 2007;30(3):141–5.

[66]. Merkel S, Mansmann U, Papadopoulos T, Wittekind C, Hohenberger W, Hermanek P. The prognostic inhomogeneity of colorectal carcinomas stage III: a proposal for subdivision of stage III. Cancer 2001;92(11):2754–2759.

[67]. Weitz J, Kienle P, Magener A, Koch M, Schrodel A et al. Detection of Disseminated Colorectal Cancer Cells in Lymph Nodes, Blood and Bone Marrow. American Association for Cancer Research, Vol. 5, 1830–1836, July 1999 Clinical Cancer Research.

[68]. MERCURY Study Group. Extramural depth of tumor invasion at thin-section MR in patients with rectal cancer: results of the MERCURY study. Radiology 2007;243(1): 132–139.

[69]. Wiley W. Souba, Mitchell P, M.D. Fink, Gregory J et al. American College of Surgeons. Surgery principles & practice. United states of America2005.

[70]. Linee guida Tumori del Colon-Retto, Edizione 2016. Associazione Italiana di Oncologia Medica (AIOM).

[71]. Heald RJ, Husband EM, Ryall RD. The mesorectum in rectal cancer surgery—the clue to pelvic recurrence? Br J Surg 1982; 69: 613–16.

[72]. Kapiteijn E, Marijnen CA, Nagtegaal ID, et al. Preoperative radiotherapy combined with total mesorectal excision for resectable rectal cancer. N Engl J Med 2001;345:638–646.

[73]. Sauer R, Becker H, Hohenberger W, et al. Preoperative versus postoperative chemoradiotherapy for rectal cancer. N Engl J Med 2004; 351:1731–1740.

[74]. Habr-Gama A, Perez RO, Nadalin W et al. Operative vs nonoperative treatment for stage 0 distal rectal cancer following chemoradiation therapy: long term results. Ann Surg 2004; 240:711-717.

[75]. Maas M, Beets-Tan RG, Lambregts DM et al. Wait and see policy for clinical complete responders after chemoradiation for locally advanced rectal cancer. Journal of Clinical oncology 2011;35:4633-40.

[76]. Dworak O, Keilholz L, Hoffmann A. Pathological features of rectal cancer after preoperative radiochemotherapy. Int J Colorectal Dis. 1997;12:19–23.

[77]. Svenjathies D. Tumor regression grading for gastrointestinal carcinomas after neoadjuvant tratment. Frontiers in Oncology 2013; 262:1-7.

[78]. W. S. McCulloch , W. Pitts , A logical calculus of the ideas immanent in nervous activity*,* in *"The bulletin of mathematical biophysics"*, volume 5, pages 115–133, 1943.

[79]. A. Krenker, J. Bester , A. Kos , Introduction to the artificial neural networks. Artificial neural networks: methodological advances and biomedical applications, in "*InTech",* pages 978–953, 2011.

[80]. J. Friedman, T. Hastie, R.Tibshirani, The elements of statistical learning, in "*Springer*", volume 1, Berlin, 2001.

[81]. V. Nair , G. E. Hinton, Rectified Linear Units Improve Restricted Boltz-mann Machines, in "*Proceedings of the 27th International Conference on International Conference on Machine Learning"*, pages 807–814, USA, 2010.

[82]. C. M. Bishop, Pattern recognition and machine learning, in *"Springer"*, volume 1, New York, 2006.

[83]. I. Goodfellow, Y. Bengio, A. Courville, *"Deep learning"*, USA, 2016.

[84]. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in "*Advances in neural information processing systems"*, pages 1097–1105, 2012.

[85]. K. Simonyan, A. Zisserman, Very deep convolutional networks for largescale image recognition, in "*Proceedings of International Conference on Learning Representations (ICLR)"*, San Diego, USA, 2015.

[86]. J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in "*Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)"*, Boston, USA, 2015.

[87]. L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected, in "*IEEE Transactions on Pattern Analysis and Machine Intelligence",* volume 40, 2016.

[88]. H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in "*Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)"*, Santiago, Chile, 2015.

[89]. J. Zbontar , Y. LeCun, Stereo matching by training a convolutional neural network to compare image patches, in *"Journal of Machine Learning Research 17"*, pages 1–32, 2016.

[90]. R. Girshick, Fast r-cnn, in *"Proceedings of The IEEE International Conference on Computer Vision (ICCV)"*, pages 1140-1148, 2015.

[91]. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real time object detection with region proposal networks, in *"Advances in Neural Information Processing Systems (NIPS)"*, 2015.

[92]. F. Lee, Convolutional Neural Networks for Visual Recognition, 2017.

[93]. S. Io, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in *"Proceedings of 32nd International Conference on Machine Learning (ICML)"*, Lille, France, 2015.

[94]. S.P. Víctor, I. Segura-Bedmar, Evaluation of Pooling Operations in Convolutional Architectures for Drug-Drug Interaction Extraction, in "*BMC Bioinformatics 19*", 2018.

[95]. S. Han, J. Pool, S. Narang, H. Mao, E. Gong, S. Tang, E. Elsen, P. Vajda, M. Paluri, J. Tran, B. Catanzaro,W. J. Dally, Dsd: Dense-sparse-dense training for deep neural networks, in *"ICLR"*, arXiv:1607.04381, 2017.

[96]. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, in *"Journal of Machine Learning Research 15"*, 2014.

[97]. D. Rumelhart, G. Hinton, R. Williams, Learning representations by backpropagating errors, in *"Nature"*, 1986.

[98]. M. H. Soomro, G. Giunta, A. Laghi, D. Caruso, M. Ciolina, C. De Marchis, S. Conforto, M. Schmid, Segmenting MR Images by Level-Set Algorithms for Perspective Colorectal Cancer Diagnosis, in "Lecture Notes in Computational Vision and Biomechanics", Springer, volume 27, 2018.

[99]. Y. Chen, A novel approach to segmentation and measurement of medical image using level set methods, in "Magnetic Resonance Imaging", volume 39, pages 175 – 193, 2017.

[100]. D. Cremers, M. Rousson, and R. Deriche, A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape, in "International Journal of Computer Vision", volume 72, April 2007.

[101]. K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, B. Glocker, D. Rueckert, Efficient multi-scale 3D cnn with fully connected crf for accurate brain lesion segmentation, in "Medical Image Analysis", volume 36, pag. 6178, 2017.

[102]. J. Dolz, C. Desrosiers, I. B. Ayed, 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study, in "NeuroImage", volume 170, pages 456-470, 2017.

[103]. M. Tang, S. Valipour, Z. V. Zhang, D. Cobzas, M. Jagersand, A deep level set method for image segmentation, in "Lecture Notes in Computer Science" , Springer, 2017.

[104]. O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in "Proceedings of the 2015 International Conference on Medical Image Computing and Computer-Assisted Intervention" , Springer, pages 234–241, 2015.

[105]. G. Huang., Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition" , 2017.

[106]. T. D. Bui, J. Shin, T. Moon, 3D densely convolution networks for volumetric segmentation, arXiv:1709.03199, 2017.

[107]. G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, K. Weinberger, Multi-scale dense networks for resource efficient image classification, in "International Conference on Learning Representations", 2018.

[108]. V. Caselles , R. Kimmel, G. Sapiro, Geodesic active contours, in "International Journal of Computer Vision", volume 22, pages 61–79, 1997.

[109]. T. S. Yoo, M. J. Ackerman, W. E. Lorensen, W. Schroeder , V. Chalana, S. Aylward, D.Metaxas, R. Whitaker, "Engineering and algorithm design for an image processing Api: a technical report on ITK– the insight toolkit," In Proc. *of Medicine Meets Virtual Reality 02/10*, vol. 85, pp. 586–592, 2002.

[110]. P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability," *Neuroimage*, vol. 31, pp. 1116–1128, 2006.

[111]. Y. Jia, E. Shelhamer, J. Donahue et al., "Caffe: Convolutional architecture for fast feature embedding," arXiv:1408.5093, 2014.

[112]. W. Liu, A. Rabinovich, A. C. Berg, "Parsenet: Looking wider to see better," arXiv:1506.04579v2, 2015.

[113]. P. Kontschieder, S.R. Bulo, H. Bischof, M. Pelillo, "Structured class-labels in random forests for semantic image labelling," *2011 International Conference on Computer Vision*, pp. 2190–2197, 2011.

[114]. L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no.3, pp. 297–302, 1945.

[115]. A. A. Taha, A. Hanbury, "Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, no. 1, pag. 29, 2015.

[116]. D.C. Ciresan, L.M. Gambardella, A. Giusti, J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," *Advances in Neural Information Processing Systems*, vol. 25, pp. 2852–2860, 2012.

[117]. A. Kronman, L. Joskowicz, "A geometric method for the detection and cor rection of segmentation leaks of anatomical structures in volumetric medical images," *International Journal of Computer Assisted Radiology and Surgery*, pp. 369–380, 2016.

[118]. Y. T. Chen, "A novel approach to segmentation and measurement of medical image using level set methods," *Magnetic Resonance Imaging*, vol. 39, pp. 175 – 193, 2017.

[119]. Ö. Çokluk-Bökeoğlu, Ş. Büyüköztürk, "Discriminant function analysis: Concept and application," *Eğitim araştırmaları dergisi*, vol. 33, pp. 73-92, 2008.

[120]. *W. N. Venables, B. D. Ripley, "Modern Applied Statistics with S (4th ed.)," Springer Verlag, 2002.*

[121]. T. Hastie, R. Tibshirani, J. Friedman, "The element of statistical learning," section 4.3, pp. 106-119, 2008.

[122]. Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *In Neural Networks: Tricks of the Trade, Springer*, pp. 437–478, 2012.

[123]. K. P. Bennett, C. Campbell, "Support vector machines: hype or hallelujah?," *ACM SIGKDD Explorations Newsletter*, vol. 2, no.2, pp. 1–13, 2000.

[124]. C. M. Bishop, "Pattern recognition and machine learning," vol. 1, *Springer New York*, 2006.

[125]. B. E. Boser, I. M. Guyon, V. N. Vapnik, "A training algorithm foroptimal margin classifiers," In Proc. *of the fifth annual workshop on Computationallearning theory*, pp. 144–152, *ACM*, 1992.

[126]. C. J.C. Burges, "A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery," vol. 2, issue. 2, pp. 121–167, *Springer*, 1998.

[127]. O. Chapelle, B. Schölkopf, A. Zien, "*Semi-supervise learning*," vol. 2, *MIT press Cambridge*, 2006.

[128]. V. Cherkassky, Y. Ma, "Practical selection of svm parameters and noise estimation for svm regression," *Neural networks*, vol. 17, no. 1, pp. 113–126, 2004.

[129]. A. P. Cunningham, S. J. Delany, "k-nearest neighbour classifiers," *Mult Classif. Syst*, pp. 1–17, 2007.

[130]. P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol 5 no. 10, pp. 78–87, 2012.

[131]. R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," In Proc. *of the 24th international conference on Machine learning*, pp. 759–766, *ACM*, 2007.

[132]. K. Weinberger, J. Blitzer, L. Saul, "Distance metric learning for large margin nearest neighbor classification," *Advances in neural information processing systems*, vol. 18, pp. 1473–1480, 2006.

[133]. G. James, D. Witten, T. Hastie, R. Tibshirani, "An Introduction to Statistical Learning with Applications in R," *Springer Texts in Statistics*, 2013.

[134]. J. Rossiter, "Multimodal Intent Recognition for Natural Human-Robotic Interaction," University of Birmingham, *PhD Thesis*, 2011.

[135]. S. Srihari, "Artificial Neural Networks," *Center of Excellence for Document Analysis and Recognition*, The State University of New York. Available [Online]: http://www.cedar.buffalo.edu/~srihari/CSE555/Chap6.Part1.pdf

[136]. K. P. Murphy, "*Naive Bayes Classifiers*," Available [Online]:http://www.ic.unicamp.br/~rocha/teaching/2011s1/mc906/aulas/naive-bayes.pdf

[137]. *S. T. Roweis, L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," Science, vol. 290, issue 5500, pp. 2323–2326, 2000.*

[138]. M. Mohri, A. Rostamizadeh, and A. Talwalkar, "Foundations of machine learning," Ch. 1, 1–3, *MIT press*, 2012.

[139]. M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?," *J. Mach. Learn. Res.*, Vol. 15, pp.3133–3181, 2014.

[140]. J. West, D. Ventura, S. Warnick, "Spring Research Presentation: A Theoretical Foundation for Inductive Transfer," *Brigham Young University, College of Physical and Mathematical Sciences. Archived from the original on* 2007-08-01. Retrieved 2007-08-05.

[141]. K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, "Return of the devil in the details: delving deep into convolutional nets," *In: British Machine Vision Conference 2014. Nottingham, UK. British Machine Vision Association*; 2014.

[142]. Vedaldi A, Lenc K. MatConvNet: Convolutional neural networks for MATLAB. *In: 23rd ACM International Conference on Multimedia. Brisbane, Australia*. 2015.

[143]. S. J. Pan, and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, pp. 1345–1359, 2010.

[144]. Parmar C, Grossmann P, Bussink J, Lambin P, Aerts HJ. Machine learning methods for quantitative radiomic biomarkers. *Sci Rep* (2015) 5:13087.doi:10.1038/srep13087.

[145]. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: a large-scale hierarchical image database. In: *IEEE Conference on Computer Vision and Pattern*

Recognition. Miami, FL. Institute of Electrical and Electronics Engineers (IEEE); 2009.

[146]. Guyon, I. & Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res*. 3, 1157–1182 (2003).

[147]. A. Trakarnsanga, M. Gönen, J. Shia, G.M. Nash, L.K. Temple et. al., "Comparison of tumor regression grade systems for locally advanced rectal cancer after multimodality treatment," *J Natl Cancer Inst* ,106(10*)*, 2014.

[148]. K.I. Abdul-Jalil, K.M. Sheehan, J. Kehoe et. al., "The prognostic value of tumour regression grade following neoadjuvant chemoradiation therapy for rectal cancer. Colorectal disease," *the official journal of the Association of Coloproctology of Great Britain and Ireland*, pp. 16-25, vol. 16 (1), 2014.

[149]. F.M. Vecchio, V. Valentini et. al., "The relationship of pathologic tumor regression grade (TRG) and outcomes after preoperative therapy in rectal cancer," *Int J Radiat Oncol Biol Phys*, vol 62(3), pp. 752-760, 2005.

[150]. N. Horvat, H. Veeraraghavan, M. Khan et al., "MRI of rectal cancer: radiomics analysis to assess treatment response after neoadjuvant therapy," ***Radiology*** vol. 287(3), pp. 833–843, 2018.

[151]. Y. Cui, X. Yang, Z. Shi, Z. Yang , X. Du, Z. Zhao, X. Cheng, "Radiomics analysis of multiparametric MRI for prediction of pathological complete response to neoadjuvant chemoradiotherapy in locally advanced rectal cancer," *Eur Radiol*. 2018. https://doi.org/10.1007/s00330-018-5683-9.

APPENDIX A

HANDCRAFTED RADIOMICS FEATURES

Handcrafted radiomics features are extracted using python based pyradiomics code available at https://pyradiomics.readthedocs.io/en/latest/, [28]. Extraction of handcrafted Radiomics features can subdivide them into classes as follow:

- Shape-Based
- First order statistics features
- Gray Level Co-occurrence Matrix (GLCM)
- Gray Level Run Length Matrix (GRLM)
- Gray Level Size Zone Matrix (GLSZM)
- Neighboring Gray Tone Difference Matrix (NGTDM)

A.1. Shape-based

The Shape-based features present inside descriptors of the 3D size and the region of interest shape. These features characteristics do not depend on the intensity coming from the voxels and are derived from the approximated shape defined by the triangle mesh. To make this build are used vertices defined as points halfway on an edge between two voxels placed one outside the region the interest and the other inside them. Connecting 3 of them are obtain a triangular mesh. Every triangle made using this technique shares each side with a nearby triangle and vice-versa. The triangular mesh is generated using a marching cubes algorithm where a 2 × 2 cube is moved through the mask space. For every position are obtained only two different response, segmented (1) or not-segmented (0). A unique cube-index is obtained, evaluating the corner in binary code, to define how many triangles are present in a cube and everything is saved in a tab. The triangles are defined by the normal (from the cross product of vector describing 2 out of 3 edges), oriented always in the same direction and in Pyradiomics [1] the calculated normal pointing outward. All of them is essential to obtain the volume, used to calculate the mesh-volume [2].

Let's introduce some values:

- N_v it represents the number of voxels in the ROI.
- N_f it represents the number of triangles defines the mesh.
- V is the mesh volume in mm^3
- A is the mesh area in mm^2

A.1.1 Mesh volume

$$V_i = \frac{O_{a_i} \cdot (O_{b_i} \times O_{c_i})}{6} \quad \text{(A.1.1)}$$

Where $V = \sum_{i=1}^{N_f} V_i$ and V is the volume that is obtained by the previous calculation of each triangle mesh volume of the ROI and after there is the sum. Every face is defined by I and the *i-th* points a_i, b_i *and* c_i. The normal identifies the sign of the volume, it must be consistently defined as either facing the outward-ROI and the inward-ROI. After that, there is the sum of all V_i and is obtained all the volume of the ROI [2].

A.1.2 Voxel volume

$$V_{voxel} = \sum_{k=1}^{N_v} V_k \quad \text{(A.1.2)}$$

V_{oxel} represents the volume of the ROI and is obtained by multiplying the number of voxels in the ROI by the volume of a single voxel V_k. This feature is an approximation and doesn't use the mesh, for that isn't used for the calculation of other shape features.

A.1.3 Surface Area

$$A_i = \frac{1}{2}\left|a_i b_i \times a_i c_i\right| \quad \text{(A.1.3)}$$

Where $A = \sum_{i=1}^{N_f} A_i$

The edges of the *i-th* triangle are represented by $a_i b_i$ and $a_i c_i$, where *ai*, *bi* and c_i are the vertices.

A_i is the surface area of each triangle, and A is the sum of all calculated sub-areas.

A.1.4 Surface Area to volume ratio

$$surface\ to\ volume\ ratio = \frac{A}{V} \quad \text{(A.1.4)}$$

This feature indicates with its value how much sphere-like the shape is.

A.1.5 Sphericity

$$sphericity = \frac{\sqrt[3]{36\pi V^2}}{A} \quad \text{(A.1.5)}$$

This feature measure the roundness of the shape of the tumor region, with a value dimensionless and orientation-independent, scale-independent. The *sphericity* is between 0 and 1 $(0 < sphericity \leq 1)$, where for 1 there are a perfect sphere.

A.1.6 Compactness 1

$$compactness\,1 = \frac{V}{\sqrt{\pi A^3}} \tag{A.1.6}$$

Following the definition: $compactness\,1 = \frac{1}{6\pi}\sqrt{compactness\,2} = \frac{1}{6\pi}\sqrt{sphericity^3}$.

This feature measure how the tumor shape is compact (relative to a sphere). Similar to *sphericity*, $0 < compactness\,1 \leq 1/6\pi$, where for $1/6\pi$ indicates a perfect sphere.

A.1.7 Compactness 2

$$compactness\,2 = 36\pi\frac{V^2}{A^3} = (sphericity)^3 \tag{A.1.7}$$

Similar to *compactness* 1, this feature is independent of scale and orientation and also is dimensionless. *Compactness* 2 is defined in the range $0 < compactness\,2 \leq 1$ where for 1 indicates a perfect sphere.

A.1.8 Spherical disproportion

$$spherical\ disproportion = \frac{A}{4\pi R^2} = \frac{A}{\sqrt[3]{36\pi V^2}}$$

$$and \tag{A.1.8}$$

$$R = \sqrt[3]{\frac{3V}{4\pi}}$$

With *R*, the radius of the sphere of the same volume of the tumor. Spherical Disproportion is a ratio obtained by the surface of the tumor region on the surface of the same tumor region. By definition is the inverse of *sphericity*. The *spherical disproportion* range is $spherical\ disproportion \geq 1$, with a value 1 indicates a perfect sphere.

A.1.9 Maximum 3D diameter

This feature is defined as the largest pairwise Euclidean distance between the vertices of the tumor surface mesh. It is also called Feret Diameter.

A.1.10 Maximum 2D diameter (Slice)

This feature is defined as the largest pairwise Euclidean distance between the vertices of the tumor surface mesh, usually in the axial plane (row-column plane).

A.1.11 Maximum 2D diameter (Column)

This feature is defined as the largest pairwise Euclidean distance between the vertices of the tumor surface mesh, usually in the coronal plane (row-slice plane).

A.1.12 Maximum 2D diameter (Row)

This feature is defined as the largest pairwise Euclidean distance between the vertices of the tumor surface mesh, usually in the sagittal plane (column-slice plane).

A.1.13 Major Axis Length

$$\text{major axis} = 4\sqrt{\lambda_{major}} \tag{A.1.13}$$

This feature yields the largest axis length of the ROI-enclosing ellipsoid, calculated using the largest principal component λ_{major} , and the principal component is performed using the coordinate centers of the voxels defining the ROI.

A.1.14 Minor Axis Length

$$minor\ axis = 4\sqrt{\lambda_{minor}} \tag{A.1.14}$$

This feature yields the second largest axis length of the ROI-enclosing ellipsoid, calculated using the largest principal component , λ_{minor} , and the principal component is performed using the coordinate centers of the voxels defining the ROI.

A.1.15 Least Axis Length

$$least\ axis = 4\sqrt{\lambda_{least}} \tag{A.1.15}$$

This feature yields the smallest axis length of the ROI-enclosing ellipsoid, calculated using the largest principal component λ_{least} , for 2D segmentation this value is 0. The principal component is performed using the coordinate centers of the voxels defining the ROI.

A.1.16 Elongation

The elongation feature shows the ratio between the largest principal components of the ROI shape. This feature is defined as the inverse of true elongation.

$$elongation = \sqrt{\frac{\lambda_{minor}}{\lambda_{major}}} \tag{A.1.16}$$

The length of the largest principal components (λ_{major}) and the length of the second largest principal components (λ_{minor}) are used in a ratio. The range of this feature is. The principal component is performed using the coordinate centers of the voxels defining the ROI. It, therefore, takes spacing into account, but does not make use of shape mesh-like previous λ .

A.1.17 Flatness

$$flateness = \sqrt{\frac{\lambda_{least}}{\lambda_{major}}} \tag{A.1.17}$$

Shows relationship between the largest and the smallest principal component axes. This feature present the following range of values $0 \leq flatness \leq 1$, where for 0 there is a flat object and for 1 a sphere-like object. The principal component is performed using the coordinate centers of the voxels defining the ROI. It, therefore, takes spacing into account, but does not make use of shape mesh-like previous λ .

A.2 First order statistics features

The First order statistic features describe the distribution intensities within by a mask through commonly used and basic metrics.

Let's introduce some values:

- X be a set of N_p voxels in the Region Of Interest.
- $P(i)$ be the first histogram with N_g discrete intensity levels and it indicates the non-zero bins equally spaced.
- $p(i)$ be the normalized first order histogram and equal to $\frac{P(i)}{N_p}$.

A.2.1 Energy

$$energy = \sum_{i=1}^{N_p} (X(i) + c)^2 \tag{A.2.1}$$

Whit c, optional value to prevent negative values, with gray level intensity closest to 0. Energy is the magnitude of the voxels in an image.

A.2.2 Total Energy

$$total\ energy = V_{voxel} \sum_{i=1}^{N_p} (X(i) + c)^2 \tag{A.2.2}$$

This feature is scaled by the volume of the voxel in mm^3.

A.2.3 Entropy

$$entropy = -\sum_{i=1}^{N_g} p(i)\log_2\left(p(i)+\in\right) \qquad \text{(A.2.3)}$$

Here, is measured the amount of information required to encode the image values. The entropy feature specifies the uncertainty/randomness in the image values.

A.2.4 Minimum

$$minimum = \min\left(X\right) \qquad \text{(A.2.4)}$$

The minimum intensity value of X .

A.2.5 10^{th} percentile

The 10th percentile of X .

A.2.6 90^{th} percentile

The 90th percentile of X .

A.2.7 Maximum

$$maximum = \max\left(X\right) \qquad \text{(A.2.7)}$$

The maximum intensity value of X , is the maximum gray level intensity in the ROI.

A.2.8 Mean

$$mean = \frac{1}{N_p}\sum_{i=1}^{N_p} X\left(i\right) \qquad \text{(A.2.8)}$$

Is the average grey level intensity of the ROI.

A.2.9 Median

Is the median grey level intensity of the ROI.

A.2.10 Interquartile range

$$interquartile\ range = P_{75} - P_{25} \qquad \text{(A.2.10)}$$

Where P_{25} *and* P_{75} are the 25^{th} and 75^{th} percentile of the image array, respectively.

A.2.11 Range

$$range = \max(X) - \min(X) \tag{A.2.11}$$

This feature represents the range of gray levels in the ROI.

A.2.12 Mean Absolute Deviation (MAD)

$$MAD = \frac{1}{N_p}\sum_{i=1}^{N_p} | X(i) - \bar{X} | \tag{A.2.12}$$

This feature represents the mean distance of all intensity values from Mean Value of the image array.

A.2.13 Robust Mean Absolute Deviation (rMAD)

$$rMAD = \frac{1}{N_{10-90}}\sum_{i=1}^{N_{10-90}} | X_{10-90}(i) - \bar{X}_{10-90}(i) | \tag{A.2.13}$$

This feature represent the mean distance of all intensity values from Mean Value. It's calculated on the subset of image array with gray level in between or equal to 10^{th} and 90^{th} .

A.2.14 Root Mean Squared (RMS)

$$RMS = \sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p}\left(X(i) + c\right)^2} \tag{A.2.14}$$

This feature is another measure of the magnitude of the image values. It is a volume-confounded feature and a high value of c increase that effect. The optional value *c* is implemented to prevent negative values in $\boldsymbol{X}$.

A.2.15 Standard deviation

$$standard\ deviation = \sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p}\left(X(i) - \bar{X}\right)^2} \tag{A.2.15}$$

The standard deviation measures the amount of variation from the mean value.

A.2.16 Skewness

$$skewness = \frac{\mu_3}{\sigma^3} = \frac{\frac{1}{N_p}\sum_{i=1}^{N_p}\left(X(i)-\bar{X}\right)^3}{\left(\sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p}\left(X(i)-\bar{X}\right)^2}\right)^3} \quad \text{(A.2.16)}$$

This feature, characterized by the 3rd central moment μ_3, represent the asymmetry of the distribution about the mean value depending on where the tail is elongated and the mass of the distribution is concentrated and can assume positive or negative values.

A.2.17 Kurtosis

$$kurtosis = \frac{\mu_4}{\sigma^4} = \frac{\frac{1}{N_p}\sum_{i=1}^{N_p}\left(X(i)-\bar{X}\right)^4}{\left(\sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p}\left(X(i)-\bar{X}\right)^2}\right)^2} \quad \text{(A.2.17)}$$

This feature, characterized by the 4th central moment μ_4, represent the peakedness of the distribution in the image of the ROI on where the mass of the distribution is concentrated towards the tail(s) rather than toward mean. A lower value of Kurtosis implies the vice-versa, with the mass concentrated around the mean value.

A.2.18 Variance

$$variance = \frac{1}{N_p}\sum_{i=1}^{N_p}\left(X(i)-\bar{X}\right)^2 \quad \text{(A.2.18)}$$

The variance represents the mean of the squared distances of each intensity value from the mean. This is a measure of the spread of the distribution about the mean.

A.2.19 Uniformity

$$uniformity = \sum_{i=1}^{N_g} p(i)^2 \quad \text{(A.2.19)}$$

This feature measures the sum of the square of each intensity value. It gives a measure of the homogeneity of the image array, where if there is a greater uniformity it's associated with a greater homogeneity or a smaller range of discrete values of the intensity level.

A.3 Gray Level Co-Occurrence Matrix (GLCM) features

A gray level Co-occurrence Matrix (GLCM) of size $N_g \times N_g$ is used to describe the second-order joint probability function using a mask on a region and is defined by $P(i, j \mid \delta, \theta)$. The element $(i,j)^{\text{the}}$ represent the number of times the combination of level *i* and *j* occur in two pixels in the image, separated by a distance of δ along an angle θ. For $\delta = 1$ this result in 2 neighbors for each of 13 angles in 3D (26-connectivity) and for $\delta = 2$ is 98-connectivity (49 unique angles). As two dimensional example, let the following matrix **I** represent an 5×5 image, having 5 discrete grey levels:

$$\mathrm{I} = \begin{bmatrix} 1 & 2 & 5 & 2 & 3 \\ 3 & 2 & 1 & 3 & 1 \\ 1 & 3 & 5 & 5 & 2 \\ 1 & 1 & 1 & 1 & 2 \\ 1 & 2 & 4 & 3 & 5 \end{bmatrix}$$

and for distance $\delta = 1$ and $\theta = 0$ the symmetrical GLCM is

$$\mathrm{P} = \begin{bmatrix} 6 & 4 & 3 & 0 & 0 \\ 4 & 0 & 2 & 1 & 3 \\ 3 & 2 & 0 & 1 & 2 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 3 & 2 & 0 & 2 \end{bmatrix}$$

Now the following values are introduced:

- $\in$ be an arbitrarily small positive number ($\approx 10^{-6}$)
- $P(i, j)$ is the co-occurrence matrix for an arbitrary value of δ and θ.
- $p(i, j)$ is the normalized co-occurrence matrix and $p(i, j) = \dfrac{P(i, j)}{\sum P(i, j)}$
- N_g is the number of discrete intensity levels in the image.
- $p_x(i) = \sum_{j=1}^{N_g} P(i, j)$ represent the marginal row probabilities
- $p_x(j) = \sum_{i=1}^{N_g} P(i, j)$ represent the marginal column probabilities

- μ_x represent the mean gray level intensity of p_x and defined as $\mu_x = \sum_{i=1}^{N_g} p_x(i) i$
- μ_y represent the mean gray level intensity of p_y and defined as $\mu_y = \sum_{j=1}^{N_g} p_y(j) j$
- σ_x is the standard deviation of p_x
- σ_y is the standard deviation of p_y
- $p_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)$, where $i + j = k$, with k=2,3,…, $2N_g$
- $p_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)$, where $|i - j| = k$, with k=0,1,…, $N_g - 1$
- $HX = -\sum_{i=1}^{N_g} p_x(i) \log_2 \left(p_x(i) + \in \right)$ is the entropy of p_x
- $HY = -\sum_{j=1}^{N_g} p_y(j) \log_2 \left(p_y(j) + \in \right)$ is the entropy of p_y
- $HXY = -\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \log_2 \left(p(i,j) + \in \right)$ is the entropy of $p(i,j)$
- $HXY1 = -\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \log_2 \left(p_x(i) p_y(j) + \in \right)$
- $HXY2 = -\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_x(i) p_y(j) \log_2 \left(p_x(i) p_y(j) + \in \right)$

The feature value is calculated on the GLCM for each angle separately, after the mean of these value return. The GLCM matrix shows the weighting factor W and then summed and normalized, after that the feature is calculated on the resultant matrix. W is defined as the distance between neighboring voxel following the formula $W = e^{-\|d\|^2}$ with *d*, the distance for the associated angle according to the norm specified in setting *weighting-Norm*.

The following class specific setting is possible:

- distances[[1]]: List of integers. Provides the distances between the center voxel and the neighbor, for which angles should be generated.
- SymmetricalGLCM[True]: Boolean. It indicates if the co-occurrences should be assessed in two directions per angle giving a symmetrical matrix with equal distribution *x* and *y*.
- weightingNorm[None]: string. It indicates what norm use when is applied the distance weighted. Enumerated setting present the following values:
- *manhattan*: first order normalization
- *Euclidean*: second order normalization
- *Infinity*: infinity normalization
- *no_weighting*: GLCMs are weighted by factor 1 and summed
- None: no-weight applied, the mean of values is calculated on separate matrices is returned.

For other values, a warning is logged and is used the *no_weighting.*

A.3.1 Autocorrelation

$$aucorrelation = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)ij \quad \text{(A.3.1)}$$

This feature measures the magnitude of the fineness and coarseness of texture.

A.3.2 Joint Average

$$joint\ average = \mu_x = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)i \quad \text{(A.3.2)}$$

This feature returns the mean grey level of the *i*-distribution, and it is independent of the *j*-distribution

A.3.3 Cluster Prominence

$$cluster\ \ prominence = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\left(i + j - \mu_x - \mu_y\right)^4 p(i,j) \quad \text{(A.3.3)}$$

This feature is a measure of the skewness and the asymmetry of the GLCM. A higher value imply asymmetry about the mean and a lower value indicates a peak near the mean value.

A.3.4 Cluster Shade

$$cluster\ shade = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\left(i + j - \mu_x - \mu_y\right)^3 p(i,j) \quad \text{(A.3.4)}$$

This feature is the measure of the skewness and uniformity of the GLCM. High cluster shade value implies greater asymmetry about the mean.

A.3.5 Cluster Tendency

$$cluster\ tendency = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\left(i + j - \mu_x - \mu_y\right)^2 p(i,j) \quad \text{(A.3.5)}$$

This feature is a measure of groupings of the voxel with similar gray level values.

A.3.6 Contrast

$$contrast = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-j)^2 p(i,j) \quad \text{(A.3.6)}$$

This feature measures the local intensity variation, favoring values away from the diagonal where *i=j*.

A.3.7 Correlation

$$correlation = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j) i j - \mu_x \mu_y}{\sigma_x(i)\sigma_y(j)} \quad \text{(A.3.7)}$$

This feature measure values correlation, the value assumed are between 0 (uncorrelated) and 1 (perfectly correlated).

A.3.8 Difference Average

$$difference\ averange = \sum_{k=0}^{N_g-1} k p_{x-y}(k) \quad \text{(A.3.8)}$$

This feature measures the relationship between occurrences of pairs with similar intensity values and occurrences of pairs with differing intensity values.

A.3.9 Difference Entropy

$$difference\ entropy = \sum_{k=0}^{N_g-1} p_{x-y}(k)\log_2\left(p_{x-y}(k)+\in\right) \quad \text{(A.3.9)}$$

This feature measures the randomness/variability in neighborhood intensity value difference.

A.3.10 Difference Variance

$$difference\ variance = \sum_{k=0}^{N_g-1} \left(k - DA\right)^2 p_{x-y}(k) \quad \text{(A.3.10)}$$

This feature measures the heterogeneity, where places higher weights on differing intensity level pairs that deviate more from the mean.

A.3.11 Dissimilarity (DEPRECATED)

$$dissimilarity = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} |\, i-j\,|\, p(i,j) \quad \text{(A.3.11)}$$

Mathematically equal to Difference Average.

A.3.12 Joint energy

$$joint\ energy = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\left(p(i,j)\right)^2 \quad \text{(A.3.12)}$$

The energy is a measure of homogeneity of the pattern in an image. A great value of this feature implies that there are more instances of intensity value pairs in the image that neighbor each other at higher frequencies.

A.3.13 Joint entropy

$$joint\ entropy = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\log_2\left(p(i,j)+\in\right) \quad \text{(A.3.13)}$$

This feature is a measure of randomness/variability in neighborhood intensity values.

A.3.14 Homogeneity 1 (DEPRECATED)

$$homogeneity\,1 = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{p(i,j)}{1+|\,i-j\,|} \quad \text{(A.3.14)}$$

This feature is mathematically equal to Inverse Difference.

A.3.15 Homogeneity 2 (DEPRECATED)

$$homogeneity\ 2 = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{p(i,j)}{1+|i-j|^2} \quad \text{(A.3.15)}$$

This feature is mathematically equal to Inverse Difference Moment.

A.3.16 Information Measure of Correlation (IMC 1)

$$IMC\,1 = \frac{HXY - HXY1}{\max\{HX, HY\}} \quad \text{(A.3.16)}$$

This feature assesses the correlation between the probability distribution of *i* and the probability distribution of *j*, and quantifies the complexity of the texture, using mutual information I(x,y):

$$\begin{aligned} I(i,j) &= \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\log_2\left(\frac{p(i,j)}{p_x(i)p_y(j)}\right) \\ &= \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\left(\log_2(p(i,j)) - \log_2\left(p_x(i)p_y(j)\right)\right) \\ &= \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\log_2(p(i,j)) - \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\log_2\left(p_x(i)p_y(j)\right) \\ &= -HXY + HXY1 \end{aligned} \quad \text{(A.3.16)}$$

The numerator of *IMC* 1 is equal to –*I*(*i*, *j*) and is divided by the maximum of the two marginal entropies.

A.3.17 Information Measure of Correlation (IMC 2)

$$IMC\ 2 = \sqrt{1 - e^{-2(HXY2 - HXY)}} \quad \text{(A.3.17)}$$

This feature also assessed the correlation between the probability distribution of *i and* j, and it quantifies the complexity of the texture.

A.3.18 Inverse differences moment (IDM)

$$IDM = \sum_{k=0}^{N_g-1}\frac{p_{x-y}(k)}{1+k^2} \quad \text{(A.3.18)}$$

This feature is equal to *Homogeneity* 2, and it measures the local homogeneity of an image. IDM weights are the inverse of the Contrast weights.

A.3.19 Maximal correlation coefficient (MCC)

$$MCC = \sqrt{second\ largest\ eigenvalue\ of\ Q} \tag{A.3.19}$$

$$Q(i,j) = \sum_{k=0}^{N_g} \frac{p(i,k)\,p(j,k)}{p_x(i)\,p_y(j)}$$

This Feature measures the complexity of the texture and. For a flat region, each GLCM matrix has a shape (1, 1), resulting in just 1 eigenvalue.

A.3.20 Inverse Difference Moment Normalized (IDMN)

$$IDMN = \sum_{k=0}^{N_g-1} \frac{p_{x-y}(k)}{1+\left(\frac{k}{N_g}\right)^2} \tag{A.3.20}$$

This feature is a measure of the local homogeneity of an image where the IDMN weights are the inverse of the Contrast weights. The IDMN normalizes the square of the difference between neighbring intensity values.

A.3.21 Inverse Difference (ID)

$$ID = \sum_{k=0}^{N_g-1} \frac{p_{x-y}(k)}{1+k} \tag{A.3.21}$$

This feature is another measure of the local homogeneity of an image.

A.3.22 Inverse Difference Normalized (IDN)

$$IDN = \sum_{k=0}^{N_g-1} \frac{p_{x-y}(k)}{1+\left(\frac{k}{N_g}\right)} \tag{A.3.22}$$

The IDN normalizes the square of the difference between neighbring intensity values.

A.3.23 Inverse Variance

$$inverse\ variance = \sum_{k=1}^{N_g-1} \frac{p_{x-y}(k)}{k^2} \tag{A.3.23}$$

Where k=0 is skipped, to avoid the division by 0.

A.3.24 Maximum Probability

$$maximum\ probability = \max(p(i,j)) \tag{A.3.24}$$

This feature is the occurrence of the most predominant pair of neighboring intensity values.

A.3.25 Sum Average (DEPRECATED)

$$sum\ average = \sum_{k=2}^{2N_g} p_{x+y}(k)k \quad \text{(A.3.25)}$$

This feature measures the relationship between the occurrence of pairs with lower intensity values and the occurrence of pairs with higher intensity values.

A.3.26 Sum Variance

$$sum\ variance = \sum_{k=2}^{2N_g} \left(k - SA\right)^2 p_{x+y}(k) \quad \text{(A.3.26)}$$

This feature is mathematically equal to Cluster Tendency.

A.3.27 Sum Entropy

$$sum\ entropy = \sum_{i=1}^{2N_g} p_{x+y}(k) \log_2 \left(p_{x+y}(k) + \in \right) \quad \text{(A.3.27)}$$

This feature is a sum of neighborhood intensity value differences.

A.3.28 Sum of Squares

$$sum\ squares = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu_x)^2 \, p(i, j) \quad \text{(A.3.28)}$$

This feature (or Variance) is a measure in the distribution of neighboring intensity level pairs about the mean intensity level in the GLCM.

A.4 Gray Level Run Lenght Matrix (GRLM)

A GLRLM, Gray Level Run Length Matrix, quantifies gray level runs of consecutive pixels that have the same gray level value. In a gray level run length matrix $P(i, j \mid \theta)$, the $(i, j)^{th}$ element describes the number of runs with gray level i and length j occur in the image (ROI) along the angle θ. For instance, with a two-dimensional example, let's consider the following 5×5 image, with 5 discrete gray levels:

$$I = \begin{bmatrix} 5 & 2 & 5 & 4 & 4 \\ 3 & 3 & 3 & 1 & 3 \\ 2 & 1 & 1 & 1 & 3 \\ 4 & 2 & 2 & 2 & 3 \\ 3 & 5 & 3 & 3 & 2 \end{bmatrix}$$

With the GRLM for $\theta = 0$, 0 degrees indicate the horizontal direction, we obtain:

$$P = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 4 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Now the following values are introduced:

- N_g is the number of discrete intensity levels in the image.
- N_r is the number of discrete run lengths in the image.
- N_p is the number of voxels in the image.
- $N_z(\theta)$ is the number of runs in the image along angle θ, and is equal to $\sum_{i=1}^{N_g}\sum_{j=1}^{N_r} P(i, j \mid \theta)$ and $1 \le N_z(\theta) \le N_p$
- $P(i, j \mid \theta)$ is the run length matrix for an arbitrary direction θ
- $p(i, j \mid \theta)$ is the normalized run length matrix as:

$$p(i, j \mid \theta) = \frac{P(i, j \mid \theta)}{N_z(\theta)}$$

The feature value is calculated on the GLCM for each angle separately, after the mean of these value return. If distance weighting is enabled. The GLRLMs are weighted by the distance between neighboring voxels and then summed and normalized, The features are calculated on the resultant matrix. The distance between neighboring voxels is calculated for each angle using the norm specified in *weightingNorm*.

The following class specific setting is possible:

- weightingNorm[None]: string. It indicates what norm use when is applied the distance weighted. Enumerated setting present the following values:
    - *manhattan*: first order normalization
    - *Euclidean*: second order normalization
    - *Infinity*: infinity normalization
    - *no_weighting*: GLCMs are weighted by factor 1 and summed
    - None: no-weight applied, the mean of values is calculated on separate matrices is returned.

For other values, a warning is logged and is used the *no_weighting*.

A.4.1 Short Run Emphasis (SRE)

$$SRE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{P(i,j\mid\theta)}{j^2}}{N_z(\theta)} \quad \text{(A.4.1)}$$

SRE is a measure of the distribution of short-run lengths, with a greater value indicative of shorter run lengths and more fine textures.

A.4.2 Long Run Emphasis (LRE)

$$LRE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j\mid\theta)\,j^2}{N_z(\theta)} \quad \text{(A.4.2)}$$

This feature measures the distribution of long-run lengths. A greater value indicates longer run lengths and more coarse structural textures.

A.4.3 Gray Level Non-Uniformity (GLN)

$$GLN = \frac{\sum_{i=1}^{N_g}\left(\sum_{j=1}^{N_r}P(i,j\mid\theta)\right)^2}{N_z(\theta)} \quad \text{(A.4.3)}$$

This feature measures the similarity of gray-level intensity in the image. A lower GLN value correlates with a greater similarity in intensity values.

A.4.4 Gray Level Non-Uniformity Normalized (GLNN)

$$GLNN = \frac{\sum_{i=1}^{N_g}\left(\sum_{j=1}^{N_r} P(i,j\mid\theta)\right)^2}{N_z(\theta)^2} \quad \text{(A.4.4)}$$

This is the normalized version of the GLN formula.

A.4.5 Run Length Non-Uniformity (RLN)

$$RLN = \frac{\sum_{j=1}^{N_r}\left(\sum_{i=1}^{N_g} P(i,j\mid\theta)\right)^2}{N_z(\theta)} \quad \text{(A.4.5)}$$

This feature measures the similarity of runs throughout the image. A lower value indicates more homogeneity among run lengths in the image.

A.4.6 Run Length Non-Uniformity Normalized (RLNN)

$$RLNN = \frac{\sum_{j=1}^{N_r}\left(\sum_{i=1}^{N_g} P(i,j\mid\theta)\right)^2}{N_z(\theta)^2} \quad \text{(A.4.6)}$$

This is the normalized version of the RLN formula.

A.4.7 Run Percentage (RP)

$$RP = \frac{N_z(\theta)}{N_p} \quad \text{(A.4.7)}$$

This feature measures the coarseness of the texture by taking the ratio of a number of runs and the number of voxels in the region of interest (ROI). The range of this value is # where a high value indicates a larger portion of ROI consist of short runs.

A.4.8 Gray Level Variance (GLV)

$$GLV = \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} p(i, j \mid \theta)(i - \mu)^2 \quad \text{(A.4.8)}$$

This feature measures the variance in gray level intensity for the runs.

A.4.9 Run Variance (RV)

$$RV = \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} p(i, j \mid \theta)(j - \mu)^2 \quad \text{(A.4.9)}$$

$$\mu = \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} p(i, j \mid \theta) j$$

This feature measures the variance in runs for the run lengths.

A.4.10 Run Entropy (RE)

$$RE = -\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} p(i, j \mid \theta) \log_2 (p(i, j \mid \theta) + \in) \quad \text{(A.4.10)}$$

$\in$ is an arbitrarily small positive value number ($\approx 10^{-16}$).

This feature measures the uncertainty/randomness in the distribution lengths and gray levels.

A.4.11 Low Gray Level Run Emphasis (LGLRE)

$$LGLRE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} \frac{P(i, j \mid \theta)}{i^2}}{N_z(\theta)} \quad \text{(A.4.11)}$$

This feature measures the distribution of lower gray level value, where a higher value indicates a greater concentration of low gray level values in the image.

A.4.12 High Gray Level Run Emphasis (HGLRE)

$$HGLRE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} P(i, j \mid \theta) i^2}{N_z(\theta)} \quad \text{(A.4.12)}$$

This feature measures the distribution of higher gray level value, where a higher value indicates a greater concentration of high gray level values in the image.

A.4.13 Short Run Gray- Level Emphasis (SRLGLE)

$$SRLGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{P(i,j|\theta)}{i^2 j^2}}{N_z(\theta)} \quad \text{(A.4.13)}$$

This feature measures the joint distribution of shorter run lengths with lower gray level values

A.4.14 Short Run High Gray- Level Emphasis (SRHGLE)

$$SRHGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{P(i,j|\theta)i^2}{j^2}}{N_z(\theta)} \text{(A.4.14)}$$

This feature measures the joint distribution of shorter run lengths with higher gray level values.

A.4.15 Long Run Low Gray- Level Emphasis (LRLGLE)

$$LRLGLRE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{P(i,j|\theta)j^2}{i^2}}{N_z(\theta)} \quad \text{(A.4.15)}$$

This feature measures the joint distribution of long-run lengths with lower gray levels values.

A.4.16 Long Run High Gray- Level Emphasis (LRHGLE)

$$LRHGLRE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j|\theta)i^2 j^2}{N_z(\theta)} \quad \text{(A.4.16)}$$

This feature measures the joint distribution of long-run lengths with higher gray level values.

A.5 Gray Level Size Zone Matrix (GLSZM) Features

A GLSZM, Gray Size Zone Matrix, quantifies gray level zones in an image. A grey level zone is defined by the number of connected voxels that have the same gray intensity value. A voxel is considered connected if the distance is 1 according to the infinity norm. In a gray level size zone matrix $P(i,j|\theta)$, the $(i,j)^{th}$ element equals

the number of zones with gray level *i* and length *j* occur in the image (ROI). GLSZM is rotation independent.

For instance, with a two-dimensional example, let's consider the following 5×5 image, with 5 discrete gray levels:

$$I = \begin{bmatrix} 5 & 2 & 5 & 4 & 4 \\ 3 & 3 & 3 & 1 & 3 \\ 2 & 1 & 1 & 1 & 3 \\ 4 & 2 & 2 & 2 & 3 \\ 3 & 5 & 3 & 3 & 2 \end{bmatrix}$$

We obtain:

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Now the following values are introduced:

- N_g is the number of discrete intensity levels in the image.
- N_r is the number of discrete run lengths in the image.
- N_p is the number of voxels in the image.
- $N_z(\theta)$ is the number of runs in the image along angle θ, and is equal to $\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} P(i, j \mid \theta)$ and $1 \leq N_z(\theta) \leq N_p$
- $P(i, j)$ is the size zone matrix.
- $p(i, j)$ is the normalized zone matrix as:

$$p(i, j) = \frac{P(i, j)}{N_z(\theta)}$$

A.5.1 Small Area Emphasis (SAE)

$$SAE = \frac{\sum_{i=1}^{N_z} \sum_{j=1}^{N_s} \frac{P(i,j)}{j^2}}{N_z} \quad \text{(A.5.1)}$$

This feature is the measure of the distribution of the small zones, a higher value indicates smaller size zones an more fine textures.

A.5.2 Large Area Emphasis (LAE)

$$LAE = \frac{\sum_{i=1}^{N_z} \sum_{j=1}^{N_s} P(i,j)\, j^2}{N_z} \quad \text{(A.5.2)}$$

This feature measures the distribution of large area size zones, with a higher value indicating low homogeneity in intensity values.

A.5.3 Gray Level Non-Uniformity (GLN)

$$GLN = \frac{\sum_{i=1}^{N_z} \left(\sum_{j=1}^{N_s} P(i,j) \right)^2}{N_z} \text{(A.5.3)}$$

This feature measures the variability of gray-level intensity values in the image, with a higher value indicating low homogeneity in intensity values.

A.5.4 Gray Level Non-Uniformity Normalized (GLNN)

$$GLNN = \frac{\sum_{i=1}^{N_z} \sum_{j=1}^{N_s} \left(P(i,j) \right)^2}{{N_z}^2} \quad \text{(A.5.4)}$$

This is the normalized version of the GLN formula.

A.5.5 Size Zone Non-Uniformity (SZN)

$$SZN = \frac{\sum_{j=1}^{N_s}\left(\sum_{i=1}^{N_g} P(i,j)\right)^2}{N_z} \quad \text{(A.5.5)}$$

This feature measures the variability of size zone volumes in the image, with a lower value indicating more homogeneity in size zone volumes.

A.5.6 Size Zone Non-Uniformity Normalized (SZNN)

$$SZNN = \frac{\sum_{j=1}^{N_s}\left(\sum_{i=1}^{N_g} P(i,j)\right)^2}{N_z^{\,2}} \quad \text{(A.5.6)}$$

This is the normalized version of the SZN formula.

A.5.7 Zone Percentage (ZP)

$$ZP = \frac{N_z}{N_p} \quad \text{(A.5.7)}$$

This feature is the measure of coarseness of the texture by taking the ratio of a number of zones and number of voxels in the ROI. The values of *ZP* are in the range $\frac{1}{N_p} \le ZP \le 1$, with a higher value indicating a larger portion of the ROI consist of small zones and a more fine texture.

A.5.8 Gray Level Variance (GLV)

$$GLV = \sum_{i=1}^{N_g}\sum_{j=1}^{N_s} p(i,j)(i-\mu)^2 \quad \text{(A.5.8)}$$

Where $\mu = \sum_{i=1}^{N_g}\sum_{j=1}^{N_s} p(i,j)i$

The feature GLV measure the variance in zone size volumes for the zones.

A.5.9 Zone Variance (ZV)

$$ZV = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_s} p(i,j)(j-\mu)^2 \quad \text{(A.5.9)}$$

Where $\mu = \sum_{i=1}^{N_g}\sum_{j=1}^{N_s} p(i,j)\,j$

This feature measure the variance in zone size volumes for the zones.

A.5.10 Zone Entropy (ZE)

$$ZE = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_s} p(i,j)\log_2\left(p(i,j)+\in\right) \quad \text{(A.5.10)}$$

$\in$ is an arbitrarily small positive value number ($\approx 10^{-16}$).

This feature measures the uncertainty/randomness in the distribution of zone sizes and gray and gray levels.

A.5.11 Low Gray Level Zone Emphasis (LGLZE)

$$LGLZE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}\frac{P(i,j)}{i^2}}{N_z} \quad \text{(A.5.11)}$$

This feature measures the distribution of lower gray level size zones, a higher value indicating a greater proportion of lower gray level value and size zones in the image.

A.5.12 High Gray Level Zone Emphasis (HGLZE)

$$HGLZE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}P(i,j)i^2}{N_z} \quad \text{(A.5.12)}$$

This feature measures the distribution of higher gray level size zones, a higher value indicating a greater proportion of higher gray level value and size zones in the image.

A.5.13 Small Area Low Gray Level Zone Emphasis (SALGLE)

$$SALGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}\frac{P(i,j)}{i^2 j^2}}{N_z} \quad \text{(A.5.13)}$$

This feature measures the proportion in the image of the joint distribution of smaller size zones with lower gray level values.

A.5.14 Small Area High Gray Level Emphasis (SAHGLE)

$$SAHGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}\frac{P(i,j)i^2}{j^2}}{N_z} \quad \text{(A.5.14)}$$

This feature measures the proportion in the image of the joint distribution of smaller size zones with higher gray level values.

A.5.15 Large Area Low Gray Level Emphasis (LALGLE)

$$LALGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}\frac{P(i,j)\,j^2}{i^2}}{N_z} \quad \text{(A.5.15)}$$

This feature measures the proportion in the image of the joint distribution of larger size zones with lower gray level values.

A.5.16 Large Area High Gray Level Emphasis (LAHGLE)

$$LAHGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}P(i,j)\,i^2\,j^2}{N_z} \quad \text{(A.5.16)}$$

This feature measures the proportion in the image of the joint distribution of larger size zones with higher gray level values.

A.6 Neighbouring Gray Tone Difference Matrix (NGTDM)

A Neighbouring Gray Tone Difference Matrix quantifies the difference between a gray value and the average gray value of its neighbors within the distance δ. The sum of the absolute differences for gray level *i* is stored in a matrix, where $\boldsymbol{X}_{gl}$ is a set of segmented voxels and (j_x, j_y, j_z), then the average gray level of the neighborhood is:

$$\bar{A}_i = \bar{A}(j_x, j_y, j_z) =$$
$$\frac{1}{W}\sum_{k_x=-\delta}^{\delta}\sum_{k_y=-\delta}^{\delta}\sum_{k_z=-\delta}^{\delta}x_{gl}(j_x + k_{x,}\, j_y + k_{y,}\, j_z + k_z)$$

Where $(k_x, k_y, k_z) \neq (0,0,0)$ and $x_{gl}(j_x + k_{x,}, j_y + k_{y,} j_z + k_z) \in X_{gl}$

The value W is the number of voxels in the neighborhood that are also in X_{gl}.

For instance, with a two-dimensional example, let's consider the following 4×4 image, with 5 discrete gray levels:

$$I = \begin{bmatrix} 1 & 2 & 5 & 2 \\ 3 & 5 & 1 & 3 \\ 1 & 3 & 5 & 5 \\ 3 & 1 & 1 & 1 \end{bmatrix}$$

We obtain the following NGTDM:

i	n_i	p_i	s_i
1	6	0.375	13.35
2	2	0.125	2.00
3	4	0.25	2.63
4	0	0.00	0.00
5	4	0.25	10.075

6 pixels have gray level 1 and we have

$$s_1 = |1 - 10/3| + |1 - 30/8| + |1 - 15/5| + |1 - 13/5| + |1 - 15/5| + |1 - 11/3| = 13.35$$

For gray level 2 there are 2 pixels

$$s_2 = \left|2 - 15/5\right| + \left|2 - 15/5\right| = 2$$

And similar for gray values 3 and 5:

$$s_3 = \left|3 - 12/5\right| + \left|3 - 18/5\right| + \left|3 - 20/8\right| + \left|3 - 5/3\right| = 3.03$$

$$s_5 = \left|5 - 14/5\right| + \left|5 - 18/5\right| + \left|5 - 20/8\right| + \left|5 - 11/5\right| = 10.075$$

Where:

n_i is the number of voxels in X_{gl} with i, the gray level.

$N_{v,p}$ is the total number of voxels in X_{gl} and equal to $\sum n_i$. The value $N_{v,p} \leq N_p$, where N_p is the number of all the voxels in the ROI.

p_i is the gray level probability and equal to $\frac{n_i}{N_v}$

$$s_i = \begin{cases} \sum^{n_i} \left| i - \bar{A}_i \right| & \text{for } n_i \neq 0 \\ 0 & \text{for } n_i = 0 \end{cases}$$

N_g is the number of gray of discrete gray levels

$N_{g,p}$ is the number of gray levels for $p_i \neq 0$

The following class specific setting is possible:
distances[[1]]: List of integers. Provides the distances between the center voxel and the neighbor, for which angles should be generated.

A.6.1 Coarseness

$$coarseness = \frac{1}{\sum_{i=1}^{N_g} p_i s_i} \quad \text{(A.6.1)}$$

This feature is the measure of the average difference between the central voxel and its neighborhood and is an indication of the spatial rate of change. A higher value indicates a lower spatial change rate and a locally more uniform texture.

A.6.2 Contrast

$$contrast = \left(\frac{1}{N_{g,p}(N_{g,p} - 1)} \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_i p_j (i-j)^2 \right) \left(\frac{1}{N_{v,p}} \sum_{i=1}^{N_g} s_i \right), \text{ where } p_i \neq 0,\ p_j \neq 0$$

(A.6.2)

This feature is the measure of the spatial intensity change and is dependent on the overall gray level dynamic range. The contrast is high when both the dynamic range and spatial change range are higher.

A.6.3 Busyness

$$busyness = \frac{\sum_{i=1}^{N_g} p_i s_i}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\left|ip_i - jp_j\right|} \quad , \quad \text{where } p_i \neq 0,\ p_j \neq 0 \quad \text{(A.6.3)}$$

This feature measures the change from a pixel to its neighbor. When this feature presents a high value indicates a busy image, with rapid changes of intensity between pixels and its neighborhood.

A.6.4 Complexity

$$complexity = \frac{1}{N_{v,p}}\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\left|i-j\right|\frac{p_i s_i + p_j s_j}{p_i + p_j}, \text{ where } p_i \neq 0,\ p_j \neq 0 \quad \text{(A.6.4)}$$

There is a complex image when there are many primitive components in the image, for example when there are many rapid changes in gray level intensity there is a non-uniform image.

A.6.5 Strength

$$strength = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\left|p_i - p_j\right|(i-j)^2}{\sum_{i=1}^{N_g} s_i}, \text{ where } p_i \neq 0,\ p_j \neq 0 \quad \text{(A.6.5)}$$

This feature is the measure of the primitive in an image. It presents a high value when the primitives are easily defined and visible.

A.7 Gray Level Dependence Matrix (GLDM)

$$I = \begin{bmatrix} 5 & 2 & 5 & 4 & 4 \\ 3 & 3 & 3 & 1 & 3 \\ 2 & 1 & 1 & 1 & 3 \\ 4 & 2 & 2 & 2 & 3 \\ 3 & 5 & 3 & 3 & 2 \end{bmatrix}$$ and for $\alpha = 0$ and $\delta = 1$ the GLDM is:

$$P = \begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 2 & 3 & 0 \\ 1 & 4 & 4 & 0 \\ 1 & 2 & 0 & 0 \\ 3 & 0 & 0 & 0 \end{bmatrix}$$

References

[1] https://pyradiomics.readthedocs.io/en/latest/features.html

[2] W. E. Lorensen, H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM SIGGRAPH Comput Graph Internet*, vol. 1987, pp. 163-9.

[3] R. Haralick, K. Shanmugan, I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 1973, pp. 610-621.

[4] https://en.wikipedia.org/wiki/Co-occurrence_matrix

[5] http://www.fp.ucalgary.ca/mhallbey/the_glcm.htm

[6] M.M. Galloway, "Texture analysis using gray level run lengths," *Computer Graphics and Image Processing*, vol. 4, pp. 172-179,1975.

[7] A. Chu, C.M. Sehgal, J. F. Greenleaf, "Use of gray value distribution of run length for texture analysis," *Pattern Recognition Letters*, vol. 11, pp. 415-419, 1990.

[8] D. Xu, A. Kurani, J. Furst, D. Raicu, "Run-Length Encoding For Volumetric Texture," *International Conference on Visualization, Imaging and Image Processing (VIIP)*, pp. 452-458, 2004.

[9] X. Tang, "Texture information in run-length matrices," *IEEE Transactions on Image Processing*, vol. 7, pp.1602-1609, 1998.

[10] N. Tustison, J. Gee, "Run-Length Matrices For Texture Analysis," *Insight Journal*, 2008.

[11] G. Thibault, B. Fertil, C. Navarro, S. Pereira, P. Cau, N. Levy, J. Sequeira, J. Mari, "Texture Indexes and Gray Level Size Zone Matrix. Application to Cell Nuclei Classification," *Pattern Recognition and Information Processing (PRIP),* pp. 140-145, 2009.

[12] https://en.wikipedia.org/wiki/Gray_level_size_zone_matrix

[13] M. Amadasun, R. King, "Textural features corresponding to textural properties. Systems, Man and Cybernetics," *IEEE Transactions*, vol. 19, pp. 1264-1274, doi: 10.1109/21.44046, 1989.

APPENDIX B

CASE 1: ANALYSIS WITH ONLY HANDCRAFTED RADIOMICS FEATURES

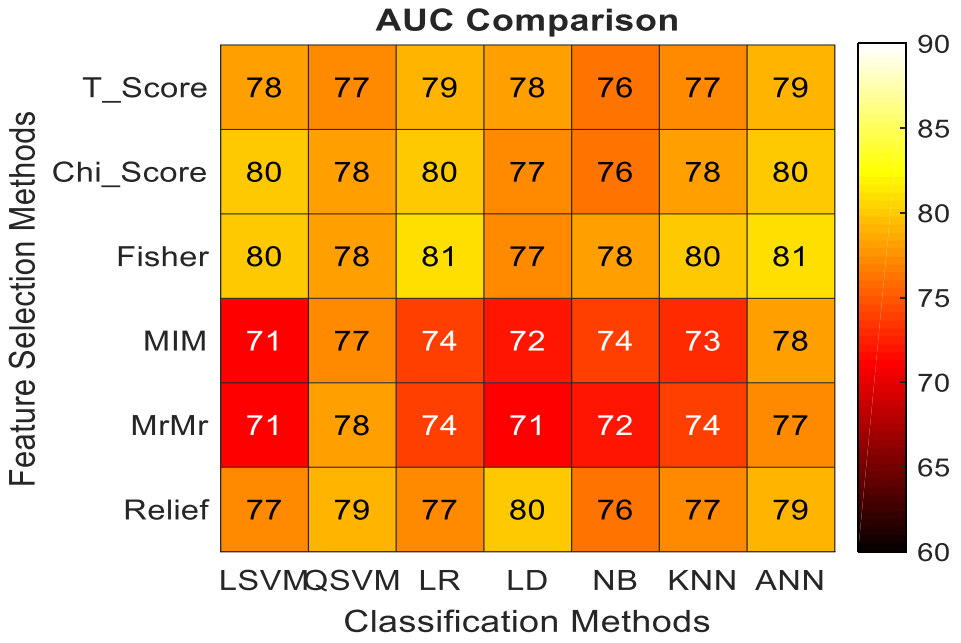


Figure 0.1: Heatmap representing the mode AUCs with size of selected features = 10; in rows for feature selection algorithms and, in columns for classification methods.

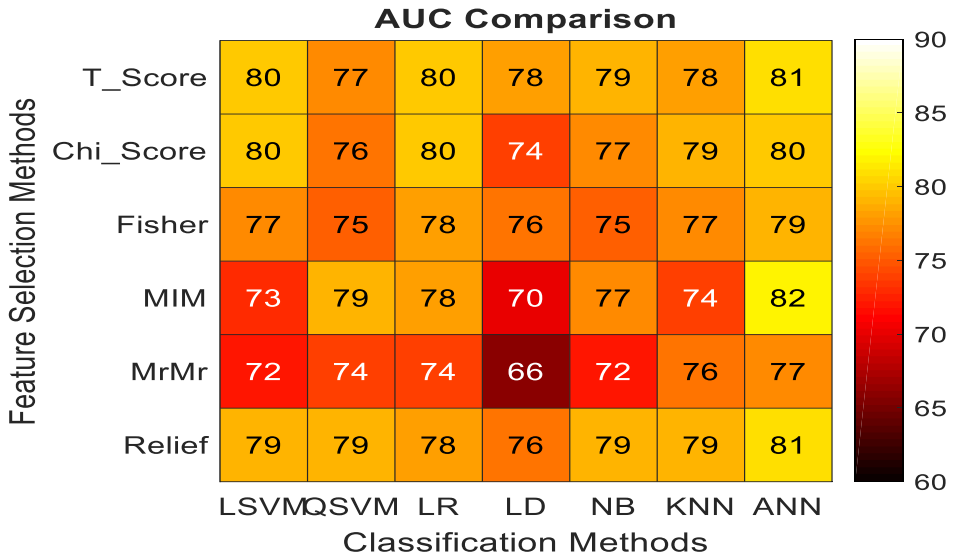


Figure 0.2: Heatmap representing the mode AUCs with size of selected features = 15; in rows for feature selection algorithms and, in columns for classification methods.

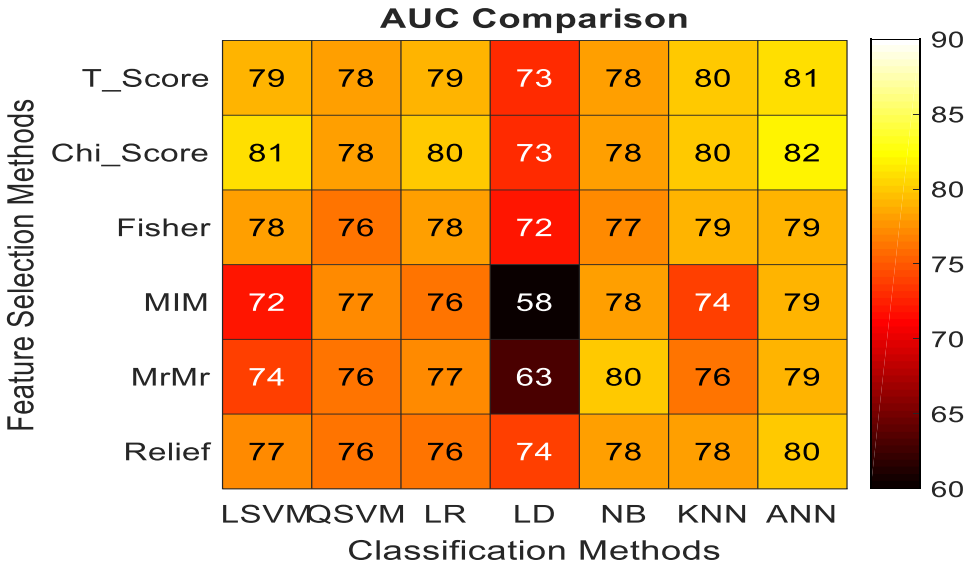


Figure 0.4: Heatmap representing the mode AUCs with size of selected features = 20; in rows for feature selection algorithms and, in columns for classification methods.

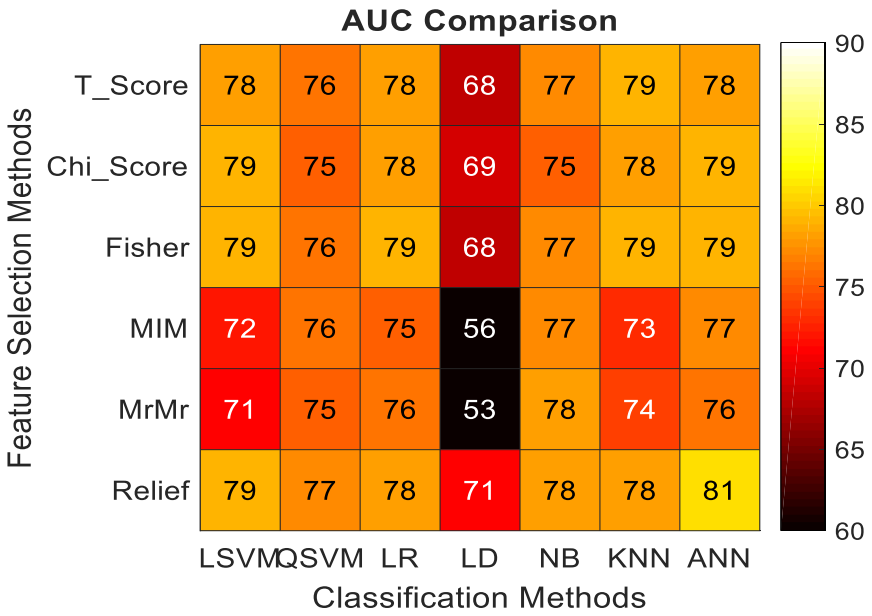


Figure 0.3: Heatmap representing the mode AUCs with size of selected features = 25; in rows for feature selection algorithms and, in columns for classification methods.

APPENDIX C

CASE 2: ANALYSIS WITH ONLY DEEP RADIOMICS FEATURES

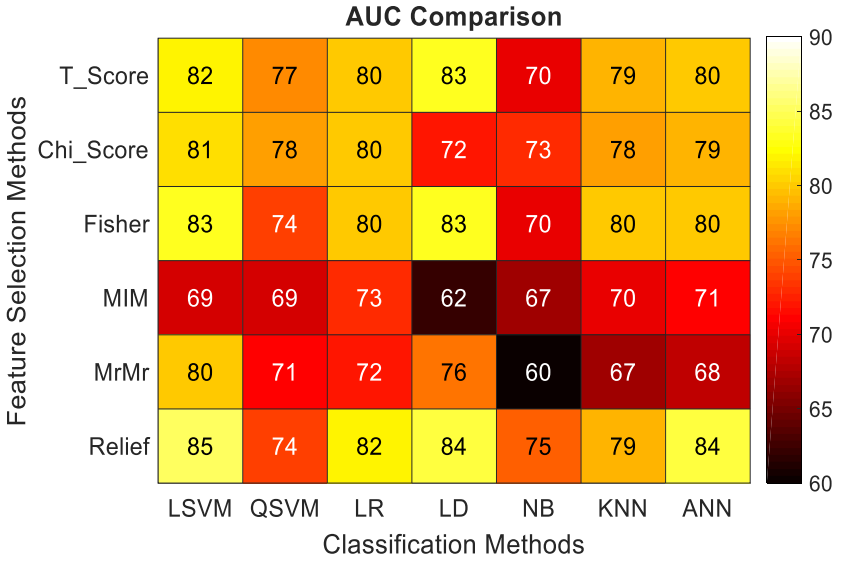


Figure 0.5: Heatmap representing the mode AUCs with size of selected features = 10; in rows for feature selection algorithms and, in columns for classification

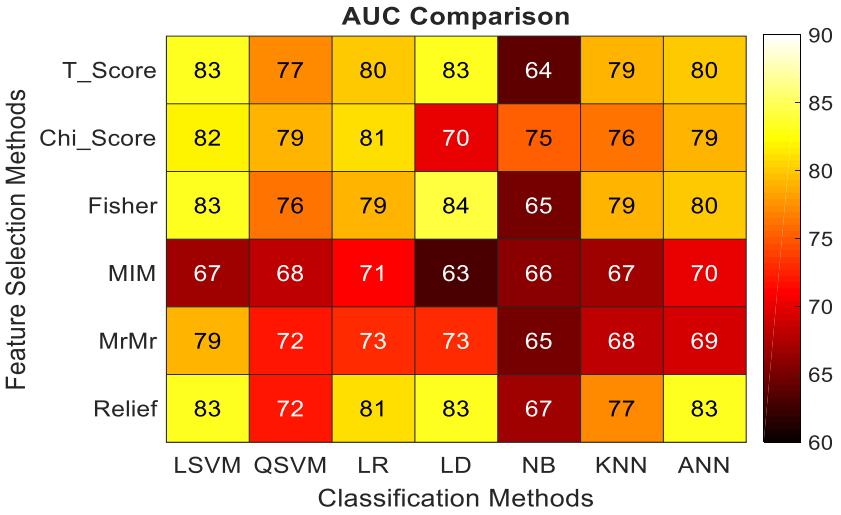


Figure 0.6: Heatmap representing the mode AUCs with size of selected features = 15; in rows for feature selection algorithms and, in columns for classification

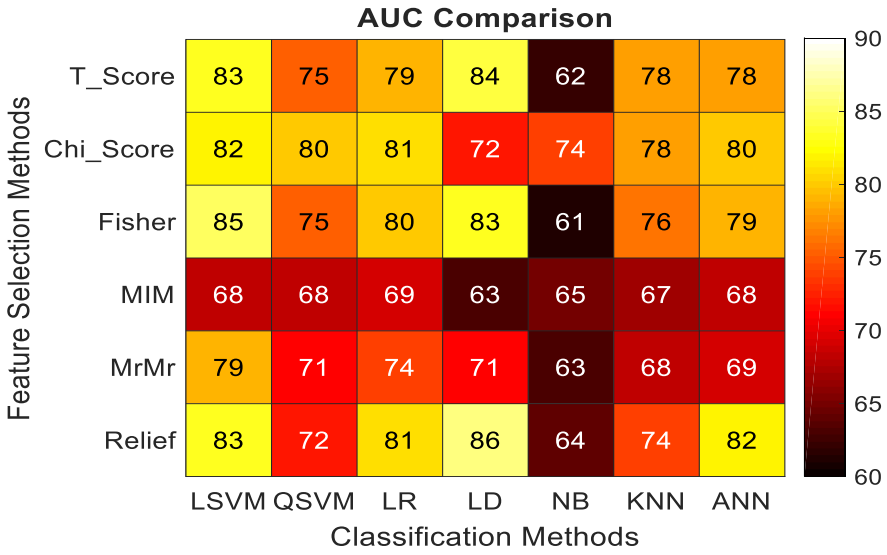


Figure 0.7: Heatmap representing the mode AUCs with size of selected features = 20; in rows for feature selection algorithms and, in columns for classification

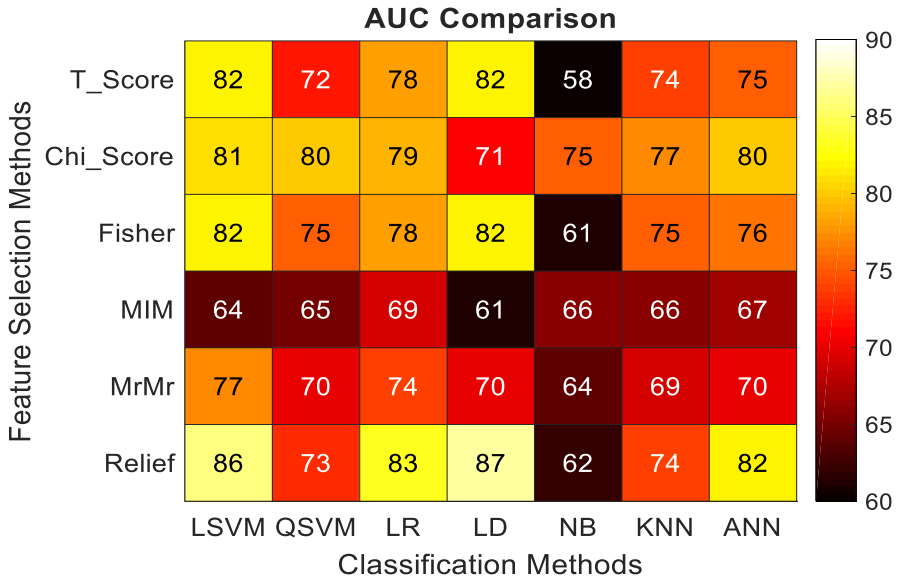


Figure 0.8: Heatmap representing the mode AUCs with size of selected features = 25; in rows for feature selection algorithms and, in columns for classification

APPENDIX D

CASE 3: ANALYSIS WITH COMBINATION OF BOTH HAND-CRAFTED RADIOMICS AND DEEP RADIOMICS FEATURES

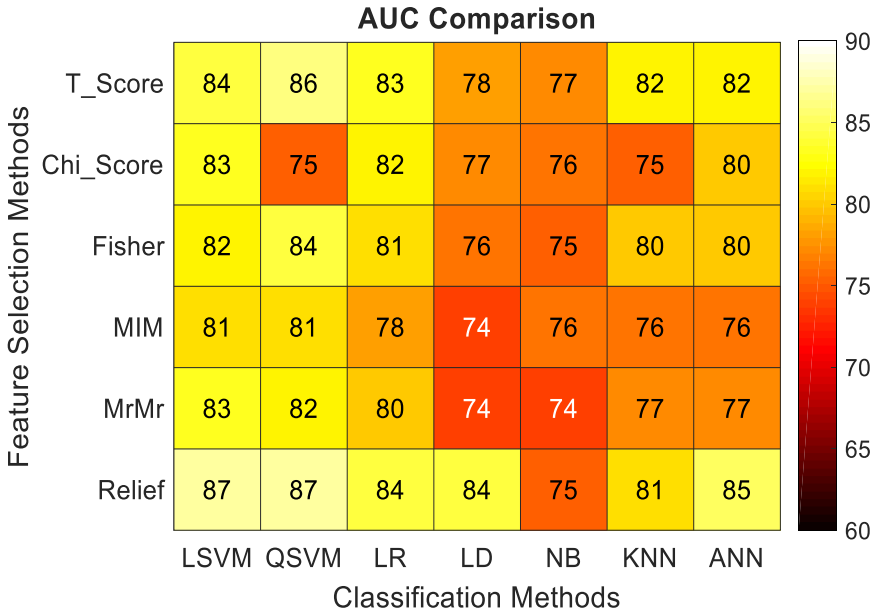


Figure 0.9: Heatmap representing the mode AUCs with size of selected features = 10; in rows for feature selection algorithms and, in columns for classification

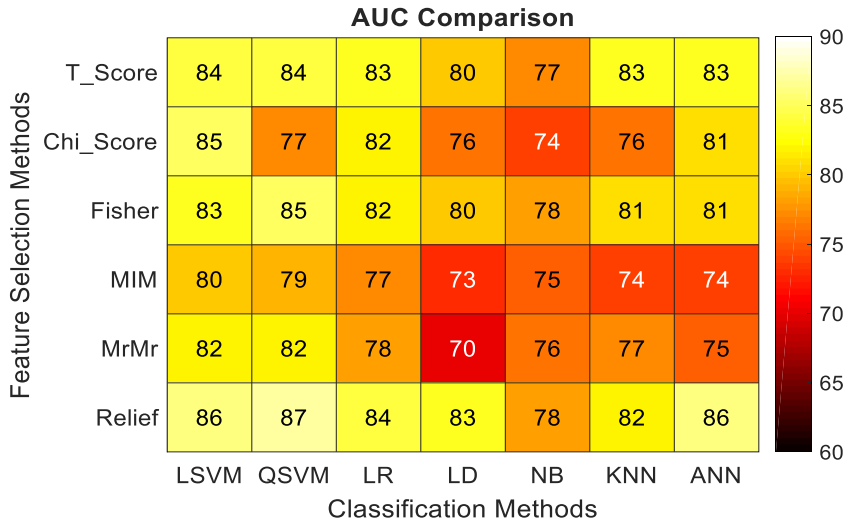


Figure 0.10: Heatmap representing the mode AUCs with size of selected features = 15; in rows for feature selection algorithms and, in columns for classification

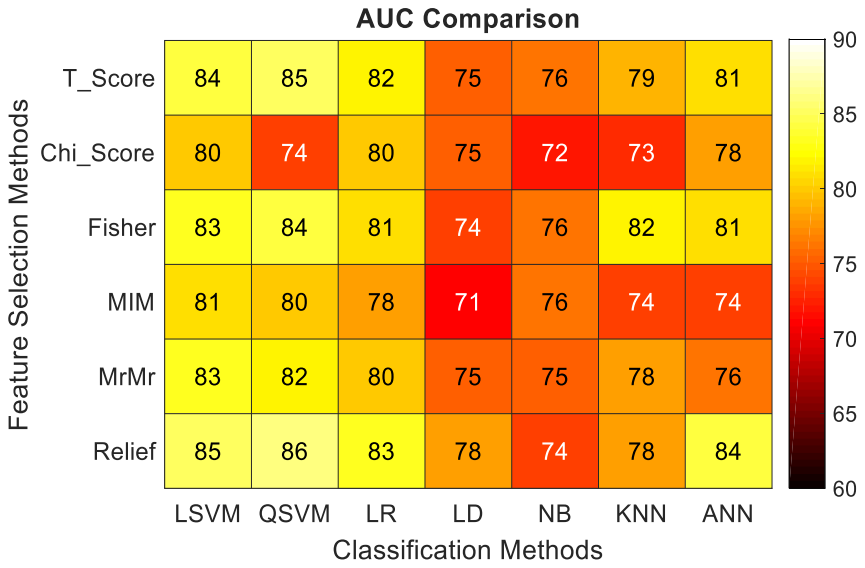


Figure 0.11: Heatmap representing the mode AUCs with size of selected features = 20; in rows for feature selection algorithms and, in columns for classification

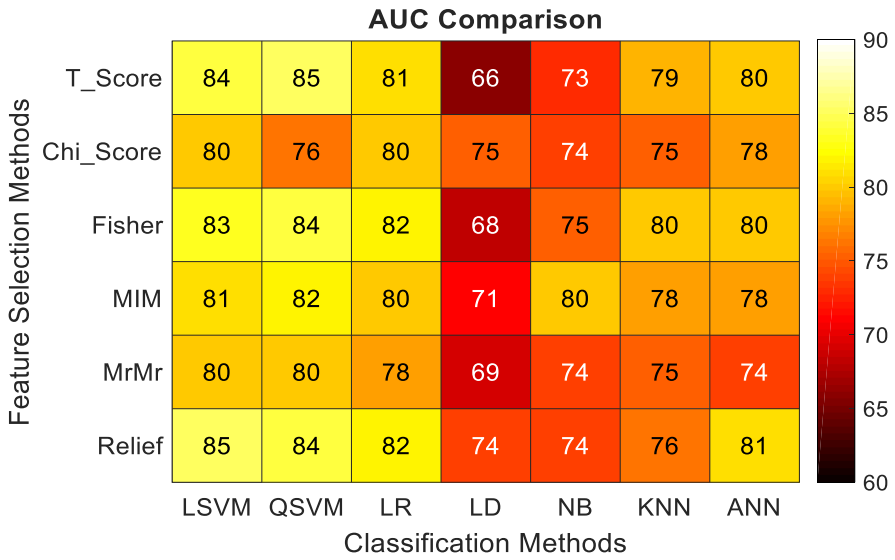


Figure 0.12: Heatmap representing the mode AUCs with size of selected features = 25; in rows for feature selection algorithms and, in columns for classification