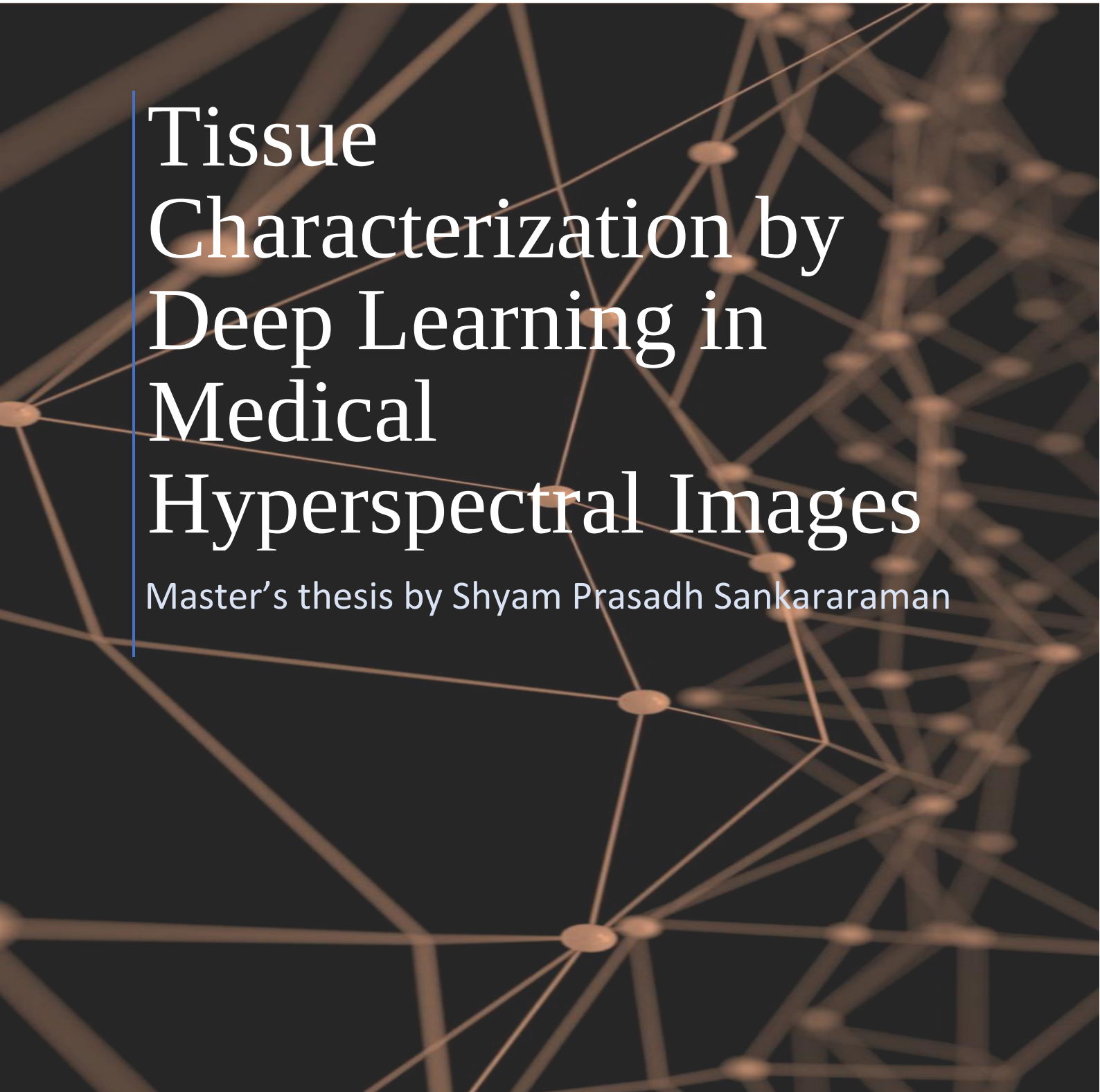
TU Delft



Tissue Characterization by Deep Learning in Medical Hyperspectral Images

Master's thesis by Shyam Prasadh Sankararaman

Tissue Characterization by Deep Learning in Medical Hyperspectral Images

by
S.P Sankararaman (4483766)

In partial fulfilment of the requirements for the degree of

Master of Science
in Mechanical – Control Engineering track

at the Delft University of Technology
to be defended on Wednesday, March 27th, 2019
in the presence of the graduation committee comprising

Prof. Dr. Gleb Vdovin (Committee chairperson, Professor at DCSC)
Dr. Caifeng Shan (Daily supervisor, Senior Scientist at Philips Research)
Dr. Ing. Raf Van de Plas (DCSC supervisor, Assistant Professor at DCSC)
Dr. Wei Pan (Committee member, Assistant Professor at Dept. of CoR)

TU Delft Delft University of Technology

PHILIPS

Abstract

Hyperspectral imaging (HSI) is a promising imaging modality in medical applications, especially for non-invasive and non-contact disease diagnosis and image-guided surgery. Encoding both spatial and spectral information, it can detect subtle changes in the biochemical and morphological properties of a tissue, revealing the early progression of a pathological condition like cancer. Previous medical hyperspectral image analysis approaches depended on handcrafted features or feature extraction principle, requiring considerable domain expertise. To overcome this, automatic feature learning approaches like convolutional neural networks (CNN), previously used in tasks like classification, detection and segmentation in medical images were applied to hyperspectral data, although in a limited number of research studies. This thesis was proposed to review the state-of-the-art in medical hyperspectral image analysis, identify the limitations in current methods, and present a proof-of-concept for using limited hyperspectral image data in CNN-driven tissue characterization.

The goal of this thesis is to characterize, using CNNs, *ex vivo* head and neck (tongue) tissue of patients affected by tumors. While previous work in this field implemented patch-based classification of tissue, in this thesis, a pixel-wise classification approach was proposed to obtain a smooth and continuous segmentation of hyperspectral images. To this end, two types of CNN models were trained from scratch using limited labelled training data, one to automatically learn the spectral features present in the hyperspectral data and the other to learn the combined spectral-spatial features from the hyperspectral data. Performance of four different trained models was evaluated by using a leave-one-out testing scheme, with the spectral-spatial learning approach with larger input spatial dimensions outperforming the other considered approaches.

Preface

It all began with a leap of faith: moving abroad, far away from the confines of my home, to fulfill my pledge to be an innovator, an engineer and a researcher *par excellence*. To be able to do that at TU Delft and later at Philips Research, was a privilege bestowed upon me. In my Bachelors, I found interest in electronics, control and robotics, which led me to my MSc programme at TU Delft, lending fodder for my interest in linear algebra and applied math. I was immensely fortunate to be part of a programme at DCSC that fostered scientific thinking and freedom in choice of study. That was the reason why in 2016, I enrolled for the Neural Networks course, back when deep learning was still in its developmental stage. The sparks of interest that it provoked in me, led me to spend sleepless nights working with learning models that could barely run in my near-obsolete CPU. Fast forward to now, my Master's graduation, having written my thesis at Philips Research on deep learning with hyperspectral images.

I am indebted to many people in my path to a successful Master's thesis. Firstly, my thanks to Caifeng, for trusting me with the research opportunity at Philips, for pushing me to do better during challenging times and for being readily available for feedback at all times. Next, I would like to thank Raf, for constantly supporting me from TU Delft, through constructive criticism and general advice during hardships, and for ensuring that my thesis progressed seamlessly. I would also like to thank Hong and Binyam, who were generous to provide me valuable feedback throughout my thesis, despite their busy schedules. I also cannot forget the friendships and good times with the PhDs and interns at Philips In-body Systems.

At a personal level, I hold utmost gratitude to my family in India for supporting me with my decision to study abroad and providing positivity during the travails of my Master's study. To Meghna, for bringing me the most joy during these years and always being there for me since 2015. Thanks to my friends in India for always staying in touch and to my friends in Netherlands for shaping my experience here. I am now excited for the next part of my life and the challenges that it brings along!

I hope that my efforts and any future efforts in medical imaging, lead to successful imaging software products that instill confidence in the minds of physicians and decision makers, relief to aggrieved patients through image-guided surgery and, early detection in at-risk groups. I dedicate all efforts from my part to my *paatti*, who could have well benefitted from the current and possible future advancements. This is to her, and others like her.

Shyam Prasadh, 2019

Contents

Table of figures xi
List of tables xiv
Introduction 1
Research question 1
Thesis outline 2
Chapter I – Hyperspectral Imaging 3
1.1 Hyperspectral versus RGB 3
1.2 Overview of a hyperspectral imaging system 4
1.3 Applications of hyperspectral imaging 5
1.3.1 Hyperspectral imaging in remote sensing 6
1.3.2 Hyperspectral imaging in medical domain 9
1.4 Advantages of hyperspectral imaging 17
1.4.1 Spatial information 17
1.4.2 Rich spectral information 17
1.4.3 Non-contact and non-invasive 18
1.5 Limitations of hyperspectral imaging 19
1.5.1 Signal-to-noise ratio 19
1.5.2 Lack of depth 19
1.5.3 Non-uniqueness 19
1.6 Why deep learning for hyperspectral imaging? 20
1.6.1 Automatic feature learning 20
1.6.2 Generalization ability 20
1.6.3 High dimensional data 20
Chapter II – Data Preprocessing 23
2.1 Image cropping 24
2.2 Image resizing and rotating 25
2.3 Cube stabilization 25
2.4 Label Preparation 27
2.4.1 RGB – hyperspectral image registration 27
2.4.2 Automatic vs manual 28
2.4.3 Label generation 29
2.5 Dataset preparation 31

2.6 Spectral signature analysis 32
2.7 Discussion 37
Chapter III – Deep Learning Setup 39
3.1 CNN theory 39
3.2 Backpropagation 41
3.3 Network Layers 42
3.3.1 Convolution 42
3.3.2 Activations 44
3.3.3 Pooling 45
3.3.4 Additional Configurations and Layers 45
3.3.5 Optimizers 47
3.3.6 Performance Metrics 49
3.3.7 Loss Functions 50
3.4 Relevant Architectures 52
3.5 Frameworks & processing capability 55
3.6 Proposed approaches 55
3.6.1 Patch classification vs pixel-wise classification 55
3.6.1 Spectral approach 56
3.6.1.1 Training set up 57
3.6.2 Spectral-Spatial Approach 59
3.6.3 Data augmentation in spectral-spatial method with 224 x 224 spatial dimension 62
3.6.4 Spectral-spatial method with spatial dimension 112 x 112 (data augmented) 64
Chapter IV – Experimental Results 67
4.1 Comparison of results 67
4.1.1 Sample #1 69
4.1.2 Sample #2 71
4.1.3 Sample #3 73
4.1.4 Sample #4 75
4.1.5 Sample #6 77
4.1.6 Sample #7 79
4.2 Discussion 81
4.3 Other relevant methods 83
4.3.1 Noisy Spectrum 83
4.3.2 Signal Smoothening 83
4.3.3 Principal Component Analysis 84
4.3.4 Non-Negative Matrix Factorization 85

4.3.5 Curvature Dependence 86
4.4 Remaining challenges and future perspective 87
Conclusion 89
Appendix I 91
Spectral reconstruction using PCA 91
Abbreviations 93
References 95

Table of figures

Figure 1: Illustration of a hyperspectral image with two spatial dimensions along x,y axes and a spectral dimension along z axis. (Left) a typical hyperspectral image composed of an image at each wavelength; (Right) The spectral signature at each pixel [5]. 4

Figure 2: Schematic diagram of a typical hyperspectral imaging system as discussed in [3]. Here a line scanning (pushbroom) image acquisition method is illustrated. 5

Figure 3: Illustration of hyperspectral image classification based on spectral-spatial features and DBN [13]. Two parallel channels are used for separately learning the spectral and spatial features respectively. A DBN followed by an LR classifier provides the output. 8

Figure 4: Different material - light interaction phenomena happening within the material. In HSI technique, diffuse reflection is mainly considered (slides of J. Workman)..... 10

Figure 5: Different applications of HSI in the medical domain. Primary categories include cancer detection and surgical guidance, which are both relevant to this thesis. 11

Figure 6: Difference between RGB vs HSI image in detecting features under the surgical bed. Left: image of the surgical bed with the residual tumor. Right: Pseudocolor visualization of the characteristics of the tissue including hematoma under the surface [31]. 15

Figure 7: Illustration of an application involving segmentation of abdominal cavity using HSI technique. Left: RGB image of the intestine. Right: segmentation based on spectral signatures [33]. 16

Figure 8: A representative result of spectral binary classification in head and neck tissue. The RGB image, output probability of cancer and its corresponding visualization are illustrated (from [35]). 17

Figure 9: Illustration comparing the structure of a hyperspectral and an RGB image as in [3]. The primary difference is in the number of channels or bands across which information is captured where a contiguous spectrum represents each point in the image, compared to discrete values in RGB. 18

Figure 10: Illustration of an HSI camera setup described in [38] with the corresponding components. 18

Figure 11: The complete workflow of the thesis. It starts from preparing training data from the patient records till obtaining the final segmentation images from the trained network..... 24

Figure 12: Translation (in pixels) between initial and final bands, across all patient records. It can be observed that this effect is prominent in certain samples (like #2) than in others before stabilization..... 26

Figure 13: Illustration of the two cube stabilization methods utilized. Left: Median band is chosen as the reference (fixed) image and the adjacent bands are moving images. These transformed bands become the reference for the subsequent bands up and down the range. Right: Median band is chosen as the reference image and all other bands in the spectral range are considered as moving images. 27

Figure 14: Collection of image point pairs by the Control Point method in MATLAB. Left: RGB image of corresponding to #3. Right: Representative hyperspectral image of #3. It is to be noted that the points are focused around the tumor affected regions for remove local image distortions..... 28

Figure 15: Images of a tongue affected by tumor. From left to right: Representative hyperspectral image; Grayscale of RGB image; After Automatic Intensity-based image registration; After Control Point image registration. 29

Figure 16: Illustration showing the process of ground truth labels creation. From the annotated pathology, RGB image and hyperspectral images from original patient records, the ground truth labels are created by registering the pathology and the hyperspectral images. 31

Figure 17: Spectral signatures of the #1 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue. 33

Figure 18: Spectral signatures of the #2 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue. 33

Figure 19: Spectral signatures of the #3 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue. 34

Figure 20: Spectral signatures of the #4 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue. 35
Figure 21: Spectral signatures of the #8 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue. 35
Figure 22: Spectral signatures of the #5 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue. 36
Figure 23: Spectral signatures of the #6 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue. 36
Figure 24: Spectral signatures of the #7 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue. 37
Figure 25: Flow of information from the retina to the visual cortex to the inferior temporal gyrus [39]. The regions V1, V2 and V4 detect edges, color, geometric shapes etc. from a scene. 39
Figure 26: Illustration of intermediate layers in a neural network obtained from [42]. Left: represents the output layer activations and the cost function. Right: represents the weights between neurons in intermediate layers. 41
Figure 27: Demonstration of the convolution operation between two matrices. A 5 x 5 matrix is convolved with a 3 x 3 kernel to produce a 3 x 3 output matrix..... 42
Figure 28: Visualization of the different convolution filters/feature maps [43] that show activations at layers 1 and 2 of a fully trained CNN, with its corresponding original image patches. 43
Figure 29: Plots showing the sigmoid function and the tanh function. The sigmoid function operates entirely in the positive range whereas tanh function operates between -1 and 1..... 44
Figure 30: Plots showing the ELU and ReLU activation functions. ReLU does not permit negative activation values whereas ELU allows smaller negative activations. 45
Figure 31: Comparison between max pooling and average pooling operations. Max pooling emphasizes on maximal value features, whereas average pooling de-emphasizes the maximal value features. 45
Figure 32: Equations that describe the BatchNorm layer as proposed in [47]..... 47
Figure 33: U-Net architecture proposed in [50]. It consists of a contractive path on the left following by an expansive path in the right, with feature concatenation occurring between layers of corresponding feature dimensions on both sides. 52
Figure 34: The spectral feature learning channel of the spectral-spatial residual network architecture [10]. 53
Figure 35:The spatial feature learning channel of the spectral-spatial residual network architecture [10]. 54
Figure 36: The simultaneous spectral-spatial learning network based on 3-D convolution layers [16]. It utilizes 3-D convolutional kernel to simultaneously learn spatial and spectral features..... 54
Figure 37: The proposed architecture for the spectral feature learning approach. It utilizes 1-D kernels of the form 1 x 1 x N, followed by two residual spectral layers for deep feature learning. 56
Figure 38: Schematic showing the constituent layers of the spectral residual block in the proposed architecture. It follows the order conv-ELU-BNorm, with a final BNorm and Dropout layer after identity summation. 57
Figure 39: Plots showing the model accuracy and loss values for 100 epochs during model training and validation for the first experiment. 58
Figure 40: The proposed architecture for the simultaneous spectral - spatial feature learning approach. Hyperspectral images of dimension 224 x 224 x 164 are provided as the input. 60
Figure 41: Plots showing the model accuracy and loss value for 50 epochs during model training and validation for the second experiment. Early stopping is applied to prevent overfitting in this case. 61
Figure 42: Plots showing the model accuracy and loss value for 50 epochs during model training and validation for the third experiment. Early stopping condition is applied to prevent overfitting in this case. 63
Figure 43: The proposed architecture for the simultaneous spectral - spatial feature learning approach. Hyperspectral images of dimension 112 x 112 x 164 are provided as the input. 64
Figure 44: Plots showing the model accuracy and loss value for 50 epochs during model training and validation for the fourth experiment. Early stopping is applied to prevent overfitting in this case. 65
Figure 45: Left column: Top to bottom, label for #1, predictions for the four methods (spectral method, spectral spatial method, spectral spatial with data augmentation, and spectral spatial with 112 x 112 x 164). Right column: confusion matrices for the four considered methods. 68

FIGURE 46: LEFT COLUMN: TOP TO BOTTOM, LABEL FOR #2, PREDICTIONS FOR THE THREE METHODS (SPECTRAL METHOD, SPECTRAL-SPATIAL METHOD WITH DATA AUGMENTATION, AND SPECTRAL-SPATIAL METHOD WITH INPUT DIMENSION 112 X 112 X 164). RIGHT COLUMN: CONFUSION MATRICES FOR THE THREE CONSIDERED APPROACHES. 70
FIGURE 47: THE ORIGINAL ROI SPATIAL REGION OF THE LABEL (LEFT) AND ITS SMALLER CROPPED AREA (RIGHT) EMPHASIZING THE TUMOR REGION. 71
FIGURE 48: LEFT COLUMN: TOP TO BOTTOM, LABEL FOR #3, PREDICTIONS FOR THE FOUR METHODS (SPECTRAL METHOD, SPECTRAL-SPATIAL METHOD, SPECTRAL-SPATIAL WITH DATA AUGMENTATION, AND SPECTRAL-SPATIAL WITH 112 X 112 X 164). RIGHT COLUMN: CONFUSION MATRICES FOR THE FOUR CONSIDERED METHODS. 72
FIGURE 49: REPRESENTATIVE HYPERSPECTRAL IMAGE WHICH SHOWS TISSUE CURVATURE ON THE CENTRAL TISSUE REGION. 73
FIGURE 50: LEFT COLUMN: TOP TO BOTTOM, LABEL FOR #4, PREDICTIONS FOR THE FOUR METHODS (SPECTRAL METHOD, SPECTRAL-SPATIAL METHOD, SPECTRAL-SPATIAL WITH DATA AUGMENTATION, AND SPECTRAL-SPATIAL METHOD WITH 112 X 112 X 164). RIGHT COLUMN: CONFUSION MATRICES FOR THE FOUR CONSIDERED METHODS. 74
FIGURE 51: REPRESENTATIVE HYPERSPECTRAL IMAGE WHICH SHOWS TISSUE CURVATURE ON THE PERIPHERY. 75
FIGURE 52: LEFT COLUMN: TOP TO BOTTOM, LABEL FOR #6, PREDICTIONS FOR THE FOUR METHODS (SPECTRAL METHOD, SPECTRAL-SPATIAL METHOD, SPECTRAL-SPATIAL WITH DATA AUGMENTATION, AND SPECTRAL-SPATIAL WITH 112 X 112 X 164). RIGHT COLUMN: CONFUSION MATRICES FOR THE FOUR CONSIDERED APPROACHES. 76
FIGURE 53: REPRESENTATIVE HYPERSPECTRAL IMAGE WHICH SHOWS TISSUE CURVATURE ON THE PERIPHERY. 77
FIGURE 54: LEFT COLUMN: TOP TO BOTTOM, LABEL FOR #7, PREDICTIONS FOR THE FOUR METHODS (SPECTRAL METHOD, SPECTRAL-SPATIAL METHOD, SPECTRAL-SPATIAL WITH DATA AUGMENTATION, AND SPECTRAL-SPATIAL METHOD WITH 112 X 112 X 164). RIGHT COLUMN: CONFUSION MATRICES FOR THE FOUR CONSIDERED APPROACHES. 78
FIGURE 55: THE ORIGINAL ROI SPATIAL REGION OF THE LABEL (LEFT) AND ITS SMALLER CROPPED AREA (RIGHT) EMPHASIZING THE TUMOR REGION. 79
FIGURE 56: COMPARISON OF PRECISION, RECALL AND F-1 SCORE METRICS CORRESPONDING TO THE TESTING SAMPLES ACROSS FOUR EXPERIMENTS. BY OBSERVING THE F-1 SCORE ACROSS ALL THE PATIENT SAMPLES, CLEARLY THE SPECTRAL-SPATIAL METHOD OUTPERFORMS THE SPECTRAL METHOD. (_S, _SS, _SSA AND _112 DENOTE THE EXPERIMENTS WITH SPECTRAL, SPECTRAL-SPATIAL, SPECTRAL-SPATIAL AUGMENTED AND SPECTRAL-SPATIAL 112X112 RESPECTIVELY.) 80
FIGURE 57: BAR GRAPH SHOWING THE MEAN AND STANDARD DEVIATION VALUES OF PRECISION, RECALL AND F-1 METRICS ACROSS ALL TESTING SAMPLES, FOR FOUR EXPERIMENTS 81
FIGURE 58: PLOTS SHOWING THE EFFECT OF FILTERING ON NOISY SPECTRAL SIGNALS. COUNTER CLOCKWISE FROM TOP: GRAPH OF RAW SPECTRAL SIGNATURES OF TUMOR AND MUSCLE CLASS; SMOOTHENING SPECTRA USING SAVITZKY-GOLAY FILTER; SMOOTHENING SPECTRA USING GAUSSIAN FILTER. 84
FIGURE 59: PLOTS SHOWING THE EFFECT OF APPROXIMATIONS ON RAW NOISY SPECTRAL SIGNALS. CLOCKWISE FROM TOP: NOISY RAW SPECTRA; SPECTRA AFTER NMF APPLICATION; SPECTRA AFTER PCA APPLICATION. 86
FIGURE 60CHANNELS 87
FIGURE 61: PLOTS SHOWING THE EFFECT OF INTEGRAL NORMALIZATION. LEFT: RAW SPECTRAL SIGNATURE OF TUMOR TISSUE; RIGHT: EFFECT OF DIVING INDIVIDUAL SPECTRUM BY ITS INTEGRAL OR AREA UNDER THE CURVE. 87

List of tables

TABLE 1: LIST OF REFERENCES OF HSI RESEARCH IN THE REMOTE SENSING DOMAIN (TRADITIONAL AND DEEP LEARNING BASED). 5
TABLE 2: LIST OF REFERENCES OF HSI RESEARCH IN THE MEDICAL DOMAIN (TRADITIONAL AND DEEP LEARNING BASED)..... 11
TABLE 3: THE METHOD UTILIZED FOR MAPPING THE RGB IMAGE TO THE HYPERSPECTRAL IMAGE COORDINATES GLOBALLY OR LOCALLY FOR DIFFERENT PATIENT SAMPLES..... 29
TABLE 4: THE RATIO OF NUMBER OF PIXELS BELONGING TO DIFFERENT CLASSES BASED ON THE LABELS BEFORE CLASS BALANCING, IN THE FORM BACKGROUND : TUMOR : MUSCLE : UNKNOWN..... 31
TABLE 5: THE RATIO OF NUMBER OF PIXELS BELONGING TO DIFFERENT CLASSES BASED ON THE LABELS AFTER CLASS BALANCING, IN THE FORM BACKGROUND : TUMOR : MUSCLE : UNKNOWN..... 32
TABLE 6: NUMBER OF SUB-CROPPED REGIONS OBTAINED FROM ROI HYPERSPECTRAL IMAGE OF EACH PATIENT SAMPLE..... 32
TABLE 7: DIFFERENT HYPERPARAMETERS DETERMINED DURING TRAINING OF FIRST EXPERIMENT. 58
TABLE 8: THE ELAPSED TIME DURING MODEL TRAINING AND TESTING PROCESS IN THE FIRST EXPERIMENT. 59
TABLE 9: DIFFERENT HYPERPARAMETERS DETERMINED DURING TRAINING OF SECOND EXPERIMENT..... 62
TABLE 10: THE ELAPSED TIME DURING MODEL TRAINING AND TESTING PROCESS IN THE SECOND EXPERIMENT 62
TABLE 11: THE ELAPSED TIME DURING MODEL TRAINING AND TESTING PROCESS IN THE THIRD EXPERIMENT 63
TABLE 12: DIFFERENT HYPERPARAMETERS DETERMINED DURING TRAINING OF THE FOURTH EXPERIMENT..... 65
TABLE 13: THE ELAPSED TIME DURING MODEL TRAINING AND TESTING PROCESS IN THE FOURTH EXPERIMENT 65
TABLE 14: DETERMINATION OF RECEPTIVE FIELD FOR EACH LAYER OF THE PROPOSED NETWORK 82

Introduction

In minimally invasive surgeries for tumor removal, it is important to diagnose the extent of the tumor and identify the tumor affected regions accurately. For this purpose, a non-contact, non-invasive imaging method called Hyperspectral Imaging (HSI) has emerged in the last decade. By utilizing the characteristics of light-tissue interaction, the change in tissue condition can be identified. Manual segmentations of these medical images require domain expertise and can be time intensive, which necessitates automatic image segmentation methods to ease the workload of the clinicians and to possibly supplement their diagnoses. This clinical problem has spurred researchers on to apply deep learning models to automatically analyze the medical images, since the advent of models like convolutional neural networks in the last four years.

At the In-body systems department of Philips Research, ongoing research on the HSI modality prompted a question "How can the state-of-the-art methods in deep learning be applied to hyperspectral images to develop a non-invasive, automatic segmentation tool that can be utilized during surgical procedures?". This Master's thesis is defined in a way to answer this research problem, which when successful could serve as a proof-of-concept solution.

Tackling this problem requires a two-fold approach: first, a study into how the acquired raw hyperspectral patient data can be made available for training a deep learning model. The second involves investigating different architectures that are currently applied in medical imaging research and designing custom artificial neural network architecture if required for our case. All the developed methods can then be compared for their segmentation performance, which will help recommend the best method for further investigation.

Research question

The aim of this Master's thesis can be expressed by formulating a research question and its associated sub-questions.

"Can a convolutional neural network perform tissue segmentation on limited patient data?"

1) **Learning features:** What are the possible approaches in learning features from a hyperspectral data cube? Are spatial features as informative compared to spectral features?

2) **Model design**: What design choices were made corresponding to the feature learning approaches?

3) **Model performance**: How do the performance metrics compare for the considered experiments?

4) **Model tuning:** How can the set of hyperparameters for a given experiment be determined?

5) **Augmenting data:** What are the effects of data augmentation on the network's performance? Can it overcome the problems due to limited patient data?

To answer the question, the acquired patient hyperspectral data is studied extensively, and relevant image processing methods are identified to create the training data and the ground truth (labels). Following this, the training data is prepared in such a way to perform pixel-wise classification or semantic segmentation. Further, different approaches to automatically learn features from the hyperspectral data are proposed, then their performances are evaluated. The content of this thesis flows to attempt answering the formulated research question and its sub-questions.

Thesis outline

Following this introduction section where the research problem is defined, Section I provides all the requisite background knowledge for the completion of this thesis, ranging from theory behind HSI to current applications in remote sensing and medical domains. Section II studies the available patient data, exploring possible processing techniques required before it can be used in training a deep neural network. In Section III, different approaches are proposed to configure the networks to perform the tissue segmentation task. Section IV discusses the results of these experiments in detail along with their performance specifications and observations for future research, before concluding the findings of this thesis.

Chapter I – Hyperspectral Imaging

Hyperspectral Imaging (HSI) is a spectral imaging technique, which integrates conventional imaging and spectroscopy to acquire spatial and also spectral data of an object. It involves capturing two-dimensional images across a wide range of the electromagnetic spectrum, making it possible to characterize materials by means of their reflectance or emittance spectra. This means, across a particular wavelength band, a contiguous spectrum of each image pixel is acquired and thus a three-dimensional cubical structure called a 3-D hypercube is obtained. The "contiguous'' aspect of the spectral bands is most significant because it can ensure that there are no gaps through which precious information could slip unnoticed. This is especially applicable in medical diagnosis, where any subtle spectral differences can be critical, which is not possible in a conventional RGB image, which has only three bands of red, blue and green colors of discrete wavelengths.

The initial attempts in HSI were for mineralogical mapping of land surface, followed by vegetation classification based on nitrogen content, ocean and coastal studies, and hazardous waste clean up around mining sites. However, due to the continuing advances in the semiconductor industry, the application space of HSI has broadened to environmental [1], food [2], medical [3], forensic and surveillance fields [4].

1.1 Hyperspectral versus RGB

In RGB digital imaging, each two-dimensional image I can be represented as an array of pixels, with x number of rows and y number of columns. If the red, green and blue color intensities of each image pixel are combined, then an RGB or truecolor image is obtained, which can be represented as I (x, y, 3). However, in spectral imaging there are multiple (>3) intensity components B per pixel, where λ represents each wavelength at which each intensity image is captured. Thus, a spectral image can be represented as I (x, y, B), where B is the number of spectral channels. One such example from [5] is shown in Figure 1. It can be seen that a hyperspectral image contains more information than an RGB image, by storing information along the spatial and spectral dimensions. It could also be said that each pixel in a hypercube possesses its own individual spectral signature that can be used to identify it with better precision, which can in turn be utilized in pixel classification methods for different applications like remote sensing, material analysis, and medical diagnosis.

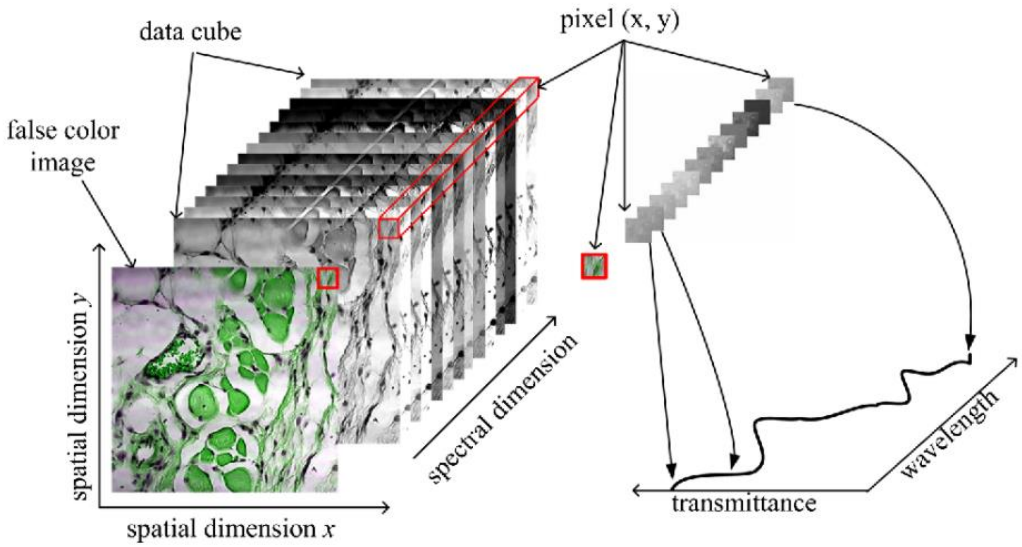


Figure 1: Illustration of a hyperspectral image with two spatial dimensions along x,y axes and a spectral dimension along z axis. (Left) a typical hyperspectral image composed of an image at each wavelength; (Right) The spectral signature at each pixel [5].

1.2 Overview of a hyperspectral imaging system

A typical HSI system comprises the following components according to the review on medical HSI [3]

1) Light source – illuminates the tissue after which it is projected to the front lens, which focuses the light into an entrance slit, permitting only a narrow line of light to pass. This controls the amount of light, which is further collimated onto the dispersion device.

2) Dispersion device – prism or grating that splits the collimated light into various wavelengths. This dispersed light is focused onto the detector arrays.

3) Detector arrays – optical detectors that can record the electromagnetic radiation

One such typical system with its components is shown schematically in Figure 2. A pushbroom HSI system is synonymous to a line-scanning HSI system.

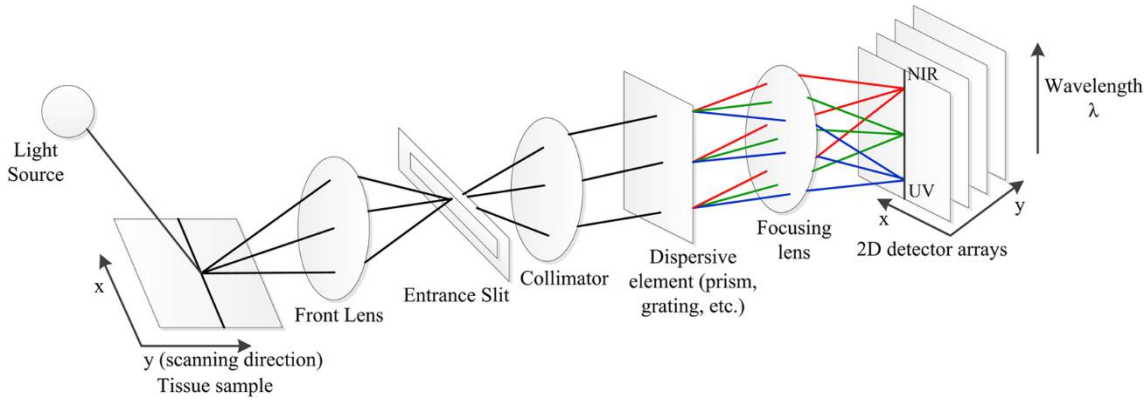


Figure 2: Schematic diagram of a typical hyperspectral imaging system as discussed in [3]. Here a line scanning (pushbroom) image acquisition method is illustrated.

1.3 Applications of hyperspectral imaging

While the HSI modality has found predominant application in the remote sensing domain, HSI is emerging as a potent tool in the medical field, specifically for non-invasive disease diagnosis and surgical guidance. In the further discussion, the different applications of HSI for land-cover classification in remote sensing and tissue characterization within medical domain are reviewed. For both the domains, the traditional classification methods used earlier shall be discussed, segueing into the most recent progress in hyperspectral image classification using deep learning. This could also serve a path to explore how the state-of-the-art algorithms in deep learning used in remote sensing could be adopted into medical imaging applications, where still traditional classifiers with feature engineering continue to be used, with limited progress in the deep learning front.

Table 1: List of references of HSI research in the remote sensing domain (traditional and deep learning based).

Reference	Method	Remarks
Traditional - remote sensing classification		
[6]	SVM	Multiple feature combining (spectral, texture and shape); manifold-learning-based dimension reduction
[7]	SVM, k-NN, CART, Naïve Bayes	Grouping of similar bands; Manifold ranking of grouped bands for group representatives
[8]	SVM	Marker map using pixels multiple classifiers assign to a particular class (majority voting); spectral-spatial information
Deep learning - remote sensing classification		

[11]	SAE, PCA/NMF, LR	Spectra-spatial features extracted separately and classified using SAE and LR
[12]	CNN	Pixel-pair joint classification using voting strategy; smaller dataset
[13]	DBN, LR	DBN-based feature extraction, followed by LR based fine-tuning; spectral, spatial (using PCA) and spectral-spatial features
[14]	1-D, 2-D, 3-D CNN	Extract spatial, spectral and spatial-spectral features using three methods; effect of L2 regularization
[15]	CNN, BLDE	Spectral dimension reduction using BLDE; CNN based spatial feature extraction; stacked features classified using SVM, LR
[16]	3-D CNN	Applying 3-D kernels on hyperspectral data to simultaneously learn spectral-spatial feature; no preprocessing
[10]	3-D CNN	Residual layers for deep feature learning; spectral-spatial classification
[18]	CNN, unpooling	Unsupervised learning of hyperspectral data representation; first residual layer features detect classes in the land cover

1.3.1 Hyperspectral imaging in remote sensing

1.3.1.1 Hyperspectral image classification by traditional methods

Studies on hyperspectral image analysis originate from the remote sensing domain, in which it is predominantly used. Certain tasks like land-cover mapping, object recognition, and anomaly detection can be performed by classifying each pixel in a hyperspectral image. Since a hyperspectral image consists of spatial information along with rich, contiguous spectral bands, the possibility of accurate classification is high. In case of traditional classification methods, more emphasis is placed on band selection and feature extraction, by reducing the high dimensionality of data and identifying the most discriminative bands. This is to counter Hughes phenomenon, which postulates loss of classification accuracy with high dimensional features in a small number of training samples [6]. This is also called as the curse of dimensionality. By combining multiple features (spectral, texture and shape) linearly and reducing the high dimensionality, a classifier like support vector machine (SVM) can be trained to learn the extracted features [7]. Feature selection methods avoid any lower dimensional projections and identify the most representative features from all the bands. This is performed in [8] by using band clustering and subsequent manifold ranking of the bands in each cluster.

1.3.1.2 Hyperspectral image classification by deep learning

Further efforts to classify hyperspectral pixels, incorporate spatial features which are correlated and provide complimentary information along with the spectral features. Depending on the levels at which the spectral and spatial information are fused, there can be three different approaches:

1) Feature level, where the spectral and spatial features are extracted independently and then concatenated. [7]

2) Decision level, where the spectral and spatial features are extracted independently and then integrated by using a majority voting strategy. [9]
3) Data level, where the spectral and spatial features are simultaneously extracted from the hyperspectral data.

In the traditional methods, a considerable effort is spent on feature engineering. It is also argued that such features do not generalize well to all scenarios and kernel-based classifiers simply do not possess the representation capacity to learn the integrated spatial-spectral features [10]. Due to these shortcomings of the feature engineering frameworks, the attention turned towards deep learning methods, which automatically learn representations that are relevant to the classification. Thus, the two-fold process of feature extraction and classifier training is simultaneously incorporated in one.

An early implementation used Stacked Auto Encoders (SAEs), which can extract deep hyperspectral features that can be classified by logistic regression [11]. It outperforms other feature extraction methods like PCA and NMF. Apart from learning spectral features, the spatial features from a PCA-reduced hyperspectral image, around a pixel's neighborhood were extracted and concatenated with its corresponding spectral features. This study was able to validate that joint spectral-spatial features helped the SAE-LR perform better than when using only spatial or spectral features of the image. A novel approach of using pixel-pair features was introduced in [12], where a pair of pixels in the labelled training data would be receive a label *L* when they are from the same class and, labelled 0 when they are different. Following this, a majority voting strategy for the label prediction is performed for the central pixel by using the neighboring pixels and their labels.

Similarly, another approach [13] made use of Deep Belief Network (DBN) and LR to classify the land cover, by using spectral, spatial and spectral-spatial features respectively. A DBN is constituted by stacking consecutive Restricted Boltzmann Machines (RBM), where the first layer of trained RBM input the learned representation or features to the next RBM layer. This chain of learning in the connected RBMs constitutes pretraining a DBN, which is then connected to an LR classifier to fine-tune the parameters by backpropagation. For spectral features, the 1-D data representation is learned by the DBN, whereas for spatial features, a PCA based feature extraction and flattening on a small spatial neighborhood can feed the data to the DBN. For spectral-spatial learning, two parallel channels with spectral and spatial learning are constructed, with feature stacking for the final classification (illustrated in Figure 3).

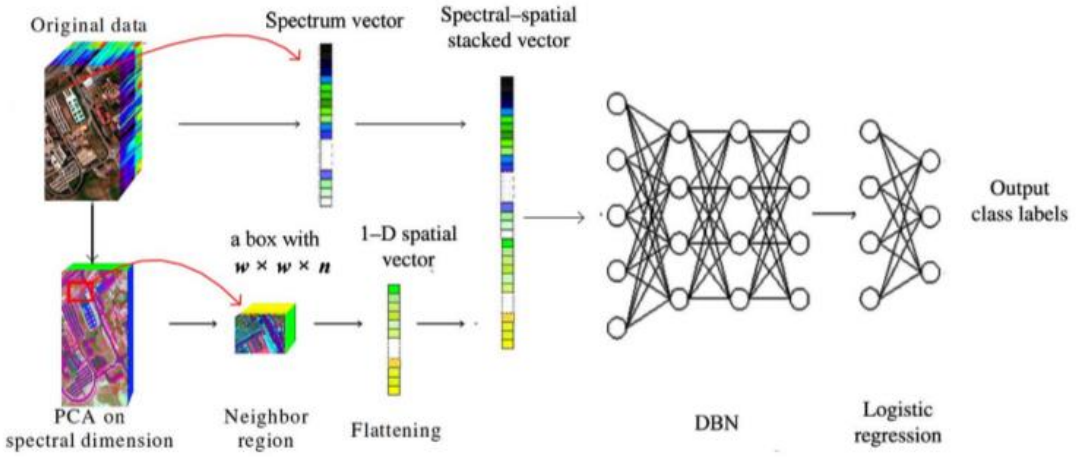


Figure 3: Illustration of hyperspectral image classification based on spectral-spatial features and DBN [13]. Two parallel channels are used for separately learning the spectral and spatial features respectively. A DBN followed by an LR classifier provides the output.

Similarly, three different strategies for the three features (spectral, spatial and spectral-spatial) were proposed in [14], which were: (1) 1-D convolution based spectral feature classification; (2) 2-D convolution based spatial feature classification; and (3) 3-D convolution-based classification of spectral-spatial features simultaneously. This study also investigated some strategies like L-2 regularization and virtual data augmentation to combat poor performance that can occur due to high dimensionality and small number of data cubes. In [15], spectral features were extracted using a balanced local discriminant embedding method (BLDE) and, combined with spectral features extracted from PCA and 2-D CNN network. It lacked the advantage of simultaneous spectral-spatial feature extraction: the correlation between these features, which was lost during PCA. Following this, another work [16] on joint spatial-spectral features using 3-D convolution, argued that using 3-D kernels or filters during convolution operation can extract spectral-spatial features simultaneously from hyperspectral images and improve classification performance. A closer look at this architecture can be found in the relevant architecture discussion in the later part of this report.

While the 3-D architectures can perform better than the previously discussed feature-fusion methods, the classification performance degrades with increase in depth of the network, thus making it harder to train such deep networks. However, deeper networks are needed to learn the discriminative spatial-spectral features in high dimensional data with a small training data set and generalize robustly on test data. In order to solve this conundrum, residual network blocks [17] were introduced in the network, along with batch normalization and reported significant increase in performance on both small and large data sets. This network [10] is discussed in detail later, as an architecture relevant to this thesis.

In one of the few efforts in unsupervised learning, a network with a conv-deconv structure with residual blocks was proposed [18]. The conv sub-network functions like an encoder, learning the abstract feature representation of the input hyperspectral image data, with max pooling to reduce the spatial feature size. In the deconv sub-network an unpooling operation was introduced to expand the spatial feature size by using the stored max pool indices. Though not intended for land cover detection, some of the learned features had activated/ suppressed pixels that denote particular classes in the land cover and could outperform other supervised learning methods (SVM, CNN etc.). This could open up potential applications classifying hyperspectral data with limited labels in an unsupervised manner.

1.3.2 Hyperspectral imaging in medical domain

1.3.2.1 Theory

Medical HSI (or MHSI) is increasingly used as an imaging modality for non-invasive medical diagnosis and surgical guidance. By understanding how medical HSI works in the context of tissue, we can fully appreciate the technique's potential for providing information about tissue constituents lying deep within the tissue. In the study of light–tissue interaction, the inhomogeneity of the tissue is an important aspect, making the optical properties vary spatially within the tissue. Multiple scattering and absorption are two important processes that occur when light interacts with matter. Scattering occurs when light crosses over media of different refractive indices while molecules absorb light, with the energy of the incident photon corresponding to the gap between the internal energy states of the molecule. Likewise, in tissue there are constituents, which scatter incident light, while some absorb light. It is observed that subcellular organelles, like mitochondria are the predominant scatterers of light [3]. In the therapeutic window from 600 to 1300nm, most tissues are weak absorbers of light, and light propagation becomes predominantly scattered and diffuse. However, at VIS wavelengths, most light is absorbed by blood and melanin and are called chromophores. From a medical standpoint, this can represent the concentration of haemoglobin and thus the oxygen concentration, and it could point for example, to signs of cancer like angiogenesis and hypermetabolism [19].

Apart from the reflection and absorption processes that occur in the tissue, there are tissue components like collagen and elastin, which are two important proteins in the connective tissue, or NAPDH and Flavin, which exhibit fluorescence and are called fluorophores. Fluorescence occurs when the absorbed light (usually UV to VIS) is re-emitted at a higher wavelength ranging from VIS to NIR region.

After multiple scattering and absorption within the tissue, light propagates back to the tissue surface, along with specular reflections, and it leads to highly randomized light directions. This phenomenon is called diffuse reflectance and is the basis for how hyperspectral image data is

acquired. Since this randomized light has propagated different sampling depths across a volume of tissue, the optical properties of this light could represent an average tissue property over this volume [3]. The implication of this morphological-optical connection is that, when the morphology of the tissue changes, there should correspondingly be a change in the measured reflected light and any changes in haemoglobin absorption (pointing to angiogenesis) should translate to change in the absorption signal. In combination, a diffuse reflected spectral signal can indicate the progression of disease. Similarly, by using fluorescence imaging, the alterations in biochemical composition of a tissue could be studied, thus paving way for a multimodal reflectance – fluorescence imaging, which can help diagnose cancer [20].

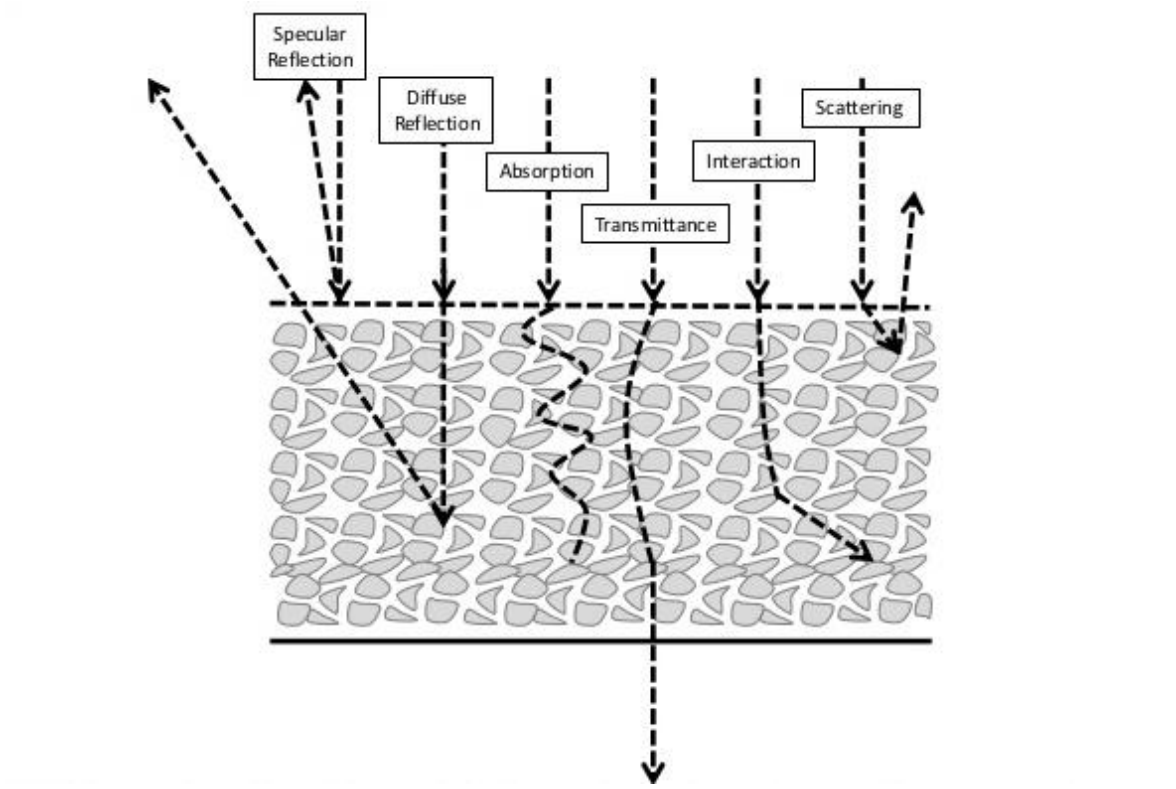


Figure 4: Different material - light interaction phenomena happening within the material. In HSI technique, diffuse reflection is mainly considered (slides of J. Workman).

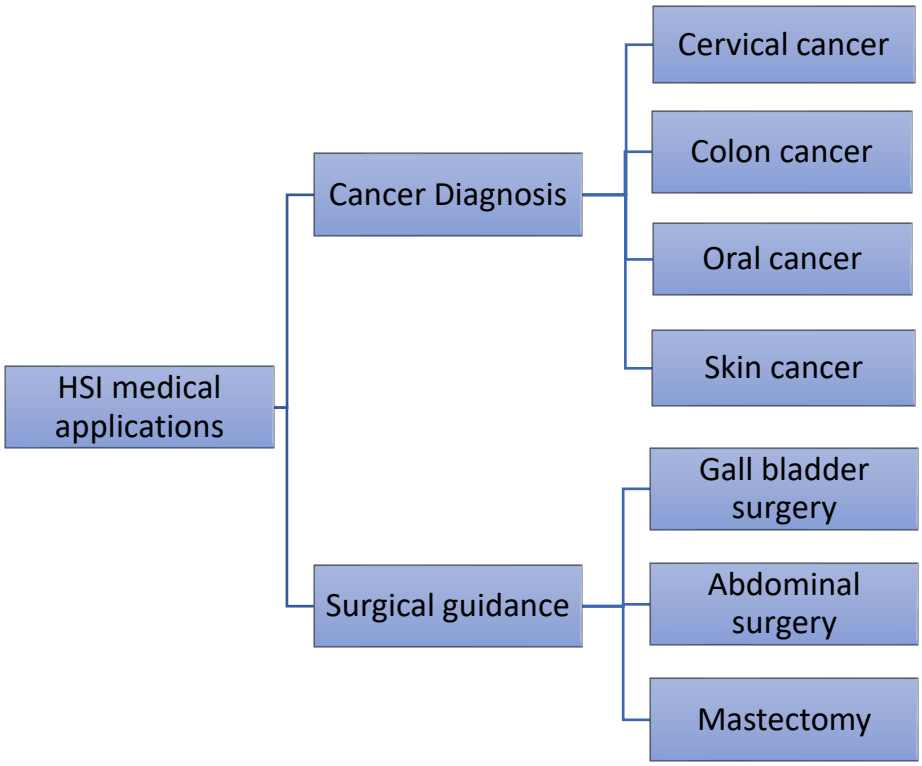


Figure 5: Different applications of HSI in the medical domain. Primary categories include cancer detection and surgical guidance, which are both relevant to this thesis.

Table 2: List of references of HSI research in the medical domain (traditional and deep learning based).

Reference	System/Method	Remarks
Traditional - Cancer diagnosis		
[21]	Multimodal	Differentiate various grades of cervical neoplasia; comparison with Pap smear.
[23]	VIS-NIR	HSI of colon biopsy slides; ICA, K-means clustering, and LDA for classification into normal and malignant.
[24]	HSI microscope	Classification of nuclei into normal, benign, or malignant.
[25]	VIS, NIR	Classification of colon tissue into fat, tumor, mucosa and healthy tissue using SVM; combination of VIS and NIR wavelength images
[26]	HSI – AOTF	Distinguish tumor and normal tissue in tongue based on sparse representation; comparison with SVM.
[27]	Snapshot	Multimodal (reflectance and autofluorescence) HSI for detection of oral cancer.
[28]	NIR, multispectral	NIR for thermal signatures, VIS for extent of tissue; assess blood volume, oxygenation to study effectiveness of treatment for Kaposi's sarcoma.
[29]	Handheld HSI – FPI	Early/ malignant melanoma tumor margin; feature selection, dimensionality reduction of spectra; Aisles procedure to reduce false positives.

[30]	Multi organ HIS	One-vs-one modeling between multiple organs with SVM and spectral classification using best model; spectral classification using MLP.
Traditional - Surgical Guidance		
[31]	HSI, multispectral	Identification of tumor in resected tissue; differentiate tumor, muscle and connective tissue; comparison with histopathology.
[32]	NIR, endoscopy	Noninvasive examination of biliary tissue; PCA, differentiate surrounding tissue; identify molecular composition of specific regions.
[33]	VIS-NIR, IR	SVM to classify normal and ischemic intestine.
Deep learning – MHSI		
[34]	CNN	Preprocessing of mice tumor hyperspectral data; spatial-averaged spectral binary classification.
[35]	CNN	Squamous cell and thyroid carcinoma detection using HSI spectra.
[36]	3-D CNN, CNN	Squamous cell and thyroid carcinoma detection, multi-class classification of normal thyroid tissue, multi-class classification of thyroid cancer.

1.3.2.2 MHSI applications using traditional methods

Cancer diagnosis

It has been theorized that any spectral changes in a tissue points to the progression of its pathological state [3]. Any morphological and biochemical changes in the tissue alter its reflectance, absorption, and fluorescence properties. It has been shown that, by observing the absorption spectrum of a tissue, it is possible to quantify the haemoglobin concentration and oxygen saturation, and detect angiogenesis [19]. With HSI it possible to not just observe the reflectance/absorption, but also capture multiple images of a particular tissue. Most research on HSI that studied cancer focuses on the following aspects:

1) Classify cancer grades by studying the morphological and structural properties of cancer affected histological specimens;

2) Identify *in vivo* precancerous and malignant lesions;

3) Quantify angiogenesis and rate of metabolism by measure haemoglobin concentration and oxygen saturation;

4) Recognize genomic alterations that characterize tumor progression in the tissue.

The first two aspects of research are aligned along the objectives of this master thesis, as discussed in the introduction section. Therefore, the literature review on HSI cancer research focuses particularly on these two aspects, and literature that discusses the problem of tumor tissue characterization is discussed below for the following types of cancer.

Cervical cancer is the leading cause of cancer death in women (in U.S) [22]. Traditionally, a Pap smear test is used for cervical cancer screening. However, it produces large false positive rates of 15 – 40%. Therefore, studies involving both reflectance and fluorescence have tried to detect pre-cancer in cervical tissue. A multimodal HSI system using the VIS to NIR range has been used to distinguish between affected and healthy tissue *in vivo*. It was able to distinguish high-grade lesions, low-grade lesions, and healthy tissue at a much greater rate than Pap smear [21].

Colon, or colorectal cancer, affects the colon, rectum or appendix, and it is the third most fatal type of cancer in both men and women [22]. Usually, the specimen is to be investigated pathologically under a microscope and the morphological changes in the cells and their distribution are observed. This process can be time-consuming and the observations inconsistent. Therefore, HSI has been used to distinguish cells, in biopsy tissues based on pathology slides, as normal and malignant, based on shape, size and other geometrical features of cellular components [23]. An extension of this experiment for classification of three grades of biopsy tissue (normal, benign and malignant) using HSI was performed in [24]. A recent study [25] examined the potential of HSI in laparoscopic surgical workflow, by using two cameras, one in the visible wavelength range (400 -1000 nm) and other in the NIR wavelength range (900 – 1700 nm). By utilizing the spectra in the hyperspectral image and an SVM classifier, the different tissue types like fat, tumor, mucosa and healthy tissue were distinguished. The main observation was that, combining images from both cameras led to a better classification performance than using only either of them.

Oral cancer is a significant health problem, which is typically detected at the later stages after which treatment becomes ineffective. It is sometimes difficult for physicians to discriminate localized oral cancer from other benign conditions. In order to non-invasively detect tumor in tongue, a medical HSI system based on reflectance data was used with a Sparse Representation algorithm [26]. It distinguished the healthy part from the tumor affected part of the tongue, based on spectral signal at each pixel. Another study based on a snapshot HSI imaging system, utilized reflectance spectra to segment the tissue and fluorescence spectra to highlight suspicious regions [27].

Two types of skin cancer, namely Kaposi's sarcoma and melanoma have been analyzed using hyperspectral/multispectral imaging. Melanoma is considered the deadliest form of skin cancer. Kaposi's sarcoma was identified in [28] using a NIR range based six-band multispectral camera in which the thermal signatures of the patient's blood volume were studied, and it was observed that blood oxygen saturation levels and blood volume were indicators of tissue angiogenesis and metabolism. In an *in vivo* study, non-invasive tumor margin identification for early and malignant melanoma was performed using a hand-held HSI camera and a Decision Tree classifier, based on feature selection and dimensionality reduction techniques [29].

A research involving classification of cancer cell cultures experimented with different cancer cells from pancreas, breast, liver, colon, bladder and vascular endothelium [30]. From the hyperspectral data, spectral features were extracted by dimensionality reduction using PCA. These features were then classified using a multilayer perceptron (MLP) based neural network, and then classified by an SVM classifier in a one-vs-one classification scheme, with ten models developed and trained on the corresponding two tissue types (e.g. pancreas vs liver, pancreas vs breast etc.) and the best performing model was chosen for classification. This also revealed the spectral similarity (pancreas and liver) and variability (breast and pancreas) in cancer from different organs.

Applications in surgical guidance

While the success of a surgery depends on the surgeon's expert judgements and visual ability, complimentary intraoperative tools are generally needed to confirm diagnosis and evaluate surgical therapy in the operation room, specifically visual aid tools like medical HSI. The role of such an imaging tool can be threefold:

1) To aid visualization of tissue in the surgical field that is spilled with blood, which is a big visual obstacle during surgeries.

2) For residual tumor detection, to maximize the removed tumor without harming the adjacent normal tissue. This can be performed real time by observing the spectral difference between tumor and normal tissue.

3) To monitor tissue oxygen saturation, which is a positive indicator of normal tissue. Thus, dynamic changes in blood flow can be captured and untoward incidents avoided.

By utilizing these three aspects of HSI in the surgical room, researchers have explored the possibilities of HSI in some surgical procedures.

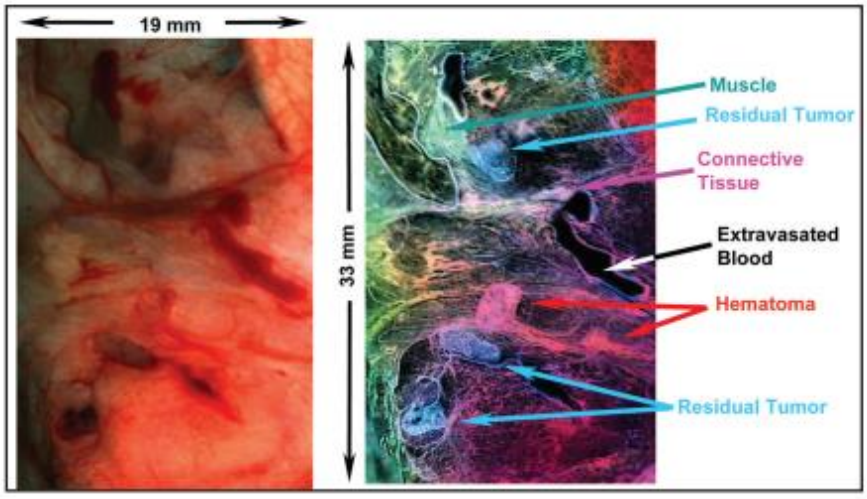


Figure 6: Difference between RGB vs HSI image in detecting features under the surgical bed. Left: image of the surgical bed with the residual tumor. Right: Pseudocolor visualization of the characteristics of the tissue including hematoma under the surface [31].

In 2017, the estimated number of new cases of breast cancer in females in U.S was 252,710 [22]. It was also the deadliest form of cancer in females and a predominant number of patients undergo mastectomy, which is the complete surgical removal of the cancer-affected breast. In some cases, lumpectomy or breast conserving surgery is performed to remove only a selective portion of the breast. While for cosmetic reasons the excised tissue should be kept minimal, it is crucial to completely remove all the cancer cells. Failure to remove it effectively will necessitate a re-excision. This can be avoided if it is possible to make an intraoperative evaluation of the residual tumor in the breast. A study on rats [31], was able to intraoperatively differentiate tumor, blood vessels, muscle, and connective tissue using HSI, after partially resecting the breast tumor tissue (illustrate in Figure 6).

The surgical procedure to remove the gall bladder is called cholecystectomy and is a commonly performed surgery where the standard procedure is laparoscopy. In closed laparoscopy, several small incisions are made in the abdomen to facilitate the entry of surgical tools and an endoscopy-based video camera. This limits tactile feedback and the visualization of tissue. Therefore, a NIR HSI modality with an endoscope was built by Zuzak *et al*. [32], to identify the anatomy of the porcine biliary tissue during surgery by using only the measured spectra inherent to each tissue, before taking any invasive action.

Figure 7: Illustration of an application involving segmentation of abdominal cavity using HSI technique. Left: RGB image of the intestine. Right: segmentation based on spectral signatures [33].

In surgical intestinal ischemia, there is diminished blood flow during which deoxygenated blood and waste products accumulate, in turn causing inflammation and ulcers. During surgery, visibility is crucial to diagnose the disease. Since the abdominal area is vast, HSI can be used to visualize different tissues and organs without any invasive action. Studies on porcine intestine using HSI identified that the spectral range 765 to 830 nm can best distinguish normal and ischemic tissue [33]. Based on spectral signatures, spleen, colon, and small intestine could also be segmented in the hyperspectral image.

1.3.2.3 MHSI applications using deep learning

Ling Ma *et al.* developed a CNN architecture, based on entirely the spectral information from 12 hyperspectral data cubes of head and neck tumor on mice and performed leave-one-out cross-validation for the detection of tumor [34]. Each spectrum obtained from the pixels was utilized to characterize the tissue into normal or tumor affected. In another study [35], hyperspectral data of excised tissue samples of 50 patients was used to classify the spectra into squamous-cell carcinoma, thyroid cancer, and normal head and neck cancer. It was confirmed that the CNN developed, outperformed other classifiers like SVM, k-NN, DTC and LDA. A study furthering this research by the same group [36] was carried out for characterizing tissue in two tissue regions namely, thyroid and oral cavity tissue. For the thyroid tissue, a 3-D CNN based on AlexNet [37] was proposed to distinguish the tissue using binary classification into normal thyroid tissue and thyroid carcinoma, and multi-class cancer classification into normal thyroid carcinoma (medullary and papillary) and multi-nodular thyroid goiter tissue. With the oral and upper aerodigestive tissue, binary classification between normal tissue and squamous-cell carcinoma, and in multi-class classification of normal tissue into epithelium, skeletal muscle and gland. For the oral tissue, an AlexNet-based CNN with convolution-only inception module was implemented. Both the experiments used hyperspectral image patches of dimension 25 x 25 x 91, classifying them into one of the above discussed classes.

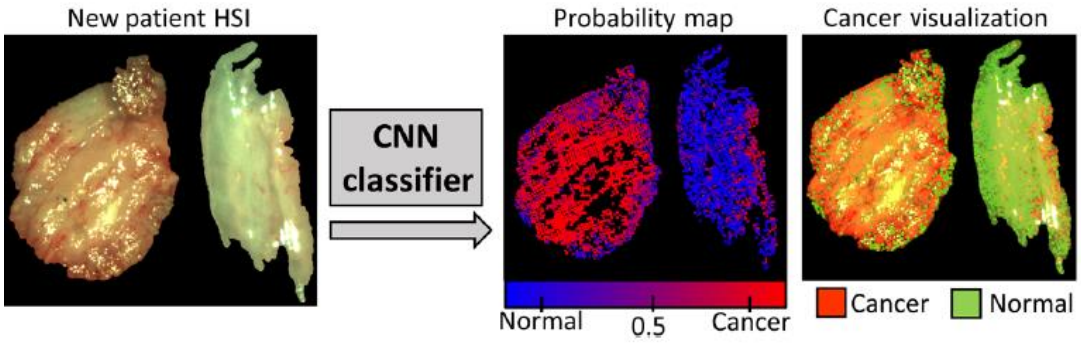


Figure 8: A representative result of spectral binary classification in head and neck tissue. The RGB image, output probability of cancer and its corresponding visualization are illustrated (from [35]).

1.4 Advantages of hyperspectral imaging

1.4.1 Spatial information

As has been discussed, HSI is spectroscopy integrated with imaging methods to obtain spectral and spatial information of a tissue under examination. Therefore, at each pixel of the tissue a spectrum is available for analysis. Due to the spatial correlation of different neighbouring spectra, more accurate models for classification and segmentation can be developed by using the spectral-spatial relationship in the image.

1.4.2 Rich spectral information

In the previous fundamental comparison between RGB/monochrome and HSI methods, the limitations of the former were discussed to establish the advantages of HSI. While RGB/monochrome images record geometric properties, color, gradient and textural information of the tissue, it is usually not adequate to distinguish between healthy and affected tissue. More information about metabolic activity and tissue compositional changes has to be utilized to characterize the affected tissue. Also, RGB color images capture information only at the red (630 nm), green (545 nm) and blue (435 nm) wavelength bands. Due to metamerism, which is the inability to distinguish materials with varying chemical composition but similar color properties, the diagnostic ability of the RGB system limits the surgeon from identifying subtle changes in the tissue properties. In contrast, HSI records spectral information commonly in the VIS to NIR range and stores information across hundreds of spectral bands, which can be invisible to human eye.

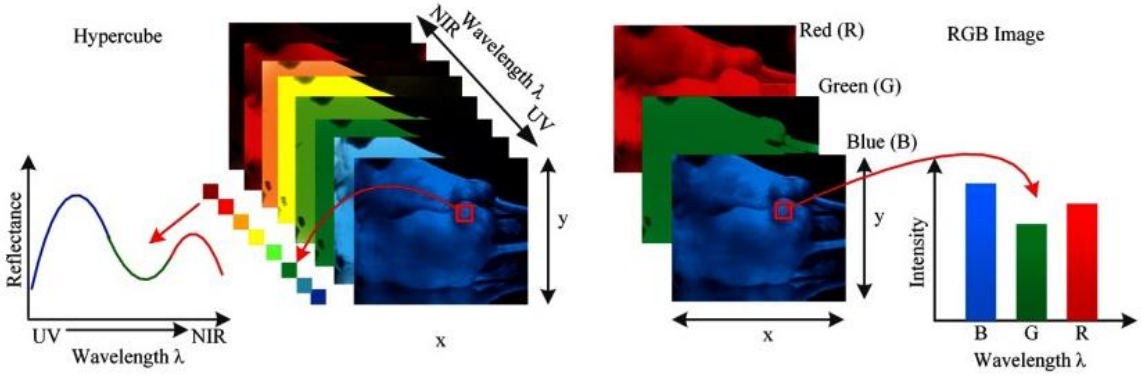


Figure 9: Illustration comparing the structure of a hyperspectral and an RGB image as in [3]. The primary difference is in the number of channels or bands across which information is captured where a contiguous spectrum represents each point in the image, compared to discrete values in RGB.

1.4.3 Non-contact and non-invasive

For medical applications like disease diagnosis and surgical guidance, it is very significant that a non-invasive and non-contact modality like HSI can be employed by making use of only the optical properties of the tissue. Since it is a wide field imaging method with a large field of view (FOV), a vast area of tissue can be analyzed without the need to excise or process the tissue. An illustration of an HSI camera configuration [38] is shown in Figure 10. Further, the non-contact nature can make it suitable for usage in sterile environments, like the surgery room and laboratories.

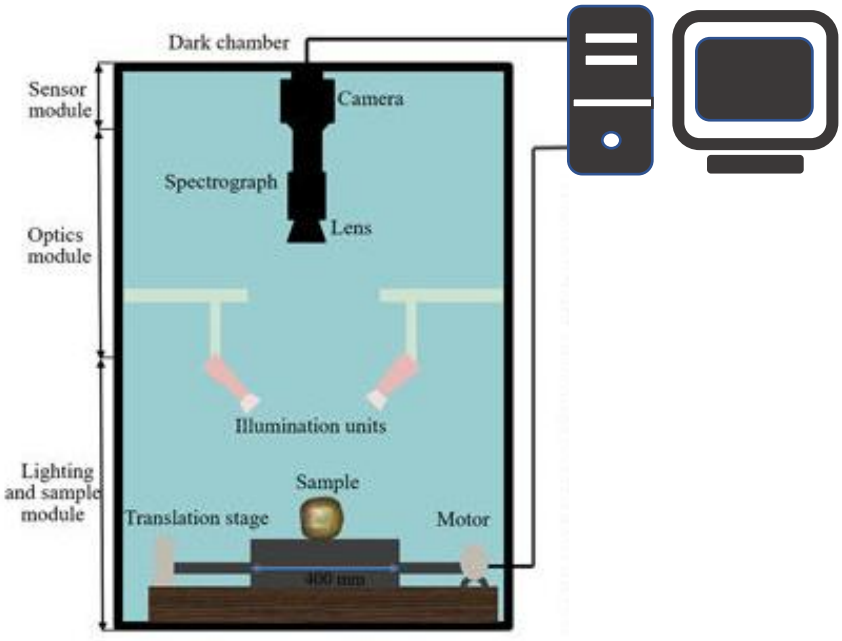


Figure 10: Illustration of an HSI camera setup described in [38] with the corresponding components.

1.5 Limitations of hyperspectral imaging

1.5.1 Signal-to-noise ratio

For HSI, the signal-to-noise ratio (SNR) is an important parameter, defined as:

$$SNR = \left(\frac{A_{signal}}{A_{noise}}\right)^2$$

where, A_{signal} is the root mean square amplitude of the signal and A_{noise} is the root mean square amplitude of the noise.

SNR measures the ratio of useful signal to noise in a measurement. For HSI measurements, each pixel in the image provides the spectrum at that point of the subject. The SNR can be low due to the higher number of spectral channels, effect of background noise, and data from the entire image.

1.5.2 Lack of depth

The optical penetration depth is defined as the tissue thickness, which reduces the intensity of incident light to 37% of the intensity at the tissue surface. The value of this optical penetration depth for an average person is 3.57 mm at 850 nm and 0.48 mm at 550 nm. This limits the application of HSI in medical domain to only investigate tissue areas near the surface. It is also possible that HSI in thermal infrared range can be highly dependent on the surface skin temperature.

1.5.3 Non-uniqueness

The change in biological properties of tissue can be indirectly deduced from the change in the measurements of reflectance or transmittance. This is done through the determination of a spatial map of optical properties under the surface of the tissue using the interaction coefficients (absorption coefficient μ_a and reduced scattering coefficient μ'_s). Therefore, it is possible that photon path is unknown and gives rise to the problem of non-uniqueness where each reflectance value can be represented by more than one coefficient pair. This can lead to two substances with different optical properties yielding similar optical measurements.

1.6 Why deep learning for hyperspectral imaging?

1.6.1 Automatic feature learning

In the previously popular statistical approaches used in machine learning like SVM, k-NN, LDC and even classical neural networks, feature extraction is a crucial step, after which the computer algorithm can optimize the decision boundary in the usually high-dimensional feature space. It is required to extract highly discriminant features that contain the most information of the data representation and is usually done by domain experts, thus being termed *handcrafted* features. This process is very cumbersome and time-consuming, hence the need to automate it and make an intelligent system learn the data representations in a highly optimized way.

Backpropagation is the fundamental cog for the recent advances in supervised learning, which is basically a gradient descent-based learning method for neural networks. In this, a loss function which is constituted by the training data and the network together is minimized with respect to the weights in the hidden layers of the network. By optimizing these weights to a minimum error mapping between the predicted values and the true values, the feature representations are learned. Thus, the step of feature extraction is absorbed within the learning step (with the exception of minor preprocessing steps) making it easier even for non-experts to analyse data, especially in the medical domain.

1.6.2 Generalization ability

By learning the hierarchical representation of data, the deep learning models can outperform the classifiers like SVM, which depend on handcrafted features. Deep models can learn features at multiple levels of abstraction, though a network with higher capacity can memorize the training dataset. By utilizing explicit regularization methods like weight decay and dropout, the generalization ability of networks can be improved. By using early stopping of training or by utilizing batch normalization, the generalization can be implicitly improved. The same cannot be said for other typical classification frameworks working on handcrafted features.

1.6.3 High dimensional data

Kernel methods like SVM are theoretically appealing because of the loss function to be minimized is convex, and in principle, a suitable choice of kernel should be able to learn any training data. Still, they have been rarely used in large-scale experiments involving high-dimensional data (order of 10^4 variables) because they are computationally very intensive and thus cannot scale easily to larger datasets. Additionally, the single-layer nonlinear

transformation of these kernel methods can only have a limited representation capacity to learn the rich features from image data (2-D, 3-D or 4-D). In contrast, deep learning networks have abundant nonlinear transformations that can learn the decision boundaries much easily for varied and complex data. Additionally, researchers are pushing efforts to make deep learning models highly robust, scalable and distributable.

Chapter II – Data Preprocessing

For this Master's thesis, the *ex vivo* tissue data from head and neck region, specifically from tongue is acquired from an HSI setup according to patient number. For each patient, there are a set of images which capture the tumor affected tissue. First, there is the raw hyperspectral image data acquired at different timeframes and an associated header file which contains information about the wavelength range, wavelength values of individual bands, image size and data format. It can be observed by reading the header file of the raw data, there are 192 wavelength bands ranging from 478 nm to 922 nm, with a mean wavelength difference of 2.79 and it is possible to identify certain bands which hold no information. These bands can be clipped off to obtain only valuable information from the raw hypercube. Since the data is obtained from pushbroom scanning of HSI imaging, the information about the number of lines covered while scanning and the number of samples acquired from each line scan is also available. As discussed in the previous section on HSI scanning, the band interleaved by line (BIL) image encoding is used for generating the hypercube. By writing a small MATLAB code, it is possible to read the raw data using the *multibandread* function and convert it into a 3-D array representing the spatial dimensions and the spectral dimensions. It is also possible to view individual images or a representative mid-range image to evaluate the hypercube, when it comes to choosing the region of interest. Each image pixel holds a value of the *uint16* data type

Second, there is a high resolution RGB image of the affected tissue for a given patient. Apart from this there is pathological slide image, which is obtained from slicing the tissue block and staining it to distinguish different cells under the microscope. On such a pathology slice, hand drawn markings indicating the different regions in the slice like tumor, muscle, fat and epithelia are made. To the pathology image, color thresholding and edge segmentation has been applied so that that a region of interest (ROI) mask can be applied to eliminate the insignificant portions of the image. Then the annotations are sketched manually on this annotate image to demarcate different portions in the tissue according to the color scheme red – tumor, green – muscle and blue – fat.

Since the RGB and the pathology images are captured at different time instances and using different sensors, it is necessary to establish a mapping between these two images so that the different tissue regions match. This is carried out by manually selecting multiple points in both these images which match and performing a geometrical transformation of the annotated image with respect to the coordinates of the RGB image. Through this process, a one-to-one mapping of the RGB and annotated images is established. More on this topic will be discussed in the Label Preparation section.

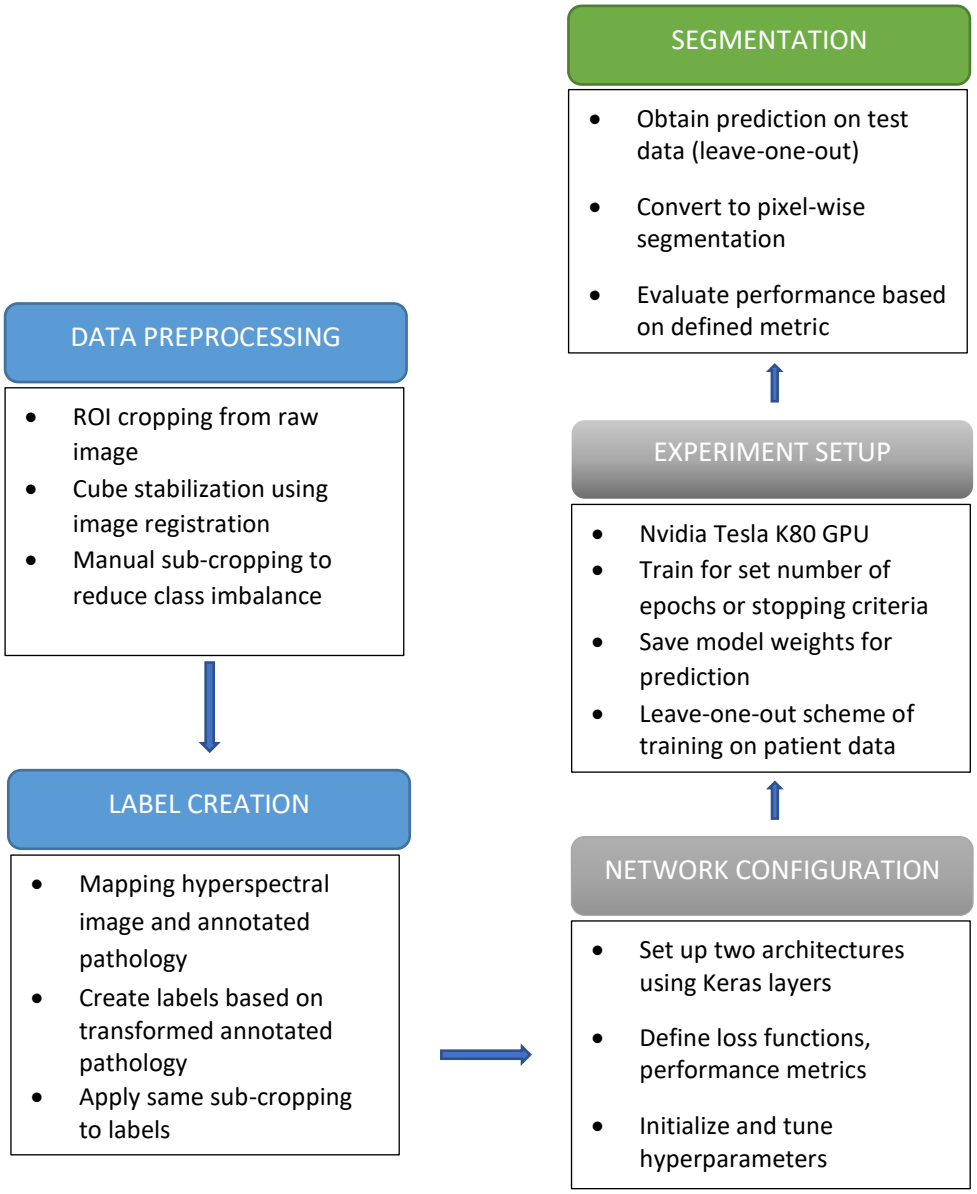


Figure 11: The complete workflow of the thesis. It starts from preparing training data from the patient records till obtaining the final segmentation images from the trained network.

Commonly, raw medical images require some preprocessing before they can be used for any application. In this case too, certain image preprocessing methods were required so that it can be used in training and testing the deep learning network for tissue characterization.

2.1 Image cropping

As mentioned previously, the raw data cube comprises high resolution images (for instance 2048 x 1155), hence it is necessary to crop the images when dealing with voluminous data like

the hypercube, in order to reduce the computation time. Therefore, a representative image slice from the mid-range band (677.54 nm) is extracted from the data cube and its region of interest is estimated. A cropping boundary is drawn around this region using a MATLAB code and the entire data cube is cropped using that cropping boundary. There is however a caveat to this, since a proper cropping boundary is required to be chosen in order to accommodate the translation error (discussed later) in the hypercube across all the bands. By cropping the images too close to the periphery of the tissue, we may risk a portion of the tissue being cropped out of the region of interest in the image. Hence a suitable margin around the tissue is considered while cropping the data cube.

2.2 Image resizing and rotating

It is also important to verify if the RGB and pathology images are in the same orientation as the representative hyperspectral image. Suitable image rotations are performed on the RGB and pathology images until they are aligned with the hyperspectral image. In the previous data cube cropping step, the ROI is selected, and the cropping area is drawn, thus reducing the dimension of the images in the hyperspectral data cube. In this step the high resolution RGB and pathology images are downsized to one dimension of the hyperspectral image, thereby maintaining the aspect ratio of the matched RGB and pathology images. It may also be necessary to apply rotation to some of the images, so that the medical image record (hyperspectral, RGB and annotated pathology) of each patient are of the same orientation.

2.3 Cube stabilization

Image registration in medical imaging is the one-to-one mapping between the coordinates in one image to another, such that points that represent the same anatomical feature are mapped together. For this a geometric transformational model is established between the two images, which can involve rotation, translation, scaling and affine modes. Thus, a moving image, the image to be mapped to the reference image or fixed image, is transformed into the registered image based on two different registration methods: (1) Feature-based; and (2) Intensity-based registration. In MATLAB, the feature-based method can be employed using the Computer Vision System Toolbox, which can detect image features like corners and blobs between the moving image and the fixed image and estimate a transformation. The intensity-based method can be implemented using Automatic Image Registration with the *imregister* command, with which a similarity metric between the moving and fixed image is maximized or minimized using an optimizer to obtain the required geometrical transformation.

In the case of hyperspectral images, which are hundreds of stacked grayscale images, the latter method was easier to work simply because of the volume of image required to be processed (for cube stabilization, which is described next), also the difficulty in detecting tissue features from the data cube in the former method.

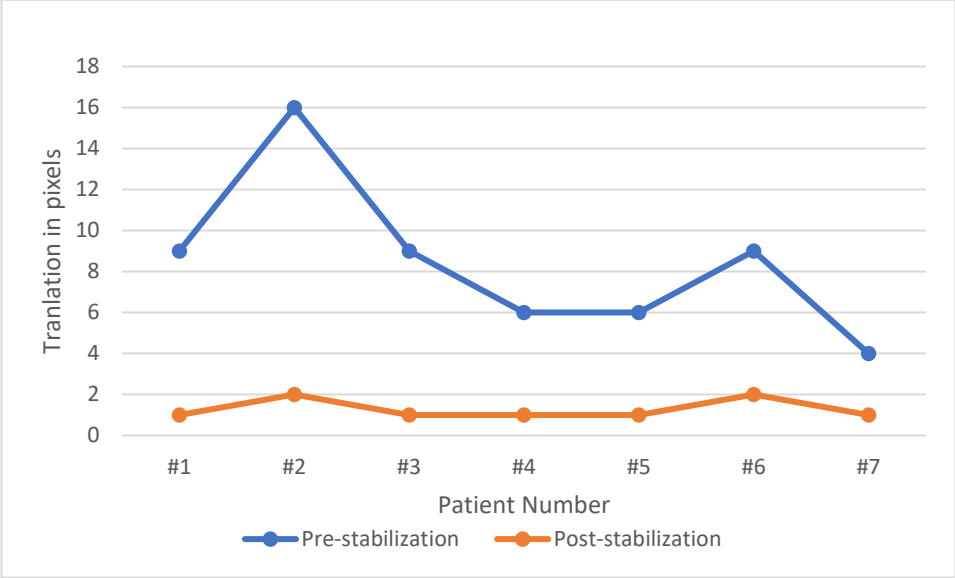


Figure 12: Translation (in pixels) between initial and final bands, across all patient records. It can be observed that this effect is prominent in certain samples (like #2) than in others before stabilization.

As discussed previously, there is discernible translation motion in the image sequence that constitutes the 3-D hypercube. This is the most undesirable effect when working on a volume of images, since there can be only one labelled image for the entire hypercube and the translations of the tissue areas could misrepresent tissue areas to the during network training. For instance, in tissues with smaller tissue regions, it is possible that one type of region is misrepresented as a different one: muscle encroaching upon the tumor region. Since there is already class imbalance between tumor and the healthy tissue, it becomes critical that the translation of the image sequence is prevented. This procedure maybe called the 'data cube stabilization'. For the cube stabilization, two different strategies were attempted based on which images in the data cube are chosen as the moving and fixed image:

1) The image at the mid-range band, which belongs to the visible spectrum (red) is chosen. All the other images in the data cube are set as the moving images and thus the entire cube is stabilized with respect to the mid-range image.

2) The image at the mid-range band is chose as the fixed image In. Its neighbouring images I_{n-1}, I_{n+1} are set as the moving images. Once these moving images have been registered, they become the fixed images to the preceding or succeeding images, respectively. Thus, with two channels of registration starting from I_{mid} to I_1 and I_{mid} to I_b (where b is the number of bands in data cube), the entire data cube is registered.

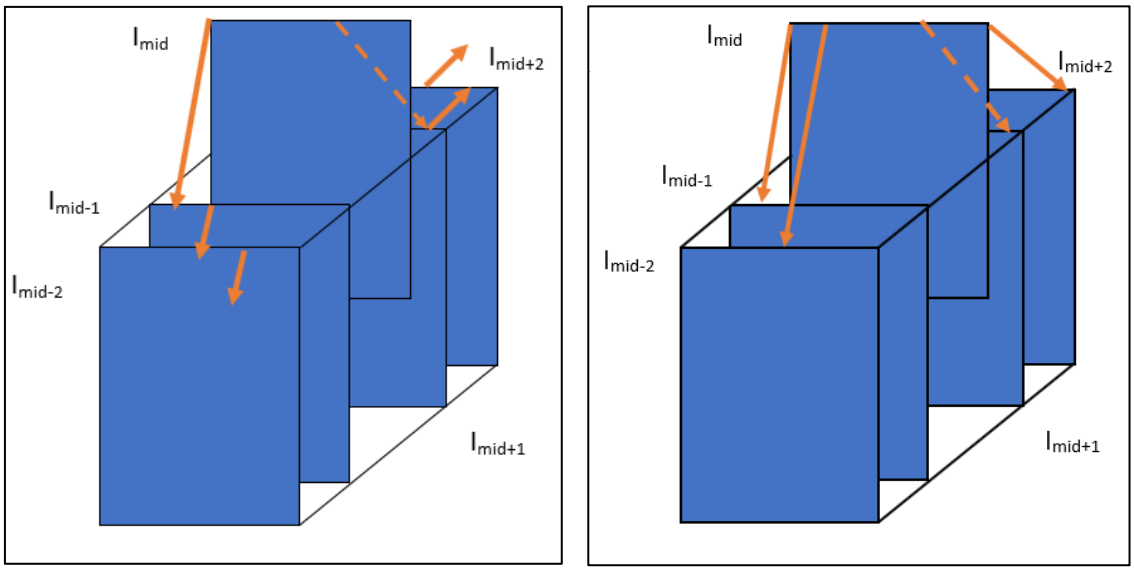


Figure 13: Illustration of the two cube stabilization methods utilized. Left: Median band is chosen as the reference (fixed) image and the adjacent bands are moving images. These transformed bands become the reference for the subsequent bands up and down the range. Right: Median band is chosen as the reference image and all other bands in the spectral range are considered as moving images.

With respect to the registration performance the first method managed to stabilize the data cube by curtailing the translation observed in the data cube (Figure 12). On the other hand, the second method could only reduce the translation to certain extent in a few bands, exacerbated by the bad image quality in certain bands and, also by the removal of uninformative bands. Thus, the error in registration progressed up or down the cube, depending on which bad quality image was assigned as the fixed or reference image.

2.4 Label Preparation

2.4.1 RGB – hyperspectral image registration

The ground truth labels are generated from the pathology images, gold standard for medical image annotation, which were already matched to their respective RGB images. Now, in order to obtain the ground truth to be overlaid on the hyperspectral data cube, a suitable image matching or registration method is required to convert the pathology annotations to the coordinates of the hyperspectral image. Thus, it is decided to establish a one-to-one mapping between the RGB image and the hyperspectral image, which would automatically establish a

mapping between the hyperspectral image and the annotated pathology. However, this was not a straightforward method like the image registration used for cube stabilization; visually, there were mismatches between the RGB images and the representative hyperspectral images in terms of rotation, scaling, orientation and even deformation of tissue. In the dataset, only a few images were image registered using the Automatic Intensity-based method and for the remaining images a robust registration mechanism is needed: Control Point registration, which is the manual mode of image registration.

2.4.2 Automatic vs manual

In the automatic method, multimodal registration is chosen since the RGB and hyperspectral images were captured using different sensors. Since the representative hyperspectral image was obtained from the mid-range band (in this case, 677.54 nm), which belongs to the red wavelength range, the logical choice would be to use the red channel of the RGB image for image registration. It was also observed that using other bands or converting the RGB image to grayscale using *rgb2gray* command did not yield satisfactory results.

Thus, using *imregister* command, with red channel of RGB image as the moving image and representative hyperspectral image as the fixed image, the automatic image registration can be implemented. While in most images it yielded a reasonable global image registration in terms of scaling, rotation etc., the local registration in terms of the anatomical tissue features to be matched was incomplete. Therefore, a two-fold image registration method involving both automatic and manual methods is developed.

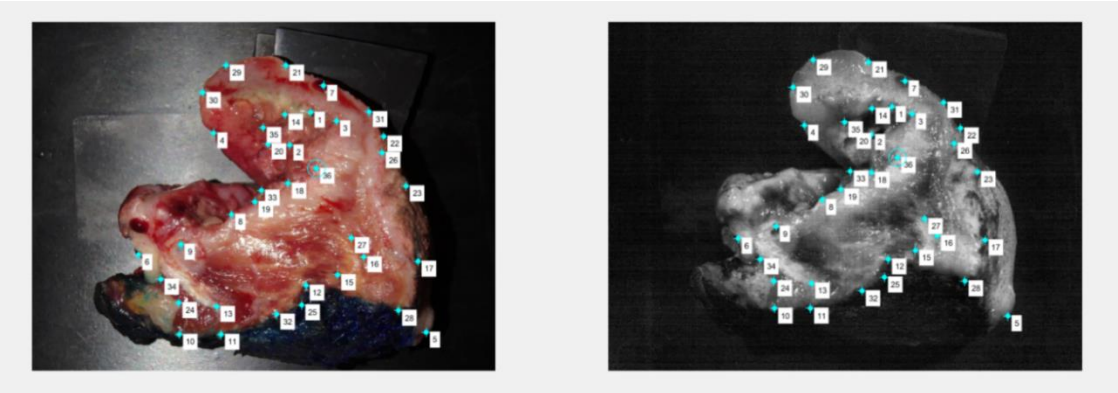


Figure 14: Collection of image point pairs by the Control Point method in MATLAB. Left: RGB image of corresponding to #3. Right: Representative hyperspectral image of #3. It is to be noted that the points are focused around the tumor affected regions for remove local image distortions.

After removing gross or global image distortions, it is possible to use the transformed image in the Control Point method to manually select point pairs of the anatomical features to be matched in both the images. This is critical especially in the tumor areas of the tissue - which

are sparse compared to the muscle or fat areas - where inaccurate point mapping can lead to adverse effects on the deep learning performance. This way, local registration can be performed by choosing the points from the desired areas in both images and saving those image point pairs as *movingPoints* and *fixedPoints*, corresponding to the moving image and fixed image (

Figure 14). A geometrical transformation (similarity) is estimated from these point pairs and applied to the fixed images. By iterative transformations, accurate image registration becomes possible. The different image registration methods implemented on different patient hyperspectral images is shown in Table 3.

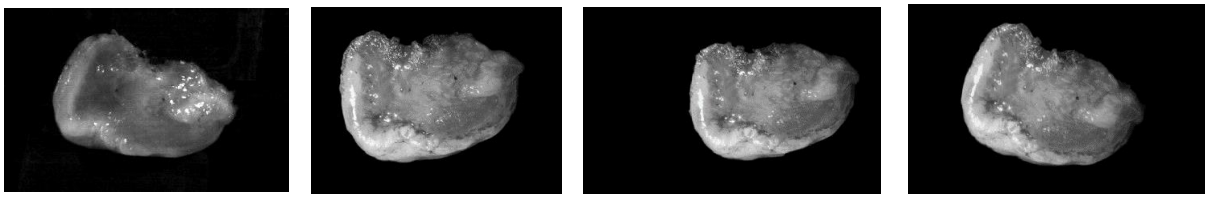


Figure 15: Images of a tongue affected by tumor. From left to right: Representative hyperspectral image; Grayscale of RGB image; After Automatic Intensity-based image registration; After Control Point image registration.

Patient sample	Global transformation	Local transformation
#1	Automatic	Control point
#2	Control point	Control point
#3	Automatic	Control point
#4	Automatic	NA
#5	Automatic	Control point
#6	Automatic	Control Point
#7	Automatic	Control point

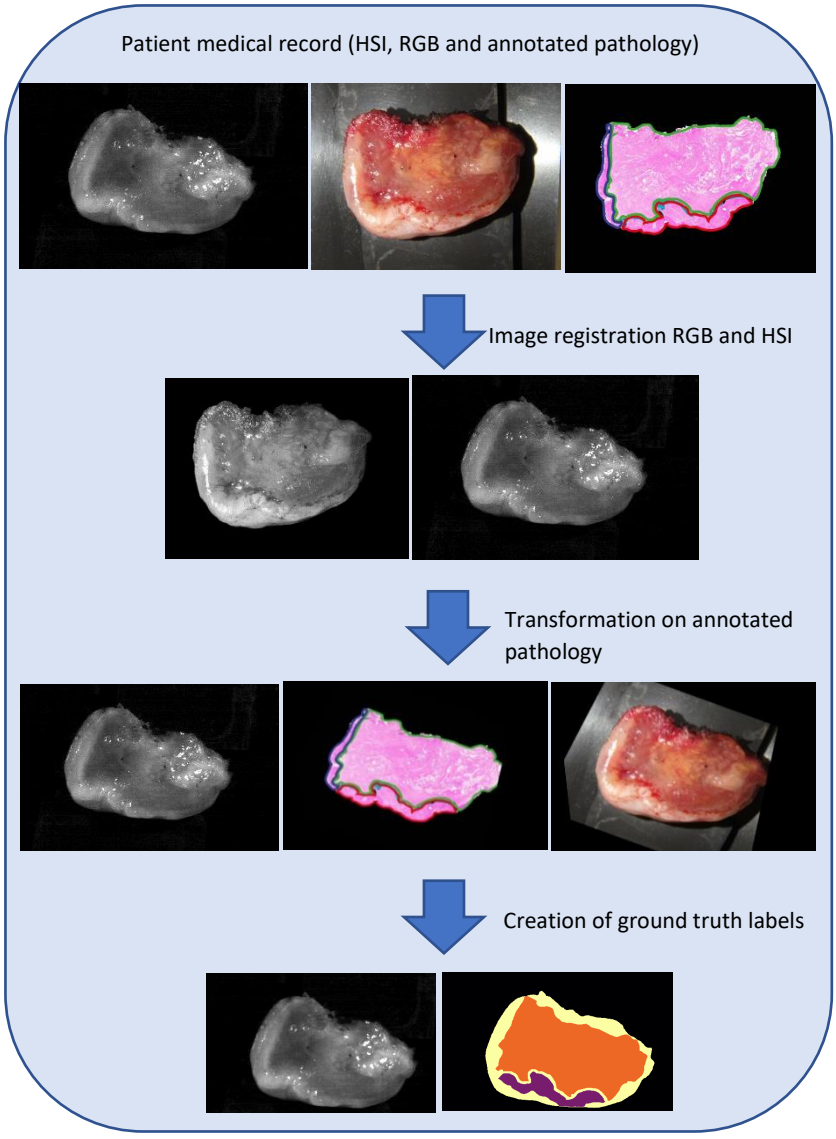
Table 3: The method utilized for mapping the RGB image to the hyperspectral image coordinates globally or locally for different patient samples.

2.4.3 Label generation

From the transformed annotated pathology images, we can obtain the segmentation label images for training the deep neural networks. Clearly, there are three categories in the annotated pathology slides of the excised tongue tissue: tumor and muscle (and fat in specific samples). However, on comparison with the hyperspectral image, we can see that some portions of tissue have not been annotated by the pathologist. This is because thin slices of the tissue block are made and annotated by the pathologist. Depending on the number of slices made and the elevation found in the tissue block, regions of the tissue may not be available for

the annotation. Thus, overlaying the label image on the original hyperspectral image would not provide complete annotation. The regions which are not covered by the pathology slide can be denoted as an 'uncertain' or 'unknown' category in the label, thus making it a label of four categories or classes namely tumor, muscle, unknown and background. Since it is a pixel-wise annotation, the regions of the tissue class can be given a pixel value like, 0 for tumor, 1 for muscle, 2 for unknown tissue and 3 for background.

The unknown tissue region can be easily obtained from a hyperspectral image channel, by performing thresholding and relevant morphological operations to obtain the outermost tissue regions not annotated by the pathology slides. The labels can be converted to a categorical image, which has the dimension (x, y, n=4), where n is the number of defined tissue classes and each tissue class is represented as a separate channel of the categorical image. This makes it easier to work with multiclass labels, especially within the TensorFlow framework, which will be discussed later. The complete process of label generation is illustrated in Figure 16.



2.5 Dataset preparation

Figure 16: Illustration showing the process of ground truth labels creation. From the annotated pathology, RGB image and hyperspectral images from original patient records, the ground truth labels are created by registering the pathology and the hyperspectral images.

Once the labels are available, we can assess the regions that are constitute a particular class, for instance, muscle, tumor or unknown tissue. Immediately, we can observe that the pixels in the label images (or hyperspectral images) that are categorized as tumor are far fewer than the pixels categorized as muscle. In Table 4 shown below, the ratio of number of pixels belonging to each class based on ground truth labels.

Table 4: The ratio of number of pixels belonging to different classes based on the labels before class balancing, in the form background : tumor : muscle : unknown.

Patient sample	Background : Tumor : Muscle : Unknown ratio for pixels per class
#1	10 : 1 : 1 : 2
#2	22 : 1 : 7 : 3
#3	10 : 2 : 2 : 3
#4	44 : 3 : 5 : 4
#5	45 : 1 : 13 : 8
#6	20 : 1 : 8 : 2
#7	40 : 1 : 3 : 2

This class imbalance can create a bias towards the tissue classes that have the maximum number of pixels (in case of hyperspectral image, spectra). In order to counter this class imbalance in tumor pixels compared with the remaining class pixels, an explicit sub-cropping scheme is introduced to ascertain that adequate pixels are represented in the crucial tumor class.

In this cropping scheme, the label images of all the samples are examined and manual cropping is performed on them. The cropping in done in an overlapping manner, such that the spatial dimension is 224 x 224 and most of the cropped regions have tumor class pixels and also limiting the regions representing the background, while making sure at least two class regions are present in each cropped region. The number of pixels represented by each tissue class, post-cropping for class balancing can be seen in Table 5.

When the 224 x 224 cropped regions were made, the coordinates of the crop (x, y, height, width) are stored separately so that they can be applied to the hyperspectral data cubes and the corresponding correct blocks can be cropped from an original hyperspectral data cube. Using

this cropping scheme, we can obtain 31 sub-blocks (224 x 224 x 164) and their corresponding label images (224 x 224 x 4). This number (Table 6) varies for different hyperspectral data cubes because the spatial size of the ROI crop is different (size of the tissue in the image is different). This constitutes the data set preparation step, which can now be read easily in Python using the H5PY library.

Table 5: The ratio of number of pixels belonging to different classes based on the labels after class balancing, in the form background : tumor : muscle : unknown.

Patient sample	Background : Tumor : Muscle : Unknown ratio for pixels per class
#1	6 : 2 : 1 : 2
#2	4 : 1 : 8 : 2
#3	4 : 4 : 5 : 3
#4	7 : 8 : 2 : 7
#5	4 : 1 : 10 : 4
#6	5 : 1 : 9 : 2
#7	5 : 2 : 10 : 2

Table 6: Number of sub-cropped regions obtained from ROI hyperspectral image of each patient sample.

Patient sample	Number of sub-cropped regions
#1	31
#2	32
#3	27
#4	31
#5	32
#6	31
#7	32

2.6 Spectral signature analysis

In this section, individual patient data cubes are studied by plotting their spectral curves for the tissue classes including tumor and muscle. The unknown and background classes are not considered in this study because the unknown class spectra have similar profiles to muscle, while the background spectra are not usually prominent compared to the other classes. Therefore, to eliminate clutter and facilitate better understanding of inter-class difference between the tumor and muscle spectra, the other two classes are not considered for the following plots. They are plotted on their reflectance values across all the 164 spectral bands. By using the labels coordinates(ground truth), spectra belonging to tumor and muscle tissue are obtained. The mean values of all the spectral signals are calculated along with their standard deviation, for both the tissue classes. This allows examining the spectral signatures for each individual patient sample.

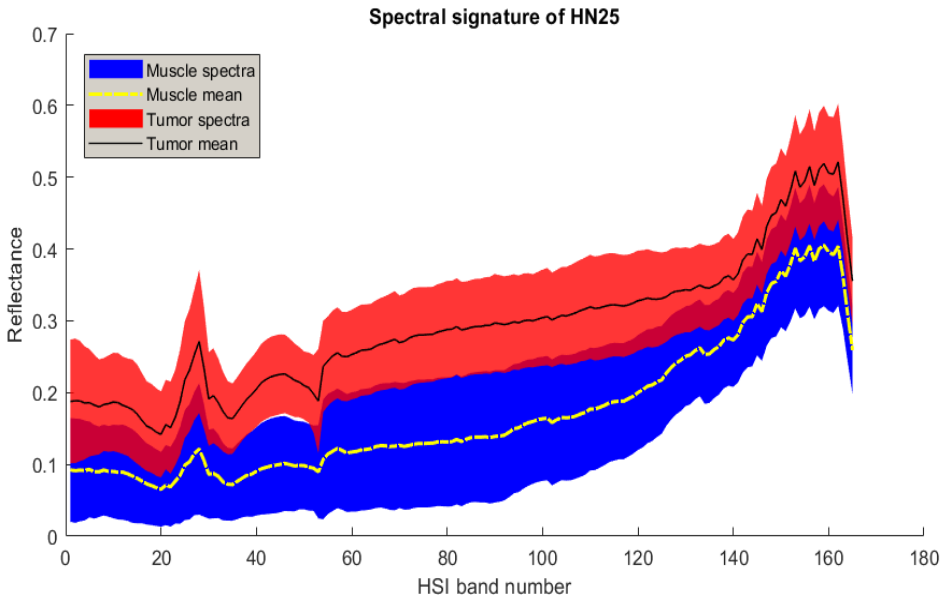


Figure 17: Spectral signatures of the #1 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue.

As can be seen from Figure 17, this patient sample has little inter-class overlap of spectral values, which indicates that it can provide good distinction between tumor and muscle class. Given that the mean tumor spectrum is similar in many characteristics to muscle spectrum, by having a separation in the intensity values, it is possible to delineate the hyperspectral data cube by classifying the individual spectra into one of the possible four categories.

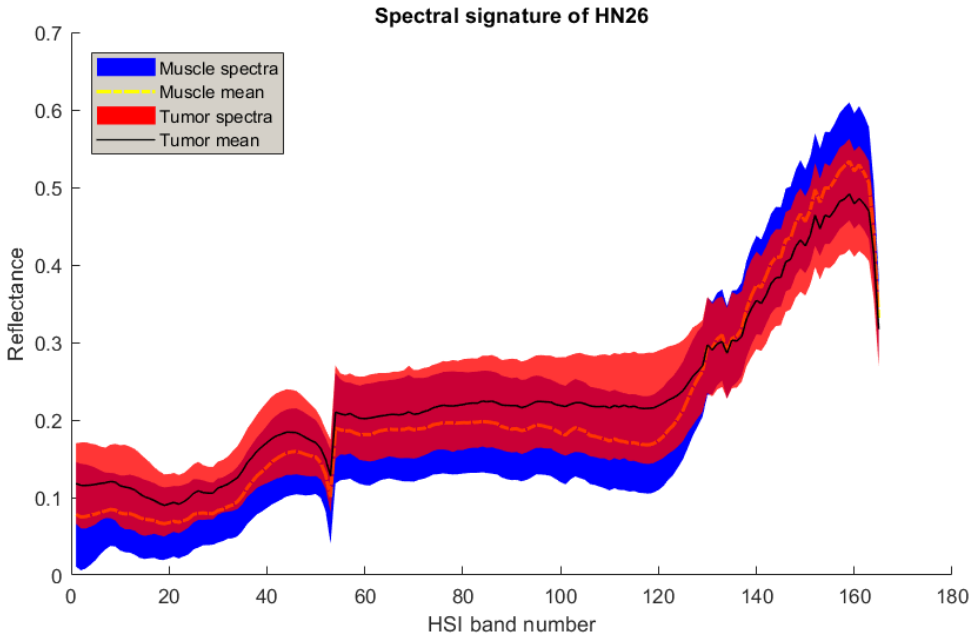


Figure 18: Spectral signatures of the #2 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue.

For the second patient sample, considerable overlap in the plotted spectra of tumor and muscle is observed. The overlapped area is represented by the darker shade of red and it is also worth noticing how close in intensity the average spectra of tumor and muscle are to each other. This decreases the inter-class separability when approaching the segmentation problem from the perspective of spectral information.

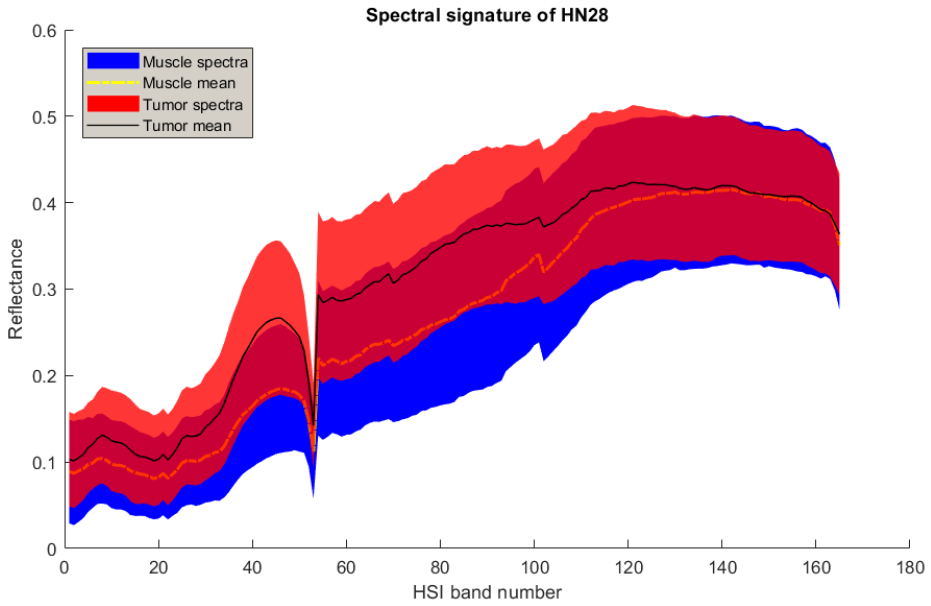


Figure 19: Spectral signatures of the #3 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue.

In the next patient sample #3, (Figure 19), there is again overlap of the spectra, but there is distinction in the spectral intensities between bands 35 and 100, which can make these bands most informative in discriminating between tumor and muscle. One other observation specific to this sample is there is significant tissue curvature, which can raise the intensity of the spectra belonging to the curved or raised regions of the sample. Adding to this, the inter-class spectral overlap could potentially deteriorate the spectrum discriminating ability of a classifier.

In this particular hyperspectral data cube (Figure 20), there is higher inter-class separability in terms of the spectral intensity, as seen from the minimal overlap (especially along the lower bands) and the separation between the average muscle and tumor spectra. In a stark contrast to this, Figure 21 representing the spectra of the patient sample #8, shows the least inter-class separability with complete overlap of the spectral curves. The hyperspectral images (and spectra) are low intensity and it is not viable to apply preprocessing techniques to improve the separation between the two classes. This can indicate that this data cube was not acquired properly from the HSI setup.

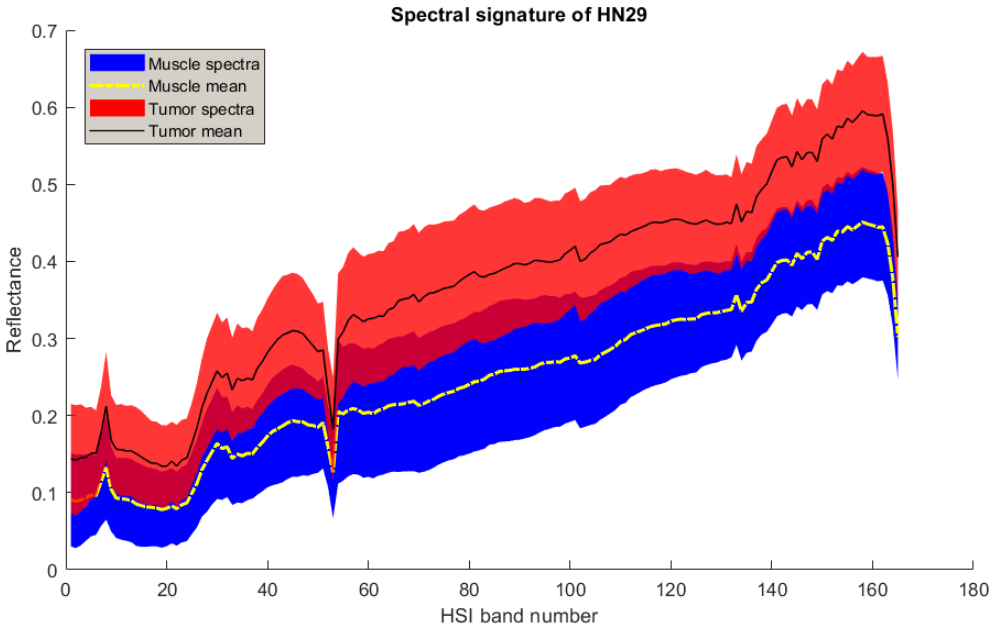


Figure 20: Spectral signatures of the #4 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue.

From the spectral plots (Figure 22) of sample #5, we can notice overlap and small intensity difference between average spectra of muscle and tumor tissue (especially in the mid bands 60 to 120). This effect worsens as we move across the bands and this separation is non-existent in the final bands of the hyperspectral data cube. For sample #6, shown in Figure 23, there is still overlap but not to the extent of the #5. There is a window upto the 120th band where the inter-class separation is still prominent and can be used to distinguish the tumor and muscle spectra.

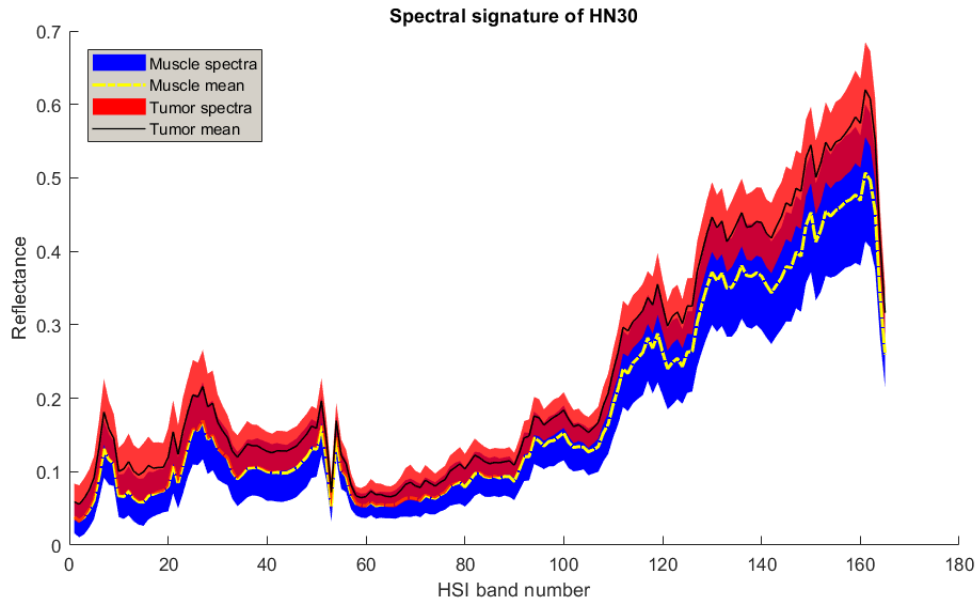


Figure 21: Spectral signatures of the #8 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue.

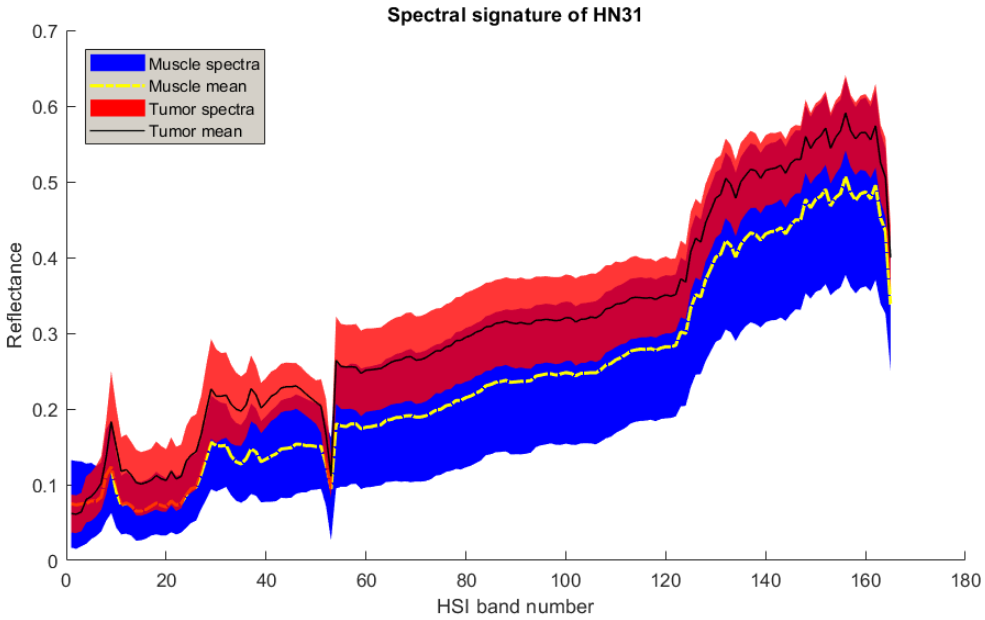


Figure 22: Spectral signatures of the #5 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue.

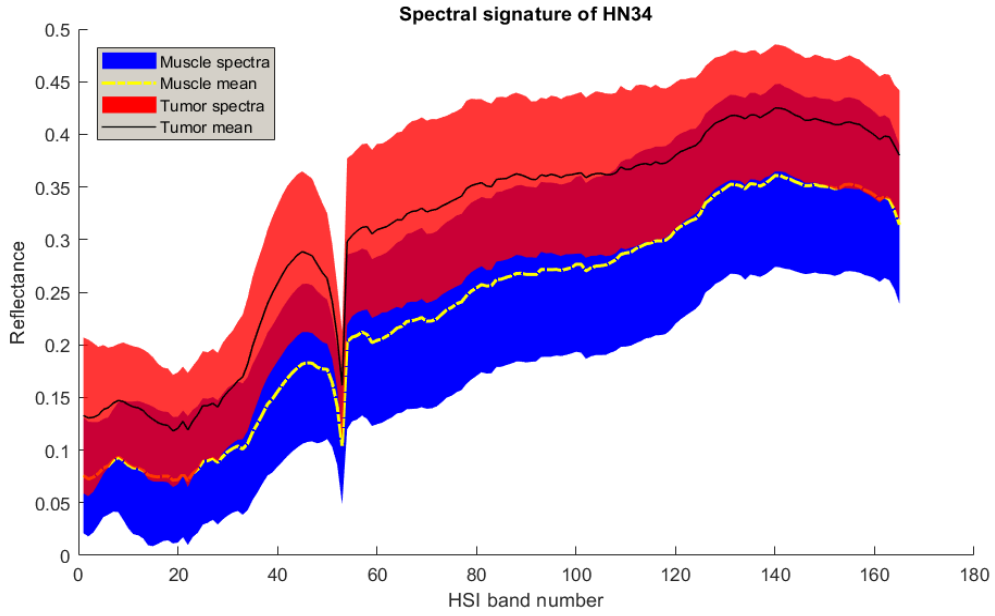


Figure 23: Spectral signatures of the #6 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue.

In the final patient sample #7 shown in Figure 24 , while there is spectral overlap along the initial bands, the inter-class separation improves strongly in the mid-range, starting from band 60 with peak separation happening around band 120. Thus, this spectral window could be informative in discriminating between the muscle and tumor class.

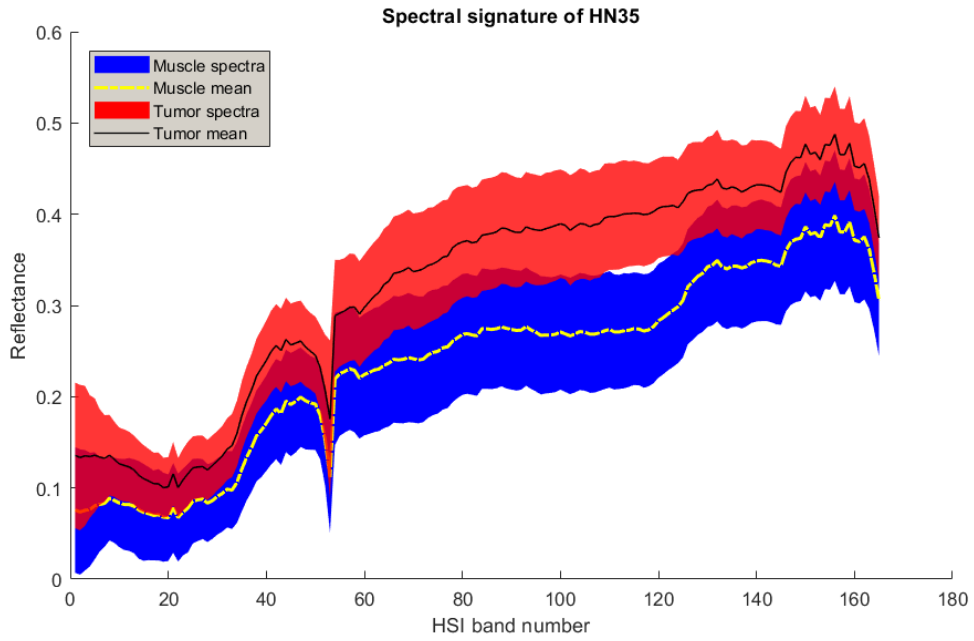


Figure 24: Spectral signatures of the #7 patient sample. It shows the confidence interval around the mean spectra of tumor and muscle tissue.

2.7 Discussion

This spectral analysis of all the patient hyperspectral data cubes reveals how informative they are in distinguishing between the tumor and muscle class spectra. By using the confidence interval about the mean of the spectra, the inter-class separation can be visualized. The separation is non-uniform across different samples, with #1, #4 and #7 showing distinct separation between tumor and muscle. Samples #2 and #8 show complete overlap between the classes, which means that the separation between both the classes is small or, the hyperspectral data cubes acquired from these patient samples have considerable similarity in the spectral profiles of tumor and muscle tissue. The remaining data cubes have spectral overlap to certain extent, but also have maximum separation across a few discriminatory bands, which can be useful when classifying spectra. Hence, inter-patient variability is considerable in the hyperspectral image data set and to what degree it could influence the prediction performance of the proposed networks could be seen in the upcoming sections.

Chapter III – Deep Learning Setup

3.1 CNN theory

Convolutional neural networks or CNNs, are an attempt to model the functioning of the human visual cortex and to replicate the human vision system. It is one of the successful models in machine learning, especially for solving computer vision problems like image classification and object recognition. In this section, the basics of the functioning of the visual cortex are briefly discussed, which can facilitate the understanding of how CNNs work.

The visual cortex is present at the back of the skull, in a region called the occipital lobe and is instrumental in the processing of visual information [39]. Visual information propagates starting from the eyes, through various brain areas, before reaching the visual cortex. V1, primary visual cortex, is the area of the visual cortex that receives the visual signals and is further managed by visual areas V2, V4 and the Inferior temporal gyrus (IT). With a focus on object recognition, it is enough to limit the explanation of functioning of regions V1, V2, V4 and IT as illustrated in Figure 25.

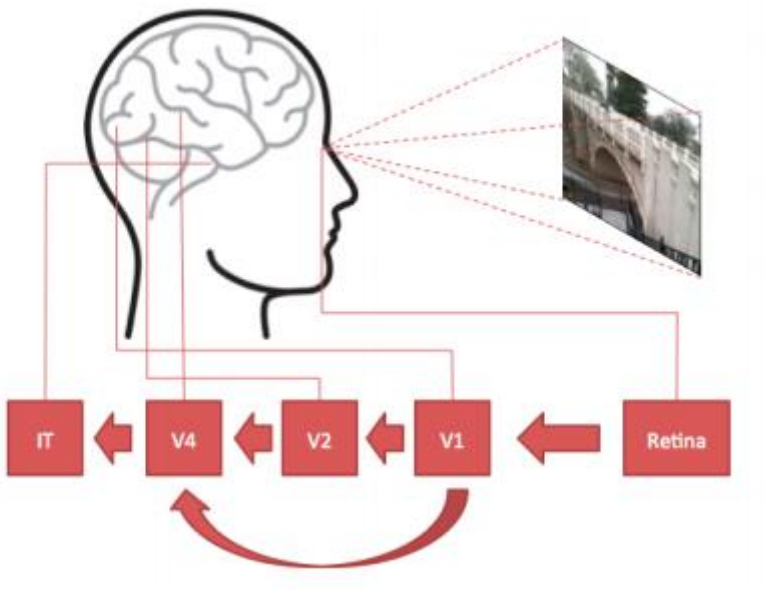


Figure 25: Flow of information from the retina to the visual cortex to the inferior temporal gyrus [39]*. The regions V1, V2 and V4 detect edges, color, geometric shapes etc. from a scene.*

1) In the retina, the visual information is converted into chemical energy, which is in turn converted into action potentials that are transferred to the visual cortex.

2) V1 performs edge detection, where areas with local contrast are highlighted.

3) V2, the secondary visual cortex, extracts simple properties like color, orientation from the signal and some complex properties from the signal from V1 and sends it further.

4) V4 detects features of intermediate complexity, like geometric shapes and it also receives direct input from V1.

5) TI performs object identification based on form and color of the object, while comparing it with the already stored memories of objects to identify it.

A precursor to the development of the Convolutional Neural Network is the Neocognitron [40], which is a learning model for pattern recognition. It consists of an input array and a cascade of modular structures, each with two layers called S-layer and C-layer containing S-cells and C-cells respectively, inspired from the S-cells or simple cells and C-cells or complex cells of the visual cortex. The S-layer serves as a feature extractor while the C-layer is responsible of organizing the extracted features. Local features like edges are detected in the lower layers, while global features like overall shape are captured in the higher layers. This imparts position invariance property to the network, i.e., a pattern is identified precisely irrespective of its position in the image. This eliminates the need to normalize the position of the image patterns and is one of the reasons for the superior performance of CNNs.

Extending this model, we get the Convolutional Neural Network or CNN, which can process data in the form of multiple arrays. In this way, most forms of data can be accepted by the CNN, like signals and sequences as 1-D, images and audio spectrograms as 2-D, and volumetric images or videos as 3-D data. The two most important aspects of the CNN are the convolution layer and the subsampling layer (pooling layer) and the first model combining both these layers was introduced by LeCun for handwritten character recognition [41]. The main difference from the Neocognitron, was the supervised learning stage that happens after the unsupervised learning stage, also called as backpropagation.

3.2 Backpropagation

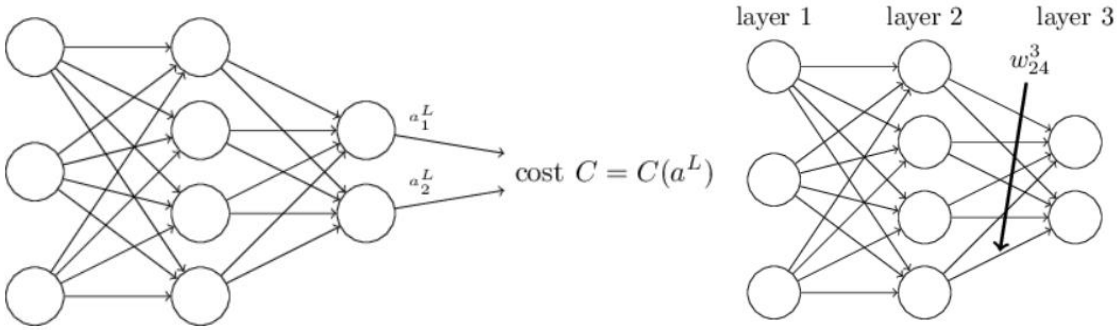


Figure 26: Illustration of intermediate layers in a neural network obtained from [42]. Left: represents the output layer activations and the cost function. Right: represents the weights between neurons in intermediate layers.

The possibility of a neural network to learn from its different inputs is due to the backpropagation algorithm. For explaining this learning method [42], we must define a loss function *C* which depends on the input and the configuration of the network and find the partial derivative of this loss function with respect to the elements, weight *w* and bias *b*. By computing these partial derivatives, we can update these network parameters once a minimum value of the cost function is reached. If *L* is the last layer in the network, then the error term at the last layer δ^L, can be computed by the equation below. The term $\nabla_a C$ denotes a vector which holds the terms of the partial derivative of *C*, with respect to the j^{th} output activation at the last layer $\frac{\partial C}{\partial a_j^L}$. By computing the Hadamard product of this vector with $\sigma'(z^L)$, which is the rate of change of the activation σ at the output layer and z^L its weighted input. The first equation is given by:

$$\delta^L = \nabla_a C \; o \; \sigma'(z^L)$$

The second equation describes the backward propagation of error in the $(l+1)^{th}$ layer, to the l^{th} layer through the transposed weight matrix of the $(l+1)^{th}$ layer and the Hadamard product with the rate of change of activation function with the weighted input z^l. This is equivalent to moving the $(l+1)^{th}$ layer's error δ^{l+1} backward to the l^{th} layer to obtain its error δ^l. The second equation is defined as the following:

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \; o \; \sigma'(z^l)$$

In the third equation we can compute the partial derivative of the loss function with respect to the bias parameter b_j^l, of the j^{th} neuron in the l^{th} layer. It is defined as the following:

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l$$

In the final equation we can compute the gradient of the loss function with respect to any weight in the network w_{jk}^{l}, between the k^{th} neuron in the $(l\text{-}1)^{th}$ and j^{th} neuron in the l^{th} layer (as shown in Figure 26). The final equation is defined as:

$$\frac{\partial C}{\partial w_{jk}^{l}} = \alpha_{k}^{l-1}\delta_{j}^{l}$$

3.3 Network Layers

3.3.1 Convolution

The convolution is a mathematical operation, involving two functions *f* and *g* to produce an integral *h*, which is the output function. The integral represents the amount of overlap of function *f* as it is shifted over the other function g, which is described as:

$$h(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t - \tau)\, d\tau$$

and it can be denoted as $h = f * g$

In a CNN, the convolution operation would be 2-D in case of images and is illustrated in the Figure 27. An input matrix is convolved with a smaller square matrix called a kernel or filter and an output matrix is obtained. As can be seen, the convolution involves the element-wise product followed by a sum between the two highlighted matrices and the same operation is applied by sliding the kernel one column to the right. Once the kernel can no longer be slid right on the input matrix, it is slid down by one row and the operation is continued. In this way, a 3 x 3 output matrix is produced and has reduced dimension compared to the input matrix (5 x 5).

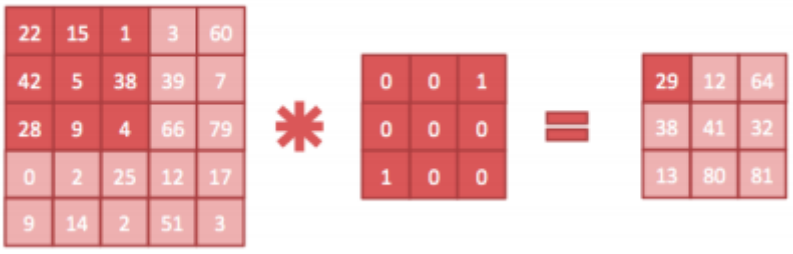


Figure 27: Demonstration of the convolution operation between two matrices. A 5 x 5 matrix is convolved with a 3 x 3 kernel to produce a 3 x 3 output matrix.

The significance of sliding the same kernel across the input matrix is to apply the operations across different regions in the input matrix and also to reduce the number of free parameters by a huge amount, since the same weight is shared by all the units in the output matrix

With respect to an image, the kernel's weights decide what type of operation is applied on the input image. Some examples of image operations include edge detection (Sobel filter), line detection, blur, sharpening, and identity. The convolution of such filters on a sample natural images is visualized in Figure 28 with the corresponding convolution filter activations [43]. In CNNs, each neuron in the convolutional layer is connected only to local region surrounding an input neuron - in contrast to regular neural networks, in which each neuron in a layer is connected to all the neurons in the previous layer – called as the local receptive field.

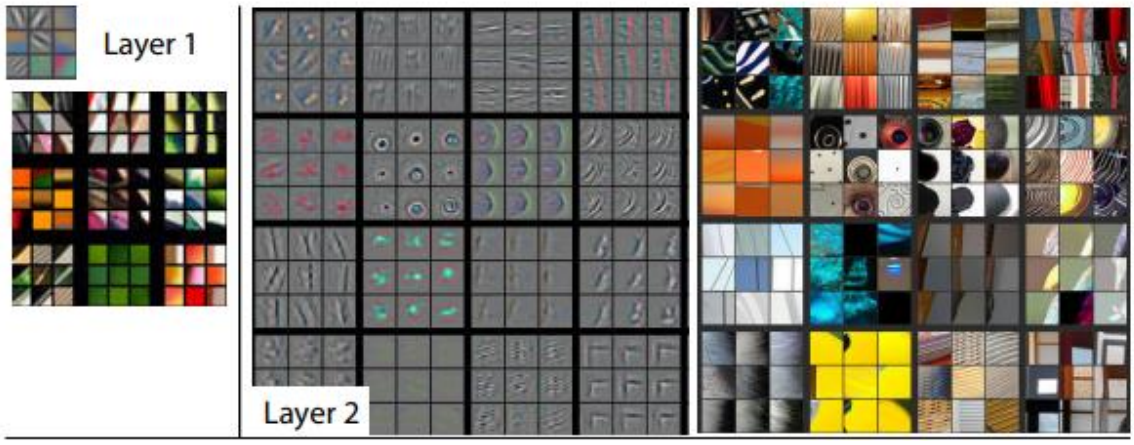


Figure 28: Visualization of the different convolution filters/feature maps [43] that show activations at layers 1 and 2 of a fully trained CNN, with its corresponding original image patches.

The output of the convolution layer is called a feature map, as it is created by the convolution of a filter with an input image and contains information about the features that are present in the image. At each layer there is a bank of filters *m*, which subsequently creates *m* feature maps of the same image. These feature maps are stacked along the direction of the image depth.

When the feature maps after the convolution operation with *m* filters are connected to the non-linearity layer, which consists of an activation function, activation maps are created. The commonly used activation functions are sigmoid or hyperbolic tangent functions, which help to extract meaningful features from the feature maps by squashing the input in the range [0, 1] or the range [-1,1] as shown in Figure 29. The sigmoid activation thus eliminates all the negative values in the image, while the *tanh* function zero centers the input.

However, the sigmoid function's undesirable feature is the tendency to saturate at 0 or 1, which makes the gradient at these regions zero. This can kill the gradient so that no signal flows through the neuron and the network barely learns. Also, since the outputs are not zero-centered,

the gradient update has zig-zag patterns, which is again undesirable. Because of these reasons, the *tanh* non-linearity is always preferred over the sigmoid function in practice.

3.3.2 Activations

ReLU or Rectified Linear Unit is a combination of an activation function and also a rectifier ($|x^{(l-1)}|$), defined as:

$$x^l = max(0, x^{l-1})$$

which is 0 when $x<0$ and linear with slope 1 when $x>0$. The rectifier component is regarded to be quite crucial in the average pooling layer because of the cancellation of negative and positive activation values, which affects the accuracy of the network. This also leads to more sparse activation layers. It has been proven that, including the ReLU layer increases the speed and effectiveness of training [44], and alleviates the vanishing gradient and exploding gradient problem in backpropagation.

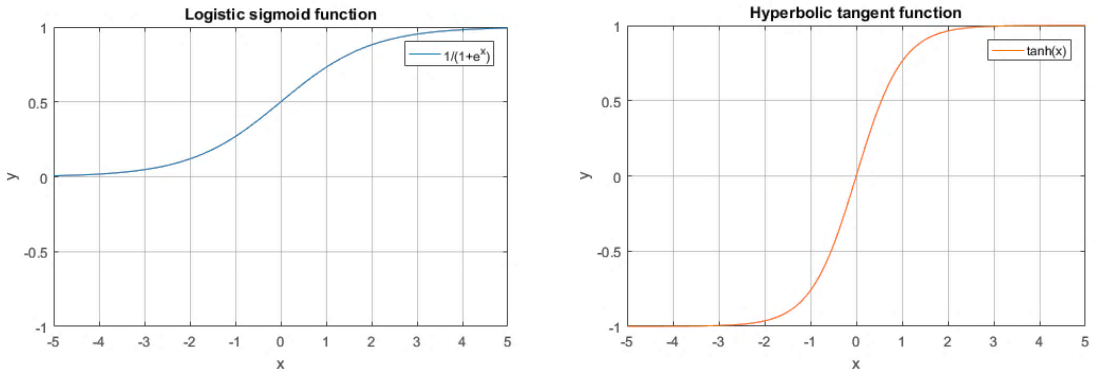


Figure 29: Plots showing the sigmoid function and the tanh function. The sigmoid function operates entirely in the positive range whereas tanh function operates between -1 and 1.

ELU or Exponential Linear Units is an alternative approach [45] to speed up learning process during training and to alleviate the vanishing gradient problem. As can be seen from the plot Figure 30 for the ELU function, it has negative values contrary to ReLU. This allows it to push the mean activations towards zero at lower computational complexity. Research shows ELUs improve the speed of learning and the generalization ability of the network.

$$f(x) = \begin{cases} x & if\ x > 0 \\ \alpha(e^x - 1) & if\ x \leq 0 \end{cases}$$

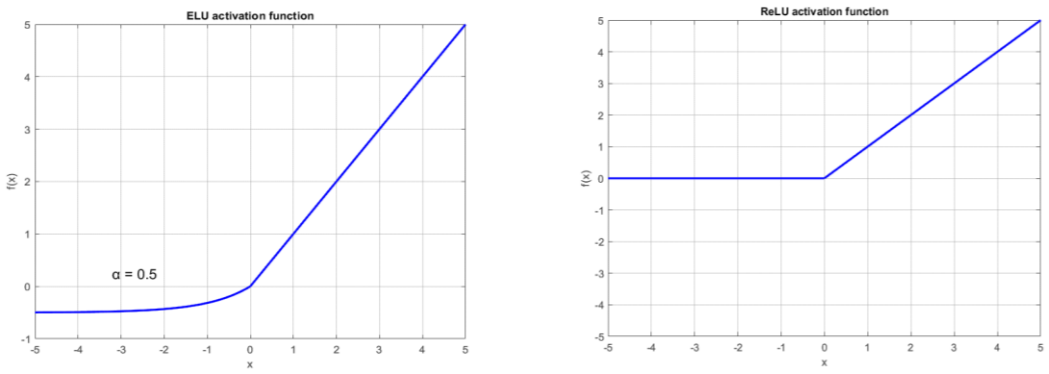


Figure 30: Plots showing the ELU and ReLU activation functions. ReLU does not permit negative activation values whereas ELU allows smaller negative activations.

3.3.3 Pooling

It is the downsampling layer applied after the activation function, with many options like max pooling, average pooling and L2-norm pooling. The max pooling layer partitions the input into non-overlapping blocks and outputs the maximum value in that block, whereas average pool outputs the average value from that block. Illustration of pooling operation is shown in Figure 31. However, is widely preferred because (1) it eliminates non-maximal values, thereby reducing computation in the layers; and (2) it provides translational invariance, because regardless of a pixel shift the max pooling layer is sensitive to the maximum value in that neighborhood.

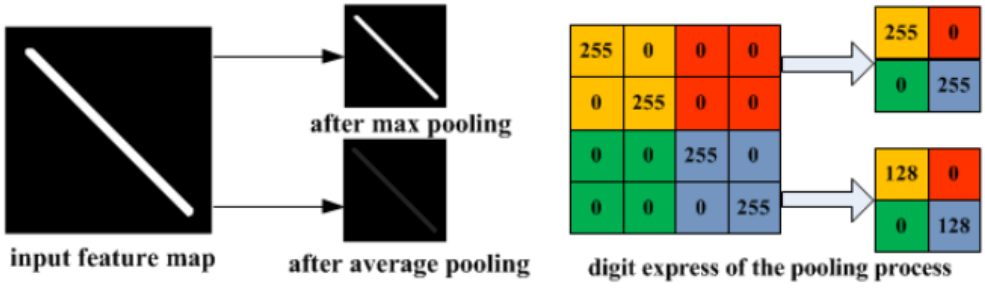


Figure 31: Comparison between max pooling and average pooling operations. Max pooling emphasizes on maximal value features, whereas average pooling de-emphasizes the maximal value features.

3.3.4 Additional Configurations and Layers

3.3.4.1 Residual layers/networks

Residual layers are building blocks of the ResNet architecture proposed by [46] . It was identified as a solution to the degradation problem in training deep neural networks, where stacking more layers does not improve the accuracy, rather counterintuitively degrades the accuracy after a state of saturation.

If a few stacked layers constitute a function H(x), then we can define another function called a residual function, F(x), which differs from H(x) by an identity mapping. Thus H(x) = F(x) + x and can be constructed by using a feedforward function involving an identity mapping (skip or shortcut connection).

Based on the hypothesis that "if a complicated function can be asymptotically approximated by multiple nonlinear functions then, these nonlinear functions can also asymptotically approximate its residual function", one can surmise that the residual function is easier to fit using nonlinear layers than the actual function H(x). This research makes it visible that

i) residual functions permit stacking of layers without adding any more parameters

ii) compared to the blocked or plain stacking of layers, the residual function method of stacking has lower training error for the same depth of the network.

This permits us to utilize the benefits of employing deep neural networks, which can learn highly discriminative features from complex data, while avoiding the degradation problem previously reported.

It can be seen in the further discussions that the proposed architectures utilize one or more residual layers because of the dimensionality of the data (spatial – spectral). For hyperspectral data without any factorization or dimensionality reduction to be used, deep layers would be required in order to extract discriminative features.

3.3.4.2 Batch Normalization

Batch Normalization or BatchNorm [47] was an important technique developed to increase the speed of training deep neural networks by reducing the internal covariate shift. While training the networks, the weights or parameters of the networks are constantly updated by error back propagation during each pass of the mini batch (in mini batch SGD optimization). Due to this the input distribution of the intermediate layers in the network keep varying, coercing the layers to learn from the varying distribution in the activation, thus increasing the time taken to train the network. If it is possible to make the inputs to the network layer all zero mean – unit variance, then the training can be accelerated, because of this stable gaussian input distribution.

While this is the conventional intuition provided for the success of BatchNorm in improving network performances, it is worth exploring the complicated effects of multiple layers in the networks whose output becomes the input to a subsequent layer. The higher order terms appearing in the Taylor's expansion of weight update around the current layer, can pose

problems during weight updates (requiring a very small step size or learning rate). By making the mean and variance of the activations to a layer independent of their values and the complicated interactions between layers, done by simplifying the learning dynamic using the learnable parameters γ and β, it can accelerate training and improve performance. The equations that describe the batch norm process are shown in Figure 32.

The most recent works on BatchNorm have proposed that its potential can be attributed to the smoothening effect on the loss optimization landscape [48].

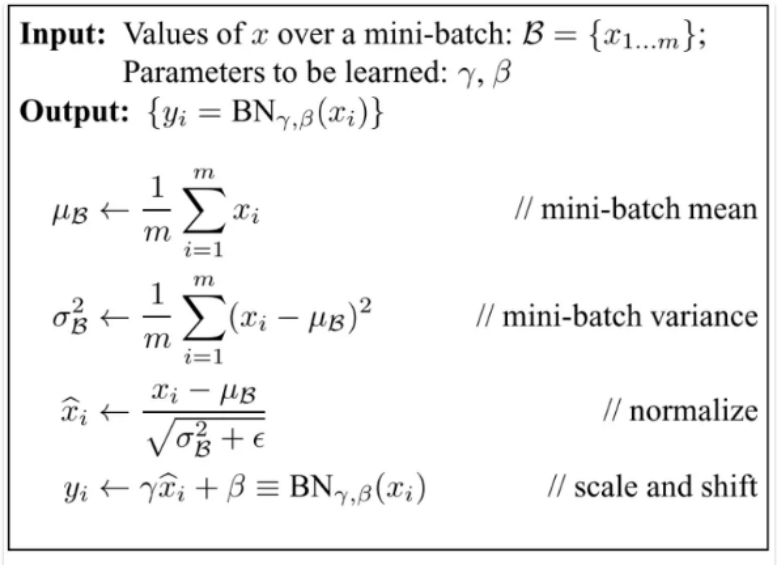


Figure 32: Equations that describe the BatchNorm layer as proposed in [47].

In the above equations, m refers to the number of mini batch samples; μ_B and σ_B^2 denote the mini-batch mean and variance respectively. As discussed earlier, the learnable parameters γ and β are used to scale and shift the activations.

3.3.5 Optimizers

Based on a chosen loss function, an optimizer or optimization algorithm can create a model of a given dataset. Gradient Descent optimization does so by minimizing a loss function in the negative direction of a gradient or slope of the error, leading towards a minimum error value. Hence it is called as Gradient Descent optimization.

Depending on after which subset of the dataset the model is updated, it can be classified into Stochastic Gradient Descent or SGD, mini-batch gradient descent or batch gradient descent.

SGD, also known as online learning algorithm, calculates the error for each sample in the dataset and updates the model accordingly. While this method of learning provides frequent updates to the model development, the learning can be noisy. As an advantage, noisy updates can help skip the local minima and prevent premature convergence of the parameters. However, this method is computationally intensive because of regular model updates and cannot be used for big datasets because of higher training time. The parameter update equation is:

$$\theta = \theta - \eta . \nabla_\theta J(\theta; x^{(i)}; y^{(i)})$$

where, θ is the model parameters, η is the learning rate, $\nabla_\theta J(\theta)$ is the gradient of the loss function $J(\theta)$, with $x^{(i)}$ and $y^{(i)}$ the individual samples and labels in the dataset.

In the batch or vanilla gradient descent, the error is computed for every sample in the dataset, but the update happens only after all the samples in the dataset have passed. The term 'epoch' refers to the training cycle of (forwards and backward pass) of the entire data set. Conversely, we can say that the network updates after each training epoch in such way that:

$$\theta = \theta - \eta . \nabla_\theta J(\theta)$$

where, $\nabla_\theta J(\theta)$ denotes the gradient of the loss function, considered over the entire data set.

Another variant of this is the mini-batch gradient descent, which is commonly used in deep learning applications. In this, the dataset is split into smaller 'mini-batches' and the error is computed, and the model parameters are updated for each pass of these mini batches. This method is more efficient than SGD, because the computational burden is reduced and the memory requirement for holding an entire dataset (as in batch gradient descent) is eased. The update rate is higher than the batch gradient descent but lower than SGD, which provides it a balance in terms of speed of convergence and computational efficiency. By splitting the data set into a number of mini-batches, denoted as 'batch size', the update occurs batch size times during an epoch and is defined as follows:

$$\theta = \theta - \eta . \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)})$$

where, $x^{(i:i+n)}; y^{(i:i+n)}$ denote the number of samples and labels considered during training according to the chosen mini batch size.

Another relevant concept is the learning rate (a multiplier to the gradients), which defines how quickly the gradients are updated. Choice of a larger learning rate like 0.1 may lead to bigger jumps in the gradient descent process and skip an existing (local) minimum, thus never being able to converge to a stable minimum. On the other hand, a smaller learning rate like 0.0001

causes shorter jumps, and updates the gradients much slower, leading to entrapment in a local minimum.

For SGD, there are landscape features like ravines, commonly present around local minima. This slows down the SGD in reaching the local minima and it starts oscillating along the ravine slopes. In order to accelerate the convergence and reduce the effect of oscillation a technique called momentum can be used. This can be used in the following update equation to provide a fraction (γ) of the previous update term to the current one:

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

$$\theta_{t+1} = \theta_t - v_t$$

3.3.5.1 Adaptive learning optimizers

This class of optimizers has adaptive learning rate, (i.e) different learning rate for each parameter at every time step *t*, making $g_{t,i}$ the gradient of the parameter θ_i at the time step *t*.

RMSprop

RMSprop [49] is an adaptive learning optimizer which has an update rule based on the exponentially decaying average of squared gradients (g^2) at time step *t*. In this the learning rate η is divided by the exponentially decaying average of g^2:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

ADAM

ADAM or Adaptive Moment Estimation [49] is similar to RMSprop in that it stores the exponentially decaying average of past gradients along with that of the past squared gradients. The former is the estimate of the first moment ($\hat{m}_t$) and the latter is the estimate of the second moment ($\hat{v}_t$). The update equation is:

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

3.3.6 Performance Metrics

3.3.6.1 Recall

If T_p is number of true positives and F_n is number of false negatives, then the metric recall is defined as:

$$R = \frac{T_p}{T_p + F_n}$$

Recall is also called as sensitivity.

3.3.6.2 Precision

If T_p is number of true positives and F_p is number of false positives, then the metric precision is defined as:

$$P = \frac{T_p}{T_p + F_p}$$

Precision can also be referred to a positive predictive value. It can be seen for our case, that the precision indicates how precise the deep learning model is in classifying a voxel/ spectrum as one of the four categories. In other words, of the voxels predicted as positive, how many are actually positive.

3.3.6.3 F-1 score

The F-1 score or Intersection over Union (IoU) or Dice coefficient is defined as the harmonic mean of recall and precision, denoted as:

$$F1 = 2 \times \frac{P \times R}{P + R}$$

From the definitions of recall and precision, we can observe trade-off relationship existing between them. By calculating their harmonic mean, we can provide a balance to these metrics.

3.3.7 Loss Functions

3.3.7.1 Softmax activation

It is utilized at the output layer of a network to squash an output vector in the range (0,1). This means the output probability range is in the range (0,1) and adds up to 1. This function calculates the probabilities of each target class over all possible target classes (C). If s is the output score (vector) of the network, then $f(s)_i$ is the function for each individual element of the vector. The softmax function is represented as:

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}}$$

Where, s_i is the element in the score corresponding to each class, and in the summation, s_j is the function score for each class in C. It can be seen, that the softmax function activations depend on all the elements of s.

3.3.7.2 Categorical cross-entropy loss

This is also called softmax loss (softmax activation + cross-entropy loss). The cross-entropy loss is defined as:

$$CE = -\sum_{i}^{C} t_i log(s_i)$$

where, t_i denotes the ground truth and s_i the output score for each class i in C. In the case of multi-class classification, the ground truth labels are one-hot encoded. Therefore, only one element in the ground truth is non-zero (s_{OH}) which eliminates the remaining terms in the summation. The categorical cross-entropy loss is hence defined as:

$$CCE = -log\left(\frac{e^{s_{OH}}}{\sum_j^C e^{s_j}}\right)$$

In case of multi-label classification, the above equation can be modified to include M positive classes of the sample, defined as follows:

$$CCE = -\frac{1}{M}\sum_{p}^{M} log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)$$

where, p is the positive class and s_p is the score corresponding to each positive class and $\frac{1}{M}$ is the scaling factor for invariance to number of positive classes. Similarly, an equation to represent the negative classes can also be obtained.

3.3.7.3 Dice coefficient loss

Dice coefficient or F-1 score is a metric that determines the overlap in segmentation, to evaluate the segmentation performance based on a ground truth label, especially to counter class imbalance in the data. While binary segmentation problems(foreground vs background) are common, in order to implement multi-class segmentation, a generalized dice coefficient loss formulated as shown below:

$$Dice\ loss = 1 - 2\frac{\sum_{l=1}^{2} w_l \sum_n t_{ln}\, p_{ln}}{\sum_{l=1}^{2} w_l \sum_n t_{ln} + p_{ln}}$$

where, l determines if it is binary or multi-class, t_{ln} and p_{ln} refer to the ground truth labels and predicted probability map respectively; w_l is a weighting term used to make the Dice score invariant to the label region size, which can be performed by dividing the contribution from each label by its volume.

3.4 Relevant Architectures

3.4.1 U-Net

The U-Net is an adaptation of the Fully Convolutional Network (FCN), which can be specifically used in the case of segmentation of biomedical images, where very little training data is usually available [50]. This architecture, illustrated in Figure 33, relies on data augmentation in the upsampling part of the network at the feature map level, thus increasing the context available to the higher resolution layers. It consists of a symmetrical pathway from the input layer to the output, with the downsampling (or contractive) path on the left and upsampling (or expanding) path on the right of the network, thus giving it a U-shaped appearance. By creating skipped connections between the contractive path and the expanding path and, concatenating the low-level and high-level feature maps from the two paths and applying convolutions and nonlinearities at each upsampling step, the so called long-skipped connections are created. They have been found useful in recovering the full spatial resolution at the output layer [51]. This can also be vital for the accurate class localization at the output, which is the prediction of the spatial location of a particular class [50].

The U-Net architecture is well known in medical applications, since it learns the whole context from entire scans/images and produces a segmentation map of the tissue/organ under consideration. This certainly gives U-Net an advantage when compared to patch-based segmentations. An extension of this architecture to 3-D medical images, used a few 2-D annotated slices (sparse annotations) to generate 3-D volumetric segmentations [52]. Two different architectures based on U-Net include the V-Net, which utilizes 3-D convolutional layers and a loss function based on Dice coefficient [53], and the FusionNet, which makes use of long skip connections in the form of residual layers, along with the U-Net skip connections to create a deep architecture for automatic Electron Microscopy segmentation [54].

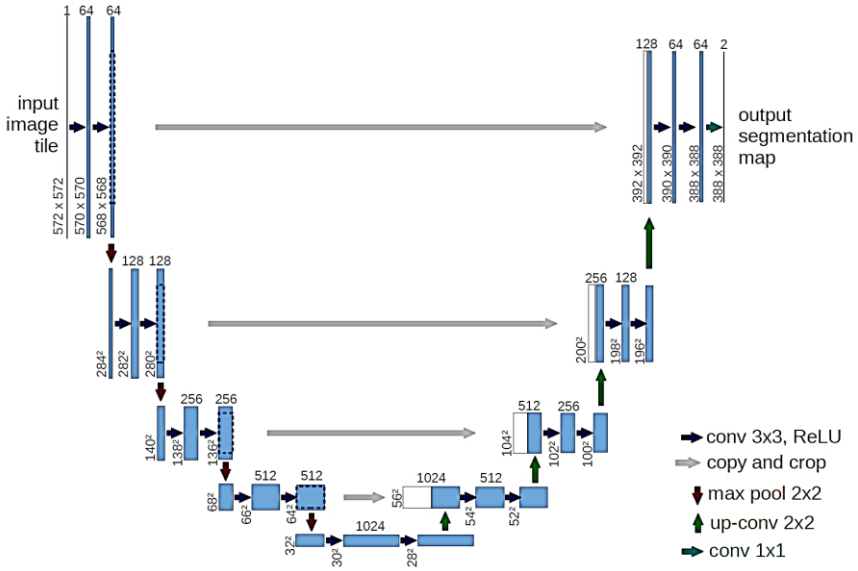


Figure 33: U-Net architecture proposed in [50]. It consists of a contractive path on the left following by an expansive path in the right, with feature concatenation occurring between layers of corresponding feature dimensions on both sides.

3.4.2 Spectral – Spatial Residual Networks

This architecture [10] tries to exploit the 3-D structure (one spectral and two spatial dimensions) of a hyperspectral data cube by consecutive learning of spectral features first and then the spatial features. In the literature review section of this report, it has been briefly mentioned, along with its potential use in the remote sensing domain. It consists of a spectral channel, connected in series with a spatial channel of feature learning. The crucial component of this architecture is the (multiple) residual block(s) in each learning channel, which will be discussed at length at the architecture components part later.

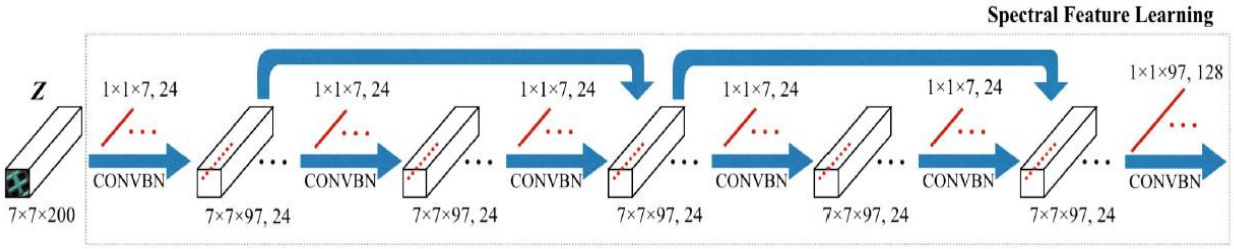


Figure 34: The spectral feature learning channel of the spectral-spatial residual network architecture [10].

This architecture classifies small hyperspectral data volumes of the dimension 7 x 7 x 200, into one of the many landcover classes from the Kennedy Space Center and University of Pavia data sets (appendix). In the spectral channel, the spectral dimension is downsized to 97 using 3-D convolution filter of 1 x 1 x 7. In the subsequent two residual blocks, the dimensions of the volume are preserved by using zero padding along the spectral dimension. It can be seen, there are 24 convolutional filters or kernels that are defined in each stage of the channel. However, at the final stage the 7 x 7 x 97 volume is converted to a 7 x 7 x 128 volume by convolving with 128 kernels of the dimension 1 x 1 x 97. By concatenating all these spectral feature kernels, the final volume is obtained, to be made the input for the spatial learning channel.

In the spatial channel, a structure similar to the spectral learning channel is adopted with different convolution filter sizes of 3 x 3 x 128. This means, in the input 7 x 7 spatial area, a 3 x 3 convolution filter is applied across all the 128 spectral channels. This is followed by residual blocks and an average pooling layer, and finally a fully connected layer that outputs a 1 x 1 x *L* vector, which could denote on of the *L* landcover categories.

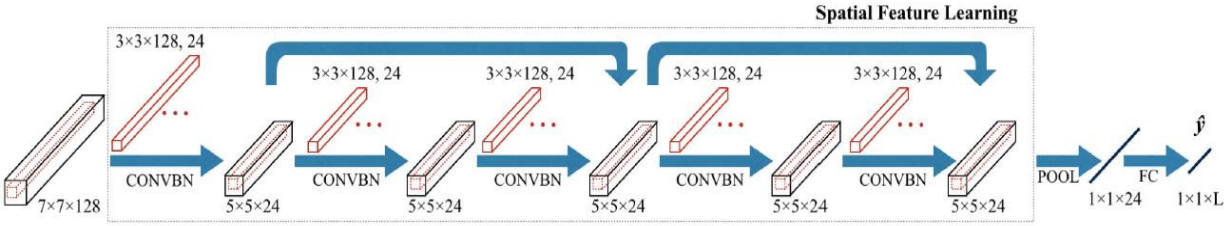


Figure 35:The spatial feature learning channel of the spectral-spatial residual network architecture [10].

3.4.3 Simultaneous 3-D convolution

While the previous approach followed a consecutive spectral and spatial approach for feature learning of hyperspectral data cubes, the simultaneous method of spectral-spatial feature learning [16] can be performed by using 3-D convolutional kernels. In this architecture (shown in Figure 36), there are two 3-D convolutional layers and their output is flattened into a feature

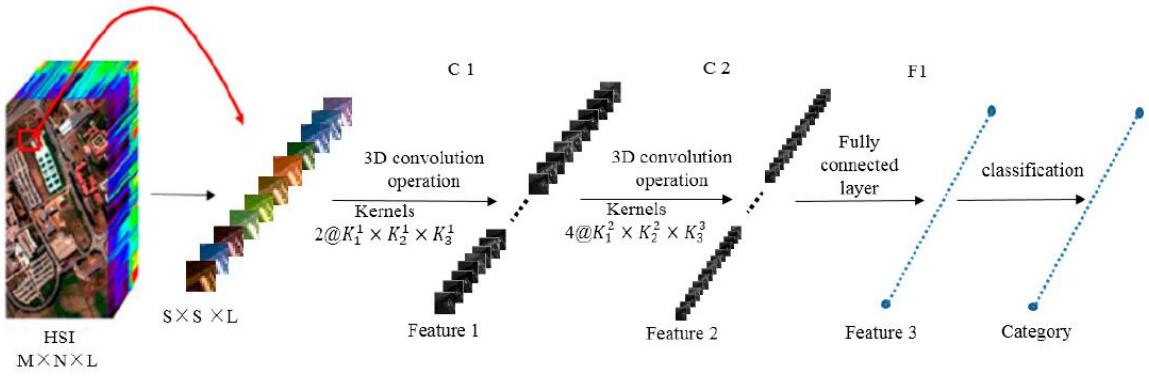


Figure 36: The simultaneous spectral-spatial learning network based on 3-D convolution layers [16]. It utilizes 3-D convolutional kernel to simultaneously learn spatial and spectral features.

vector before being fed to a fully connected layer with a softmax activation to classify the input volumes into only of the multiple landcover categories, obtained from the Pavia University Scene, Botswana Scene and Indian Pines Scene (refer appendix) datasets. Between the first and second 3-D convolutional layers, the number of kernels/ filters is in the 1:2 ratio. Since a hyperspectral data cube has rich information in the form of spatial – spectral correlations, it may be necessary to preserve those correlations by simultaneously learning the joint spectral – spatial features. In comparison, a 2-D convolution-based configuration has no convolution occurring along the spectral dimension, thus not able to preserve the useful spectral information. This architecture also avoids any pooling operation in order to eliminate further reduction of the spatial resolution. The simultaneous 3-D convolution operation also eliminates the need for any dimensionality reduction (like PCA, NMF) required along the spectral dimension, which would be necessary were 2-D convolution layers to be used. This is because each 2-D convolution applied to the spatial dimension of a hyperspectral data cube would create multiple kernels/ filters for each 2-D channel. Combining with about 100 – 200 channels, a huge number of learnable kernels or parameters would arise, leading to a highly overfitting network. In contrast, a 3-D convolutional network would possess fewer parameters to train and hence lower computation costs.

3.5 Frameworks & processing capability

For experimenting with deep learning, a terminal with an access to Philips Linux-based compute cluster was utilized. While this was adequate for debugging and small computations like image preprocessing and visualization, a separate batch server with Nvidia Tesla K80 GPUs proved to be useful for training the network and predictions. For creating the networks and training them, frameworks like Caffe2 and TensorFlow were initially considered. TensorFlow was preferred for its multi-language support including Python, which was utilized for programming a major part of this project. Also, it has emerged as the industry standard for deep learning development, with a vast number of repositories and documentations. With a high-level wrapper in Keras and TensorFlow backend, the experiments and prototyping can be performed seamlessly. Keras also receives regular updates with respect to the latest developments happening through deep learning research (example: advanced activations, convolution layers etc.)

"What are the possible approaches in learning features from hyperspectral images?"

"What design choices were made corresponding to the feature learning approaches?"

3.6 Proposed approaches

3.6.1 Patch classification vs pixel-wise classification

In all the HSI research on landcover classification, a small spatial patch of size 7 x 7 or higher is considered as the input to the network, which usually classifies the patch into one of the multiple landcover classes. While this is cogent for the scale involved in the landcover or aerial images of a geographical area, a small spatial patch like 7 x 7 is still considerably smaller in the geographical scale and for convenience, can represent a basic spatial unit that constitutes the image. However, this need not be replicated for a medical image, since for a hyperspectral image of 224 x 224 spatial dimension, a 7 x 7 spatial neighborhood becomes a significant area. Given that the labels are derived from hand drawn pathological slides, there is a level of uncertainty in the borders between the tissue categories, for instance between tumor and muscle. We have in fact modelled this uncertainty in annotation, as a separate tissue class 'unknown'. Any additional noise introduced into the labels during (local) image registration process could also affect the quality of labelling. Also, a patch-based classification assumes that each patch belongs to a unitary category. Crucial regions (like tumor) that do not fit into this spatial area, might be rejected. Given that, we have limited patient samples, it may be prudent to entirely utilize the regions of tissue. Designing this problem as pixel-wise segmentation also permits a spatial area to consist of more than one tissue class. By explicitly retaining this spatial correlation, we can facilitate better discrimination between the tumor and

muscle tissue classes. By line of this reasoning, it is decided to adapt a pixelwise classification approach, initially with inputs of spatial size 16 x 16.

3.6.1 Spectral approach

In this approach, from each of the 31 sub-cropped hyperspectral image blocks of dimension 224 x 224 x 164, input volumes corresponding to a 16 x 16 spatial neighborhood are extracted in a non-overlapping fashion. Each input volume is of the dimension 16 x 16 x 164 and the spectral information is kept intact during this procedure. The reasoning for the choice of a smaller spatial neighborhood is that by associating an individual spectrum with its corresponding spatial neighborhood spectra (in this case 16 x 16), we can provide spatial correlation to that spectrum. The previously explored method does not take this correlation into consideration, because the network is trained on each individual spectrum in each hyperspectral sub-crop of 224 x 224 spatial size. By providing this form of spatial correlation, we can segue into answering if spatial information is indeed essential to perform pixel-wise segmentation of hyperspectral images.

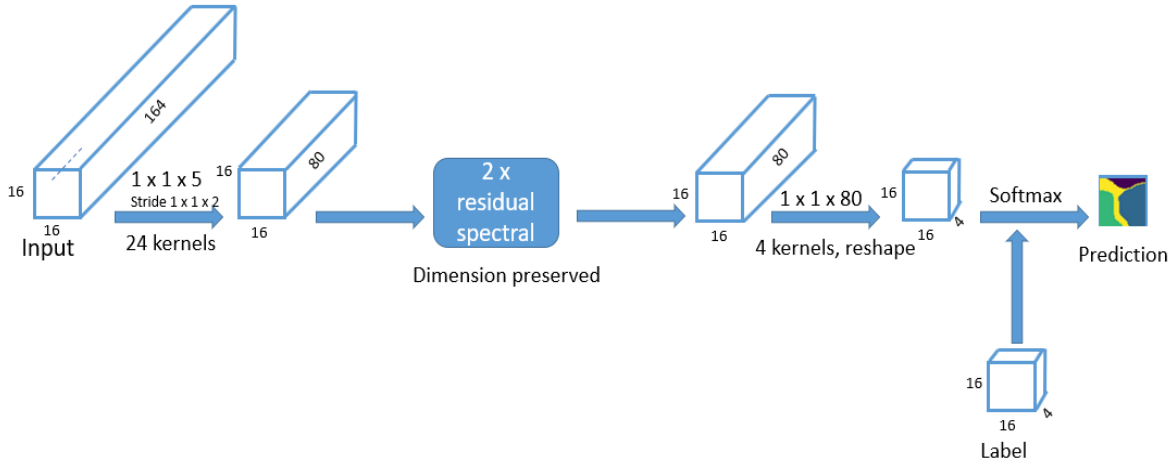


Figure 37: The proposed architecture for the spectral feature learning approach. It utilizes 1-D kernels of the form 1 x 1 x N, followed by two residual spectral layers for deep feature learning.

An illustration of the architecture is shown in Figure 37. In this configuration we have the 16 x 16 x 164 input volume provided to the input layer, after which it is passed to the 3-D convolution layer, where 3-D convolution kernels of size 1 x 1 x 5 are defined, which stride along the input volume at 2 units along the spectral direction. After this, two 'residual spectral' blocks, which are basically two residual blocks stacked in series, process the output from the previous layer. A spectral residual block is illustrated in Figure 38.

It comprises twice- stacked 3-D convolution layers with 1 x 1 x 3 kernels, ELU activation and a batchnorm layer which are then added to a shortcut connection, finishing with another ELU activation and a spatial dropout layer. In these spectral residual blocks, zero padding is done to preserve the dimension of the data volumes, wherever necessary. Since the problem is a pixel-

wise segmentation, we configure the network to be Fully Convolutional, thus avoiding any flattening or fully connected layer. Therefore, the next layer is a 3-D convolution which diminishes the spectral dimension using convolving with 1 x 1 x 80 kernels, producing a 16 x 16 spatial output. By convolving them once more by using 4 kernels and concatenating them we obtain the 16 x 16 x 4 final volume (comparable with the categorical label of 16 x 16 x 4 size), which will then be passed through a softmax activation layer that can produce a probability map of the 16 x 16 spatial area. By utilizing these residual layers, we can stack six 3-D convolutional layers in the network.

3.6.1.1 Training set up

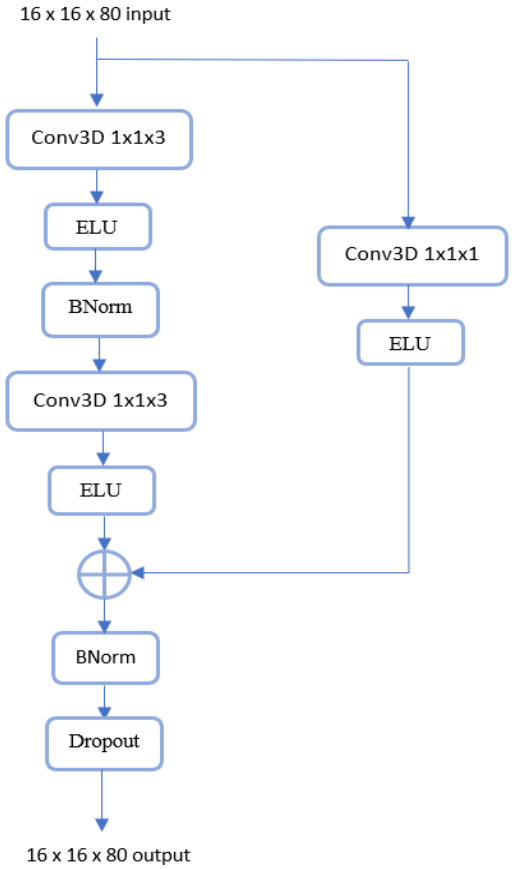


Figure 38: Schematic showing the constituent layers of the spectral residual block in the proposed architecture. It follows the order conv-ELU-BNorm, with a final BNorm and Dropout layer after identity summation.

The data obtained by converting the sub-cropped hyperspectral volumes into 16 x 16 x 164 are used in this configuration. This is further split into training data (80%) and validation data (20%) for identifying overfitting during the training process. In accordance with TensorFlow data shape requirements, the N inputs of size 16 x 16 x 164 are reshaped into (N x 16 x 16 x 164 x 1) to be introduced into the input layer.

Again, a leave-one-out scheme of testing new patient data is performed. Thus, after training it on 6 patients' data in the form of 16 x 16 x 164 input, we test it on the one remaining patient data of the same shape. For performing this prediction, we require the model weights to be stored after training. Sometimes, it might be required to save the entire model if one wants to

retrain the network or fine tune the network, because the latest optimizer states are required to continue training an already trained network.

The network is trained by optimizing the categorical cross-entropy loss. It is trained for 100 epochs using a minibatch size of 80. The preferred optimizer in this case is Adam at a learning rate of 1e-3. The following (Table 7) showcases all the hyperparameters that were determined for this configuration. The graphs corresponding to the model accuracy vs number of epochs and model loss vs number of epochs during model training are shown in Figure 39.

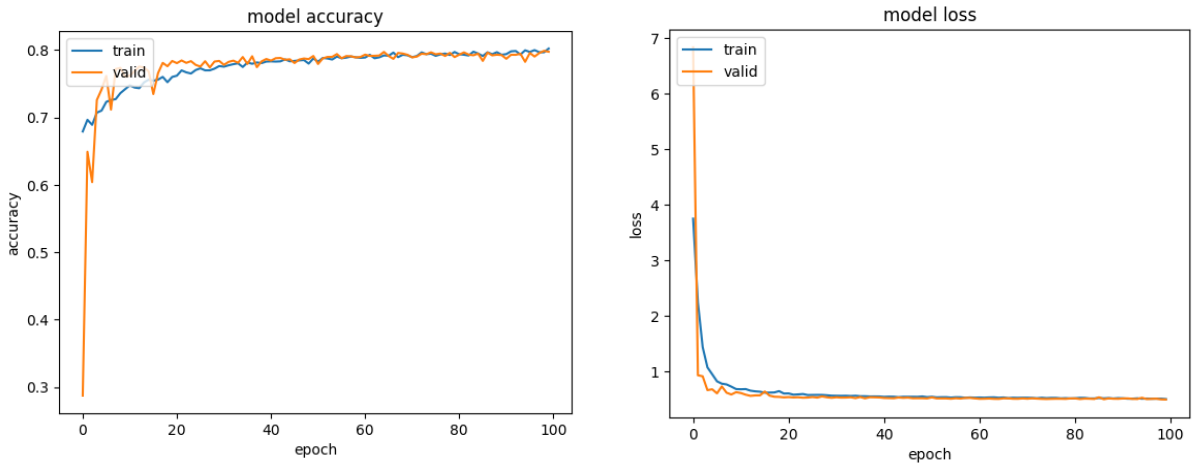


Figure 39: Plots showing the model accuracy and loss values for 100 epochs during model training and validation for the first experiment.

"How can the set of hyperparameters for a given experiment be determined?"

The choice of the number of residual spectral layers, kernel size, minibatch size, learning rate were determined using a grid-search algorithm from a grid of kernel sizes [1x1x5, 1x1x7, 1x1x9], learning rates [0.0001, 0.005, 0.001, 0.05], minibatch sizes [32, 64, 80, 96], optimizers [Adam, RMSprop], Activations [ELU, ReLU] and number of residual blocks [1,2,3,4]

Table 7: Different hyperparameters determined during training of first experiment.

Hyperparameter	Value/ choice
During training	
Epochs	100
Mini batch size	80
Learning rate	0.001
Network structure	
Optimizer	Adam
Number of residual blocks	2
Activation	ELU

Table 8: The elapsed time during model training and testing process in the first experiment.

Stage	Time for completion (hours)
Model training	4
Model testing / prediction	0.025

3.6.1.2 Prediction

Given that the training input data cubes 16 x 16 x 164 are obtained from the sub-cropped volumes of size 224 x 224 x 164, in order to visualize the probability map or segmentation of the leave-one-out testing scheme, we crop the remaining one original patient data cube into non-overlapping blocks of size 224 x 224 x 164 and further into 16 x 16 x 164. In contrast, the training data is obtained by manually cropping relevant sub-cropped blocks, though of the same volume. Thus, to obtain the segmentation of the complete testing data (remaining one data cube), it is necessary to stitch the 16 x 16 non-overlapping regions together into the 224 x 224 sub-crop. By stitching back all the sub-crops corresponding to the whole testing data, we get its pixel-wise classification.

3.6.2 Spectral-Spatial Approach

While the previous approach did include the spatial correlation in the small data volumes (16 x 16 x 164) by considering the spatial neighborhood of a spectrum, it explicitly learned only the spectral information. No convolution operation (or striding) was performed on the spatial region, therefore a new approach can be proposed to include both the spectral and spatial information simultaneously during convolution. Similar to the previous architecture discussed, this too can be realized using the 3-D convolution layer available in TensorFlow. The difference in the former however, is that the kernels by definition are 3-D, but the dimensions are set to 1 x 1 x *N*, making them behave like 1-D convolution. For the spectral-spatial architecture, the 3-D kernels are defined in the spatial dimension also, indicated by the form *M* x *M* x *N*. By using such a 3-D kernel, we can concurrently perform convolution on the spatial plane and the spectral plane of the 3-D hypercube. This approach can compactly learn spectral-spatial features from the hypercube volume, without having to lose spatial information by convolving along the spectral plane or lose the spectral information by only convolving along the spatial plane (not to forget the linearly increasing number of parameters to learn).

The architecture developed for this spectral-spatial approach is illustrated in Figure 40. In this case, we use the 31 sub-cropped data cubes of size 224 x 224 x 164 from the original 7 patient data, without narrowing the spatial size down to a smaller neighborhood like in the previous approach. This choice of spatial dimension stems from the notion that the distinctive spatial features of a tissue category like tumor (or muscle) are appreciable in size to the spatial resolution of 224 x 224 and by preserving this, we can provide better distinction with the remaining categories. When we analyze the spectra of the hyperspectral data, we can observe that the spectral signatures of tumor and muscle tissue for a few samples are dissimilar (as they should be theoretically), the remaining samples have similar spectral signatures for tumor and muscle [refer Chapter II]. By providing spatial context to these spectra, we can explicitly form an inter-class distinction. However, it is important to bear in mind that this size of 3-D data cube can place restrictions on choice of number of kernels, intermediate volumes (through convolution, volume concatenation) and network depth. While working with TensorFlow with the GPU, the tensors created can often consume a major portion of the memory. It is thus important to design an architecture which circumvents these limitations and some of the design choices are discussed further.

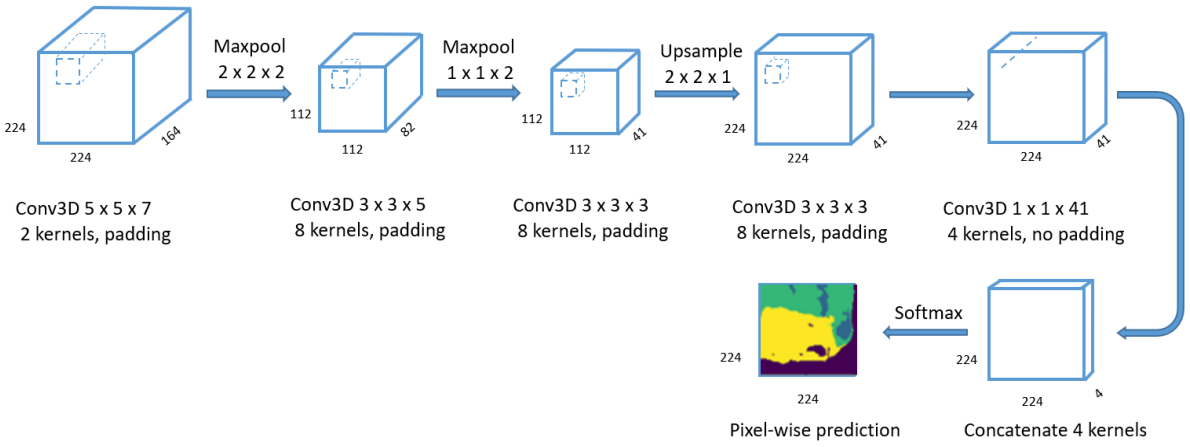


Figure 40: The proposed architecture for the simultaneous spectral - spatial feature learning approach. Hyperspectral images of dimension 224 x 224 x 164 are provided as the input.

In the first layer, 3-D convolution is performed by using two kernels of size 5 x 5 x 7 and including padding. This non-standard kernel can help capture the global features along both the spatial and spectral dimensions faster. This can be useful considering the noisy nature of the spectral signals (separate strategies to overcome this problem are described at a later stage). In order to work around the memory restrictions, a max-pooling layer is used to downsize the 3-D volume from the previous layer. By using the *MaxPooling3D* operation, we can downsize the volume to 112 x 112 x 82. On this smaller volume we can now perform 3 x 3 x 5 convolution by increasing the number of kernels to 8. By only downsizing the volume along the spectral dimension and preserving the spatial dimension we can prevent the further loss of spatial information. Thus, the subsequent 3-D max-pool layer creates a data volume of 112 x 112 x 41, upon which a standard 3 x 3 x 3 convolution operation using 8 kernels is performed. For this pixel-wise segmentation task, we must restore the output to the original spatial resolution, which can be done by a *UpSampling3D* layer only on the spatial dimension leading to a 224 x

224 x 41 volume. To match this with an associated label image size, we perform a 1 x 1 x 41 convolution with 4 kernels, which can then be concatenated to create an output volume of dimension 224 x 224 x 4. After this, a *softmax* classifier creates a probability map corresponding to the output from the previous layer.

3.6.2.1 Training set up

The 31 sub-cropped data cubes of size 224 x 224 x 164 corresponding to the 7 patient samples, form the data set. This is further split into training data (80%) and validation data (20%) for identifying overfitting during the training process. In accordance with TensorFlow data shape requirements, the N_b inputs of size 224 x 224 x 164 are reshaped into (N_b x 224 x 224 x 164 x 1)* to be introduced into the input layer.

Again, a leave-one-out scheme of testing new patient data is performed. Thus, after training it on 6 patients' data in the form of 224 x 224 x 164 input, we test it on the one remaining patient data of the same shape. For performing this prediction, we require the model weights to be stored after training. If it is required to retrain the network or fine tune the network, we may choose to save the whole model because the latest optimizer states are required to continue training an already trained network.

The network is trained by optimizing the Dice coefficient loss, described previously. It is trained for 100 epochs using a minibatch size of 4. The preferred optimizer in this case is Adam at a learning rate of 5e-3. The following Table 9 showcases all the hyperparameters that were determined for this configuration. The graphs corresponding to the model accuracy vs number of epochs and model loss vs number of epochs during model training are shown in Figure 41.

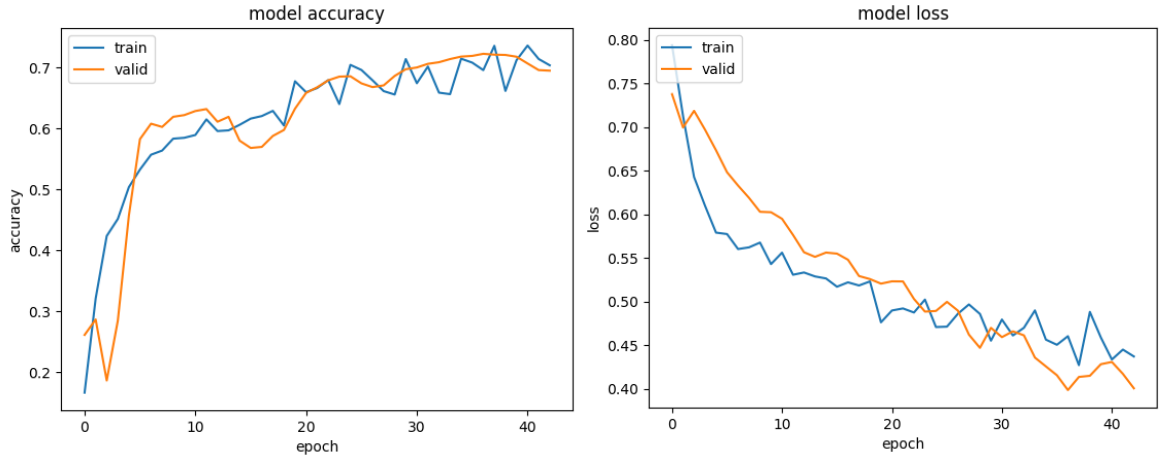


Figure 41: Plots showing the model accuracy and loss value for 50 epochs during model training and validation for the second experiment. Early stopping is applied to prevent overfitting in this case.

The choice of the number of residual spectral layers, kernel size, minibatch size, learning rate were similarly determined using a grid-search algorithm. The choice of the number of kernel

size, minibatch size, optimizers and learning rate were determined using a grid-search algorithm from a grid of kernel sizes [5x5x7, 3x3x5, 7x7x9], learning rates [0.0001, 0.005, 0.001, 0.05], minibatch sizes [3, 4, 5, 6] and optimizers [Adam, RMSprop].

Table 9: Different hyperparameters determined during training of second experiment

Hyperparameter	Value/ choice
During training	
Epochs	50
Mini batch size	3
Learning rate	0.005
Network structure	
Optimizer	Adam
Activation	ELU

Table 10: The elapsed time during model training and testing process in the second experiment

Stage	Time for completion (hours)
Model training	3
Model testing / prediction	0.016

3.6.2.2 Prediction

In the first leave-one-out training and testing scheme, the first patient sample data cube (#1) was treated as the testing data cube, in the second, the #2 was treated as the testing data cube and so on. In this manner, 7 testing schemes were instituted after which their corresponding pixel-wise segmentation images were generated. From Table 6, it can be seen all schemes did not have the same number of sub-cropped regions or blocks due to the varying size of the tissue in the region of interest.

The pixel-wise segmentation output for a leave-one-out testing scheme is obtained by stitching all the segmented 224 x 224 (non-overlapping) sub-crops together. This is easier than the stitching procedure in the previous architecture. This probability map can be converted into the final segmentation by using the arg-max operation across the four channels corresponding to the four tissue categories.

3.6.3 Data augmentation in spectral-spatial method with 224 x 224 spatial dimension

While working with limited patient data and fewer training sub-cropped data cubes (example 31) and a simpler network without concatenating features (like U-Net), it may be necessary to formulate a data augmentation solution to try to improve performance. Therefore, transformations like rotation and flipping can be utilized to augment more data to the original training data. By using 90° rotation and horizontal flipping of the data cubes, the number of sub-cropped data can be tripled. Various observations from this experiment are recorded in the next section. However, increasing the data more than three times (vertical flipping, deformation) did not improve the performance any further or introduced some undesirable effects. This could be due to simply multiplying the number of data cubes which are themselves correlated across different channels.

This experiment is similar to the previous method, except with increase number of training samples (thrice the previous). The hyperparameters and the training set up are exactly the same as the previous method.

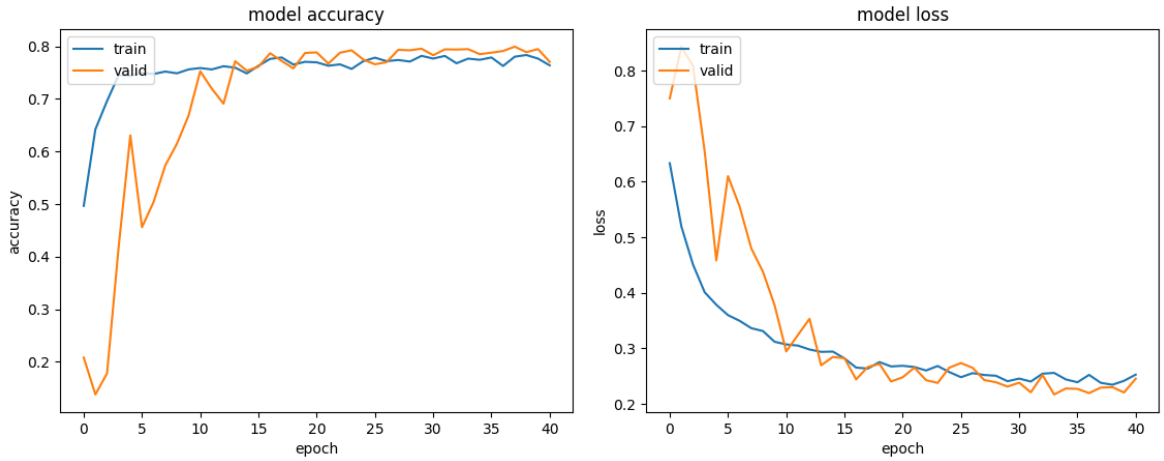


Figure 42: Plots showing the model accuracy and loss value for 50 epochs during model training and validation for the third experiment. Early stopping condition is applied to prevent overfitting in this case.

Table 11: The elapsed time during model training and testing process in the third experiment

Stage	Time for completion
Model training	4
Model testing / prediction	0.016

3.6.4 Spectral-spatial method with spatial dimension 112 x 112 (data augmented)

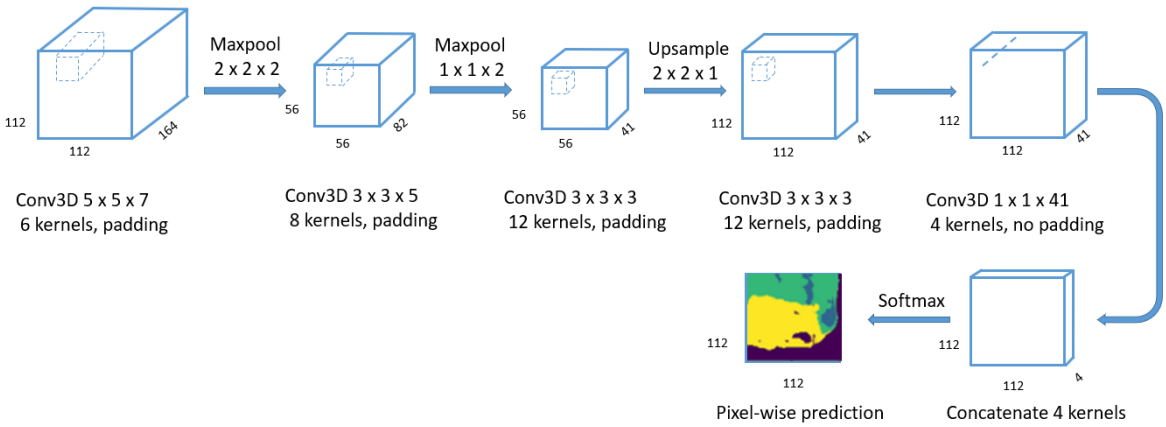


Figure 43: The proposed architecture for the simultaneous spectral - spatial feature learning approach. Hyperspectral images of dimension 112 x 112 x 164 are provided as the input.

In the final approach, the same spectral-spatial combined feature learning network is implemented on training data of dimension 112 x 112 x 164. This method is motivated due to the constraints in GPU memory in creating tensors for performing operations on inputs of the dimension 224 x 224 x 164. This places a restriction on the choice of number of learnable kernels, configuration of the architecture (feature concatenation), depth of the network and minibatch size. By working with smaller input hyperspectral image dimensions, an effort to alleviate the problems can be put in place. From each 224 x 224 spatial region, four 112 x 112 spatial regions can be extracted. Four sub-blocks from each 224 x 224 x 164 data cube can be used for training data, thus creating four times as many training data samples as the previous approach. Further, it was decided to augment additional data by applying geometrical transformations to increase the data samples by an additional three times (90° rotation and horizontal flipping).

Further experiments include, using data cubes of dimension 112 x 112 x 164 without data augmentation, using an intermediate spatial sized data cube with dimension 160 x 160 x 164, architectures with skip connections (residual layer, U-Net feature concatenation). These experiments failed to provide optimal results in terms of lower segmentation performance, convergence problems of the parameters or even memory constraints in case of the 160 x 160 x 164 input and they will not be discussed hence forth.

As can be seen from the illustration in Figure 43, this method utilizes a similar architecture from the previous two methods. However, due to the reduced spatial dimension, there is leniency in the choice of number of kernels. The number of 3-D kernels is increased to 6, 8 and 12, from 2, 8 and 8 in the original architecture. The depth of the network is still the same so as to not lose the spatial features further through max-pooling, while trying to keep the number

of parameters in the subsequent layers under check. The 3-D kernels are of the same dimension and the following hyperparameters have been determined as in Table 12.

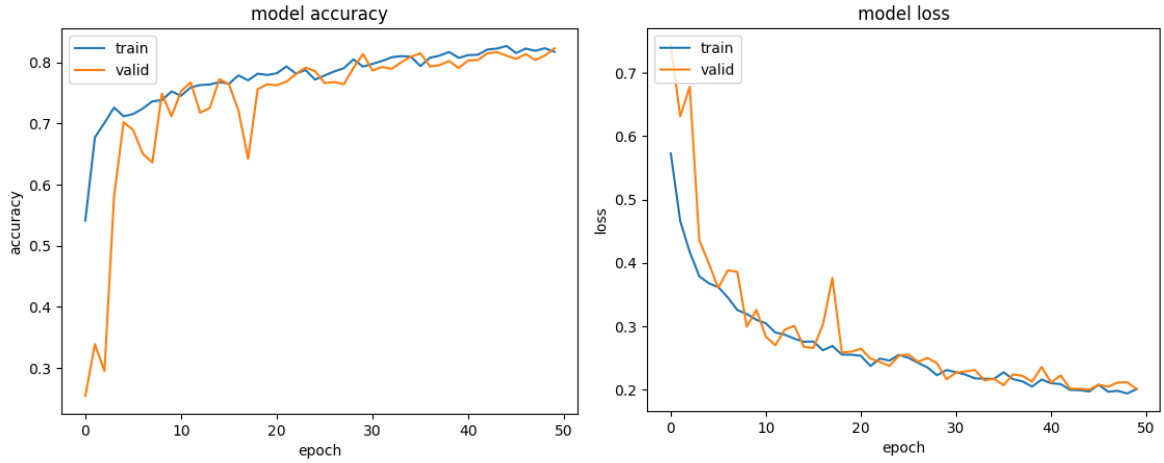


Figure 44: Plots showing the model accuracy and loss value for 50 epochs during model training and validation for the fourth experiment. Early stopping is applied to prevent overfitting in this case.

Table 12: Different hyperparameters determined during training of the fourth experiment

Hyperparameter	Value/ choice
During training	
Epochs	50
Mini batch size	12
Learning rate	0.005
Network structure	
Optimizer	Adam
Activation	ELU

Table 13: The elapsed time during model training and testing process in the fourth experiment

Stage	Time for completion (hours)
Model training	4.5
Model testing / prediction	0.025

Chapter IV – Experimental Results

"How do the performance metrics compare for the considered experiments?"

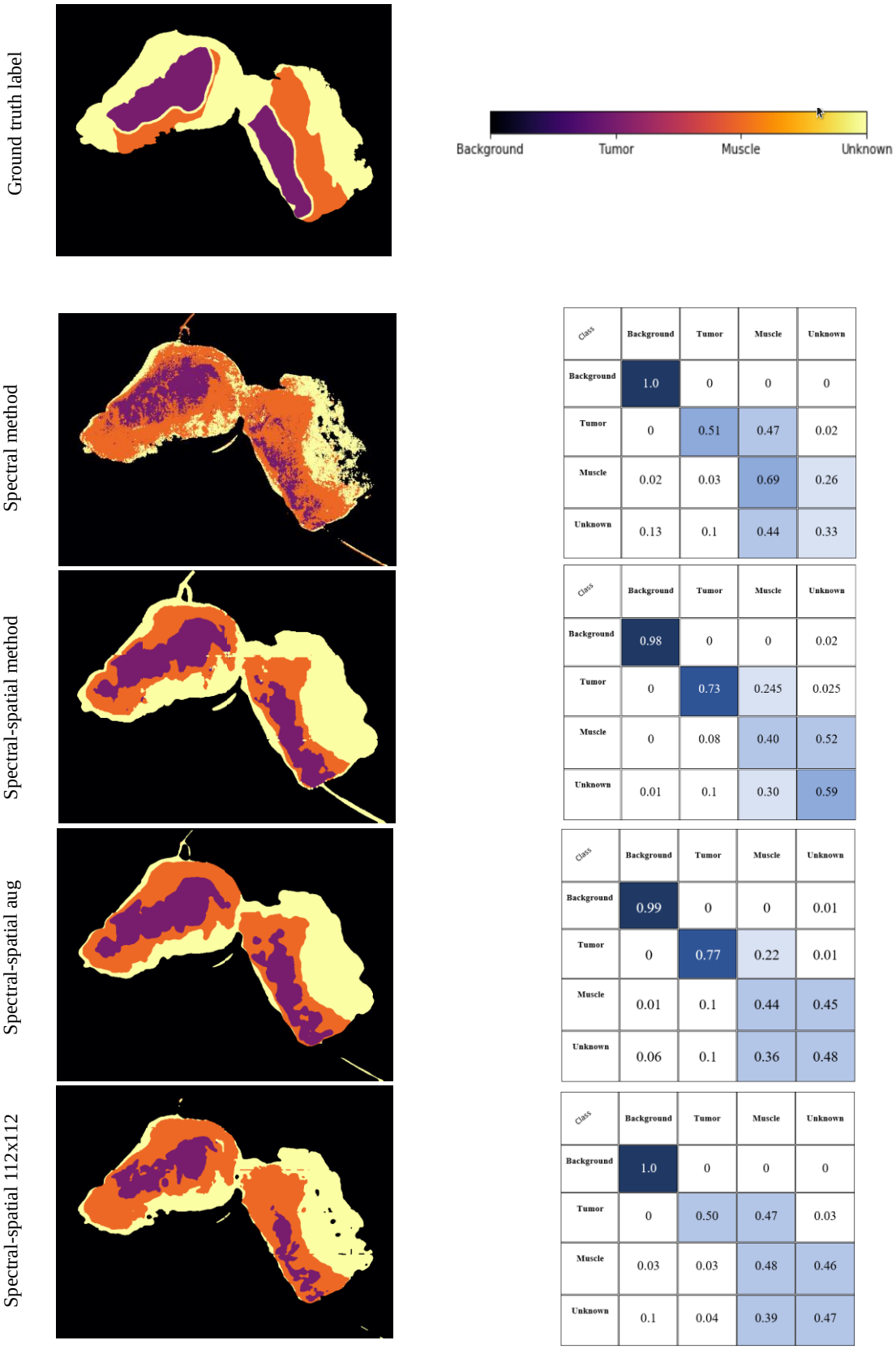
"What are the effects of data augmentation on the network's performance? Can it overcome the problems due to limited patient data?"

4.1 Comparison of results

For comparing the performances of both the spectral and spectral-spatial architectures, their leave-one-out prediction segmentation images are considered. Based on these two architectures, four different experiment predictions based on the leave-one-out training and testing scheme, and the performance metrics are published in this section. In the first experiment, a network based on the spectral feature learning method is trained using an input dimension of 16 x 16 x 164. For the second experiment, a simultaneous spectral-spatial feature learning network that trains on input samples of dimension 224 x 224 x 164 with its number of training samples based on Table 6. In the third experiment, data augmentation is performed on the training samples of dimension 224 x 224 x 164, with 90° clockwise rotation and vertical flipping, thus increasing the number of samples to thrice the previous experiment. In the final experiment, smaller sized input samples of dimension 112 x 112 x 164 are trained on an architecture similar to the second and third experiments, but with increased number of convolutional kernels.

There are seven patient samples considered for the experiments (#1, #2, #3, #4, #5, #6, and #7). Patient sample #8 is excluded from the training dataset because the spectral signals lack any inter-class difference (based on the reasoning in Section II, spectral data analysis). As for sample #5, the inter-class separation is low, but it exists across a few spectral bands. Therefore, we opted to include the sample in the training dataset. However, after performing these four experiments it can be observed that none of the methods could predict any tumor pixels in the final segmentations corresponding to this sample. Hence, #5 is utilized only for training in the leave-one-out scheme, but in testing it is not included and not accounted for in the performance metrics determined afterwards. For the six remaining patient samples, the final multi-class segmentations are compared with the labels and the precision, recall, and F-1 scores are determined. In order to understand the misclassifications of pixels, the true negative, true positive, false negative, and false positive values are determined for each tissue class and displayed using a confusion matrix representation. This is repeated for each of the samples and finally performances of all the experiments are juxtaposed to provide a comparison of the considered approaches and to answer the sub-questions in the research approach. While the research problem was instituted as a multi-class semantic segmentation to delineate different regions of the tissue and to model the uncertainty in manual annotation as a separate tissue class, the prominence of how accurately tumor is predicted cannot be understated. The performance metrics for the tumor class carry significant weight, while that of the background

and unknown tissue carry the least importance. Thus, the experiments are compared for the metrics corresponding to tumor prediction only.



Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0
Tumor	0	0.51	0.47	0.02
Muscle	0.02	0.03	0.69	0.26
Unknown	0.13	0.1	0.44	0.33

Class	Background	Tumor	Muscle	Unknown
Background	0.98	0	0	0.02
Tumor	0	0.73	0.245	0.025
Muscle	0	0.08	0.40	0.52
Unknown	0.01	0.1	0.30	0.59

Class	Background	Tumor	Muscle	Unknown
Background	0.99	0	0	0.01
Tumor	0	0.77	0.22	0.01
Muscle	0.01	0.1	0.44	0.45
Unknown	0.06	0.1	0.36	0.48

Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0
Tumor	0	0.50	0.47	0.03
Muscle	0.03	0.03	0.48	0.46
Unknown	0.1	0.04	0.39	0.47

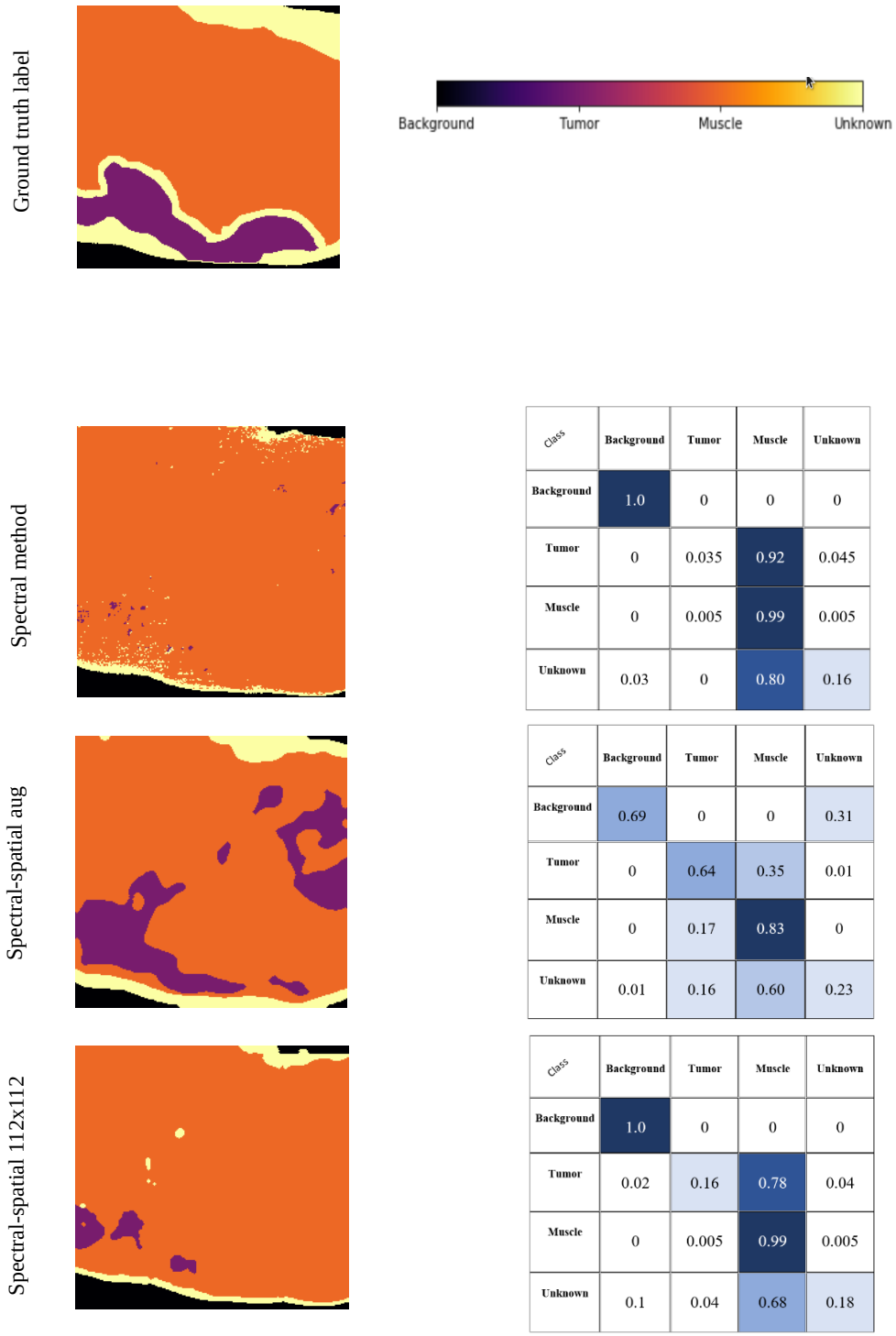
Figure 45: Left column: Top to bottom, label for #1, predictions for the four methods (spectral method, spectral spatial method, spectral spatial with data augmentation, and spectral spatial with 112 x 112 x 164). Right column: confusion matrices for the four considered methods.

4.1.1 Sample #1

The final segmentations corresponding to the hyperspectral data cube of patient sample #1 are displayed in Figure 45, along with their corresponding confusion matrices. The label is also provided for comparing the segmentations visually. The colorbar provided at the top of the figure provides a legend of the colormap scheme for different tissue classes. As for the confusion matrices, the intensity of the monochrome colormap varies with the value of the normalized confusion matrix (row elements adding up to 1.0). Thus, the darkest box indicates highest value, while lightest (white) box indicates lowest values. The diagonal elements indicate the recall value of each tissue class, while the off-diagonal elements provide the false positive (FP) and false negative (FN) rates.

By comparing the segmentation image of the first experiment with the ground truth label, it is possible to observe that the tumor regions are faintly predicted (especially the right excised tongue). This corresponds to the low recall value in the confusion matrix for tumor class, which also shows almost half the tumor class spectra misclassified as muscle class. The segmentation is discontinuous and pixelated because of the spectral feature classification approach and importantly the categorical cross-entropy loss function. The confidence rate for unknown prediction is low because of misclassification with background and tumor class. This low confidence rate for unknown tissue is insignificant to this research problem, however it is interesting to see misclassification of some of the unknown class spectra as background class (which could be due to the low intensity of the spectra corresponding to these two classes).

For the second experiment, the segmentation image shows discernible improvement in the confidence of tumor prediction (especially for the second tongue tissue), as can also be seen from the second confusion matrix. There is reduction in misclassification rate of tumor region to muscle class. Also, across the other classes like muscle and unknown, there is reduction in misclassification. The segmentation is smooth due to the Dice coefficient loss function in the spectral-spatial approach. Compared to this, the third experiment with data augmentation further improves the accuracy of prediction of tumor and muscle class in the segmentation. And it can be seen from Figure 56, this experiment provides the best F-1 score for this particular sample. In the fourth experiment with a smaller spatial dimension of 112 x 112 x 164, there is a drop in the confidence in tumor prediction, nearly to the values of the spectral approach in the first experiment. Thus, it can be concluded that data augmentation with spontaneous spectral-spatial learning performs the best for this patient sample.



Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0
Tumor	0	0.035	0.92	0.045
Muscle	0	0.005	0.99	0.005
Unknown	0.03	0	0.80	0.16

Class	Background	Tumor	Muscle	Unknown
Background	0.69	0	0	0.31
Tumor	0	0.64	0.35	0.01
Muscle	0	0.17	0.83	0
Unknown	0.01	0.16	0.60	0.23

Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0
Tumor	0.02	0.16	0.78	0.04
Muscle	0	0.005	0.99	0.005
Unknown	0.1	0.04	0.68	0.18

Figure 46: Left column: Top to bottom, label for #2, predictions for the three methods (spectral method, spectral-spatial method with data augmentation, and spectral-spatial method with input dimension 112 x 112 x 164). Right column: confusion matrices for the three considered approaches.

4.1.2 Sample #2

In Figure 46, the final segmentations of the experiments on sample #2 are shown along with their confusion matrices. After observing the final segmentations for this sample, it can be noticed that the first experiment provides very faint prediction of tumor and the second experiment with spectral-spatial method does not provide any prediction on tumor tissue. Owing to this, these experiment results are excluded from the results comparison section. This is also corroborated by the first confusion matrix which displays a negligible recall metric corresponding to tumor class. The second method does not predict tumor at all, therefore it is not displayed in the figure. Since this hyperspectral image has a smaller spatial dimension (560 x 336), in order to test for segmentation, only a single sub-crop of the spatial dimension 224 x 224 can be extracted. This particular sub-crop is chosen such that it represents majority of the true tumor pixels in the label. It is uniformly used on all the experiments to calculate the segmentation scores, even though full ROI segmentation is possible on the first and fourth experiments.

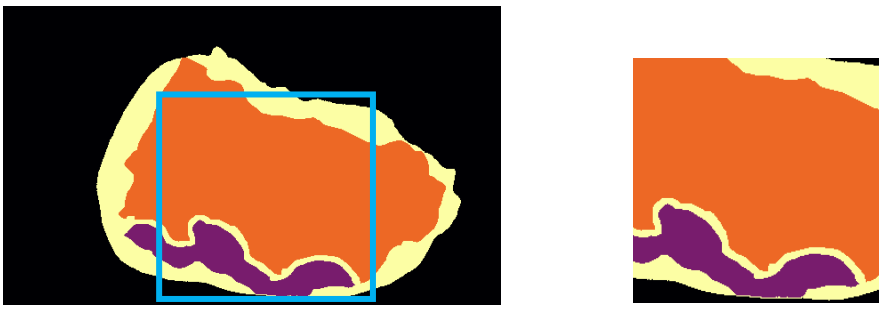
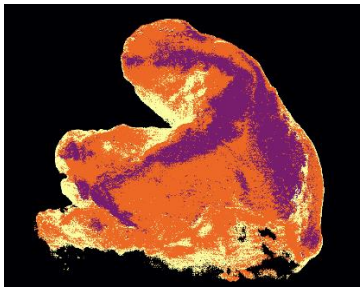
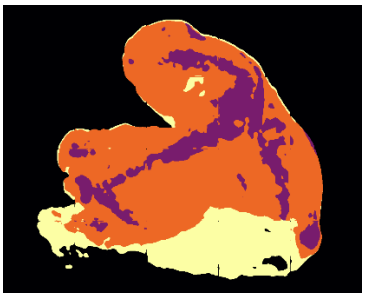


Figure 47: The original ROI spatial region of the label (left) and its smaller cropped area (right) emphasizing the tumor region.

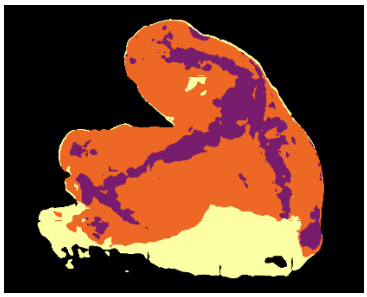
When trained on augmented data, the segmentation shows prediction of tumor regions and more than half of the true tumor pixels are correctly classified as tumor by the network. There are some false positives in the glare pixel region of the hyperspectral image, which could indicate generalization problems with unseen data. The fourth experiment shows a smaller predicted tumor area, with about 75% of the true tumor pixels appearing as false negatives. Interestingly, the false positives on the glare regions from the previous method no longer appear on the segmentation. While not large enough to contain the global contextual features to distinguish tumor regions from the rest in the tissue ROI, this particular approach can reduce the misclassification of glare regions.



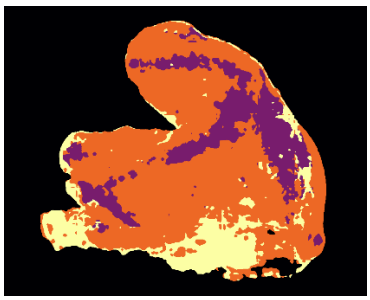
Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0
Tumor	0	0.47	0.49	0.04
Muscle	0	0.26	0.67	0.07
Unknown	0.22	0.07	0.44	0.27



Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0
Tumor	0	0.36	0.63	0.01
Muscle	0	0.16	0.76	0.08
Unknown	0.15	0.04	0.41	0.40



Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0.01
Tumor	0	0.40	0.58	0.02
Muscle	0	0.17	0.74	0.09
Unknown	0.15	0.05	0.38	0.42



Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0
Tumor	0	0.33	0.65	0.02
Muscle	0	0.17	0.74	0.09
Unknown	0.22	0.05	0.46	0.27

Figure 48: Left column: Top to bottom, label for #3, predictions for the four methods (spectral method, spectral-spatial method, spectral-spatial with data augmentation, and spectral-spatial with 112 x 112 x 164). Right column: confusion matrices for the four considered methods.

4.1.3 Sample #3

For the four experiments, the segmentations for the data cube of patient sample #3, along with the ground truth label can be seen in Figure 48. The confusion matrices are also available beside the segmentations. In the segmentation corresponding to the first experiment, it can be observed that the confidence of tumor prediction is higher on the convex regions on the tissue surface, and the segmentation follows the curvature of the tissue surface (as can be seen from the sample's hyperspectral or RGB image). Because of this, almost half the tumor spectra are misclassified as muscle, likewise some of the muscle spectra are misclassified as tumor. As can be seen in the spectral analysis of the samples, the inter-class separation in terms of spectral intensity is low, which can complicate the differentiation of tumor and muscle spectra.

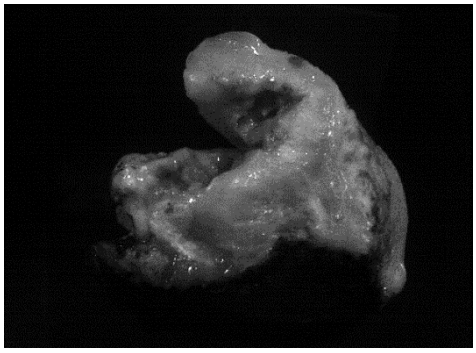
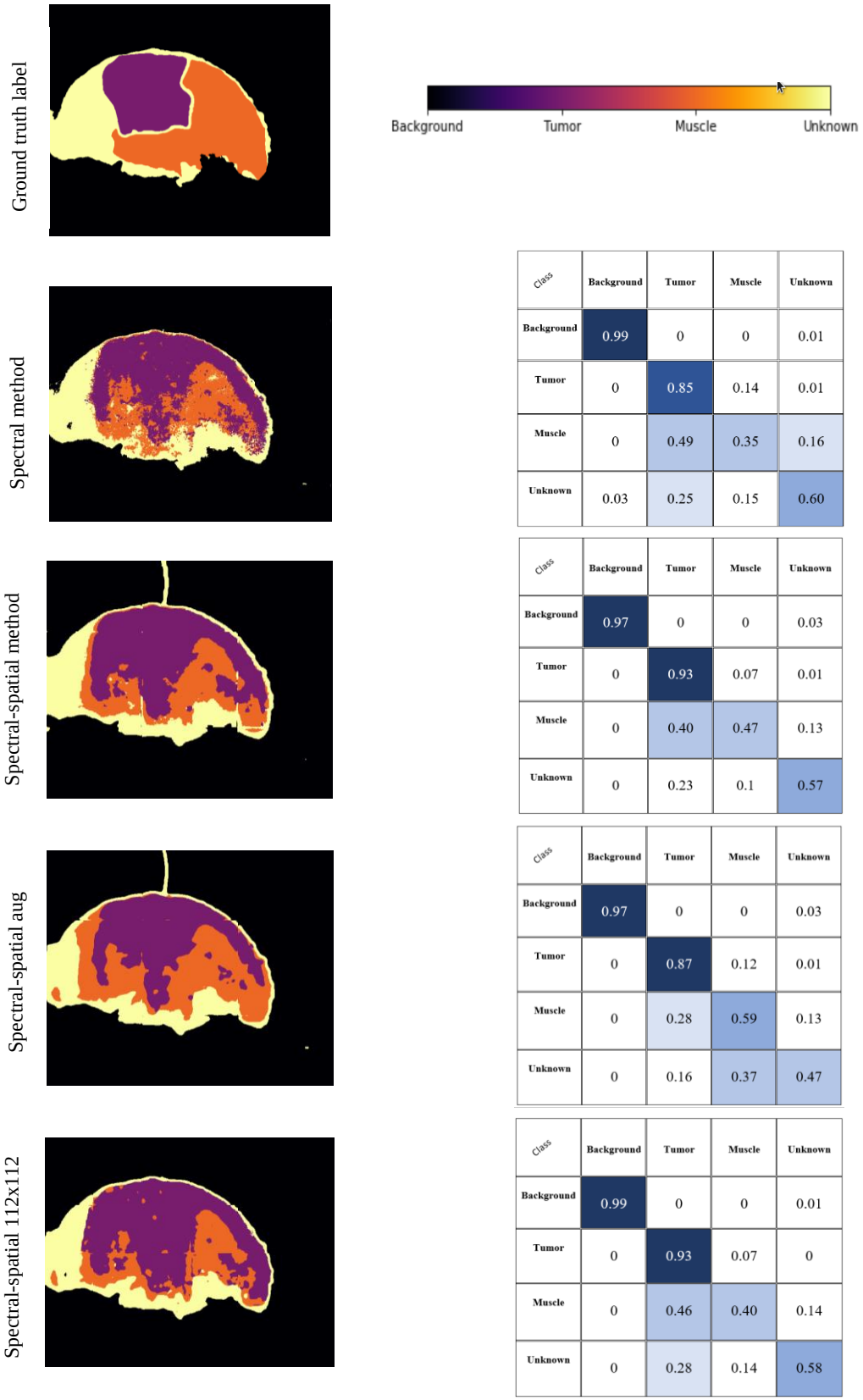


Figure 49: Representative hyperspectral image which shows tissue curvature on the central tissue region.

In the second segmentation, based on the spectral-spatial feature learning, there is higher misclassification of tumor spectra as muscle, but lower misclassification of muscle spectra as tumor. This reflects on the confusion matrix corresponding to this experiment. The influence of tissue curvature still remains significant in the segmentation. Since this particular data cube has a larger ROI, the number of sub-cropped areas available for training is higher compared to the data cubes of the other samples (seen from Table 6). This reduced number of training samples in the leave-one-out cross validation could have a bearing on the accuracy of tumor prediction in the #3 sample. By increasing the number of training samples through data augmentation, there is marginal increase in the tumor accuracy in the third experiment, however still lower than the value for learning individual spectral features in the first experiment. As for the final experiment with smaller input spatial dimensions, the misclassification is the highest for both tumor spectra to muscle and vice versa.



Class	Background	Tumor	Muscle	Unknown
Background	0.99	0	0	0.01
Tumor	0	0.85	0.14	0.01
Muscle	0	0.49	0.35	0.16
Unknown	0.03	0.25	0.15	0.60

Class	Background	Tumor	Muscle	Unknown
Background	0.97	0	0	0.03
Tumor	0	0.93	0.07	0.01
Muscle	0	0.40	0.47	0.13
Unknown	0	0.23	0.1	0.57

Class	Background	Tumor	Muscle	Unknown
Background	0.97	0	0	0.03
Tumor	0	0.87	0.12	0.01
Muscle	0	0.28	0.59	0.13
Unknown	0	0.16	0.37	0.47

Class	Background	Tumor	Muscle	Unknown
Background	0.99	0	0	0.01
Tumor	0	0.93	0.07	0
Muscle	0	0.46	0.40	0.14
Unknown	0	0.28	0.14	0.58

Figure 50: Left column: Top to bottom, label for #4, predictions for the four methods (spectral method, spectral-spatial method, spectral-spatial with data augmentation, and spectral-spatial method with 112 x 112 x 164). Right column: confusion matrices for the four considered methods.

4.1.4 Sample #4

For the sample #4, the labels and the segmentation results of the four experiments are shown in Figure 50. The first segmentation is a pixel-wise classification and by comparing with its label it can be noticed that there are many false positives (FP) in the segmentation. Almost half the spectra of the muscle tissue are misclassified as tumor. This can be noticed in the hyperspectral image of the sample, with the tissue along the top periphery having a convex surface (Figure 51). The spectra corresponding to this tissue region have higher intensities, hence have a high possibility of getting misclassified as tumor. This is consistent with the findings of the spectral analysis of this individual sample.

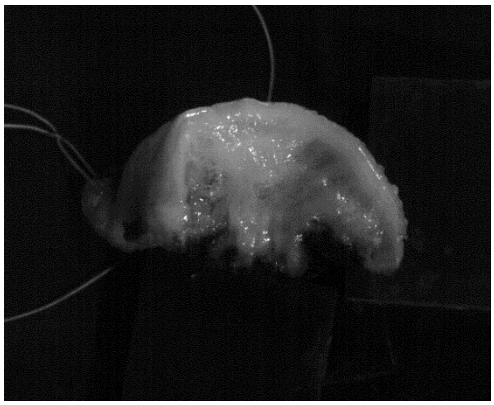
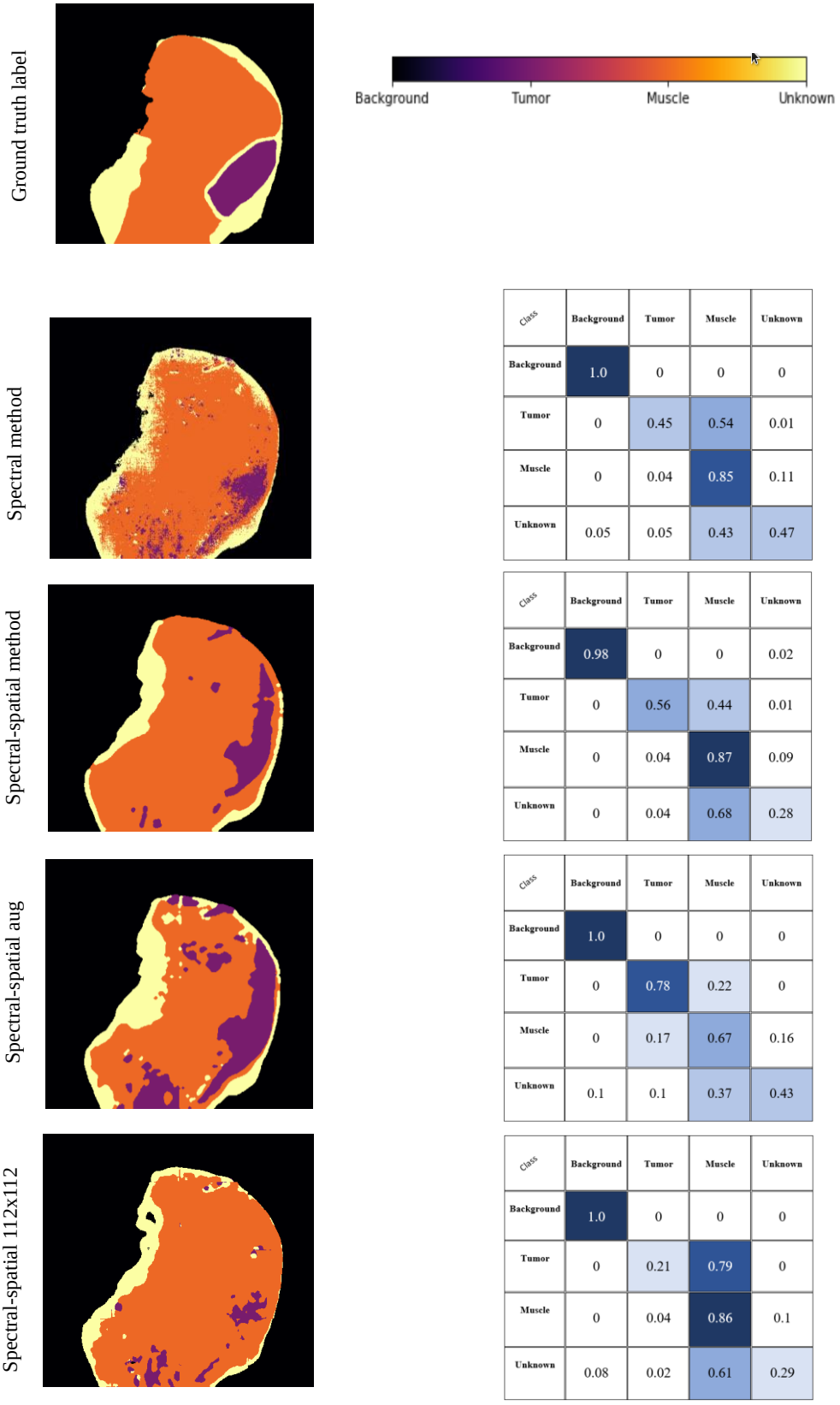


Figure 51: Representative hyperspectral image which shows tissue curvature on the periphery.

In the second segmentation image, the influence of the tissue curvature along the periphery still exists, while as per the confusion matrix the misclassification of the muscle spectra as tumor is slightly reduced. With data augmentation in the third segmentation, there is noticeable reduction in the convex muscle tissue areas misclassified as tumor, which is also reflected in the confusion matrix corresponding to muscle class. There is reduction in the stray tissue predictions on the right, which correspond to the high intensity glare pixels. Being one of the three samples (also #6 and #7, but they have higher intensity glare pixels, as seen from histogram) that has significant number of pixels as glare pixels (or spectra), removing it from the training dataset and performing data augmentation reduces the number of misclassifications of glare pixels that are comparatively low intensity.

In the final experiment, the confusion matrix shows increase in number of false positives for tumor and the segmentation image shows increase in the tumor regions corresponding to convex tissue regions and glare pixels. This result is very similar to the one produced in the first experiment and decreasing the spatial dimension of the input or increased number of learnable kernels has no positive effect, moving away from the spectral learning approach.



Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0
Tumor	0	0.45	0.54	0.01
Muscle	0	0.04	0.85	0.11
Unknown	0.05	0.05	0.43	0.47

Class	Background	Tumor	Muscle	Unknown
Background	0.98	0	0	0.02
Tumor	0	0.56	0.44	0.01
Muscle	0	0.04	0.87	0.09
Unknown	0	0.04	0.68	0.28

Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0
Tumor	0	0.78	0.22	0
Muscle	0	0.17	0.67	0.16
Unknown	0.1	0.1	0.37	0.43

Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0
Tumor	0	0.21	0.79	0
Muscle	0	0.04	0.86	0.1
Unknown	0.08	0.02	0.61	0.29

Figure 52: Left column: Top to bottom, label for #6, predictions for the four methods (spectral method, spectral-spatial method, spectral-spatial with data augmentation, and spectral-spatial with 112 x 112 x 164). Right column: confusion matrices for the four considered approaches.

4.1.5 Sample #6

The segmentations of the hyperspectral data cubes obtained by means of four experiments and the ground truth label of the data cube are available in Figure 52. In the first segmentation, the tumor region is localized but almost half the true tumor pixels turn out to be false negatives, misclassified as muscle class. There are also some false positives in the lower end of the tissue, which correspond to the glare pixels present in the hyperspectral image. By learning the combined spectral-spatial features in experiment two, there is a decrease in the number of false negatives as can be seen in the corresponding confusion matrix, but there is a noticeable increase in false positives, with respect to the raised portion of tissue along the periphery on the left misclassified as tumor region.

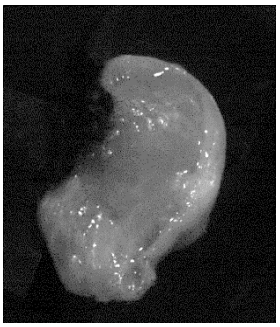
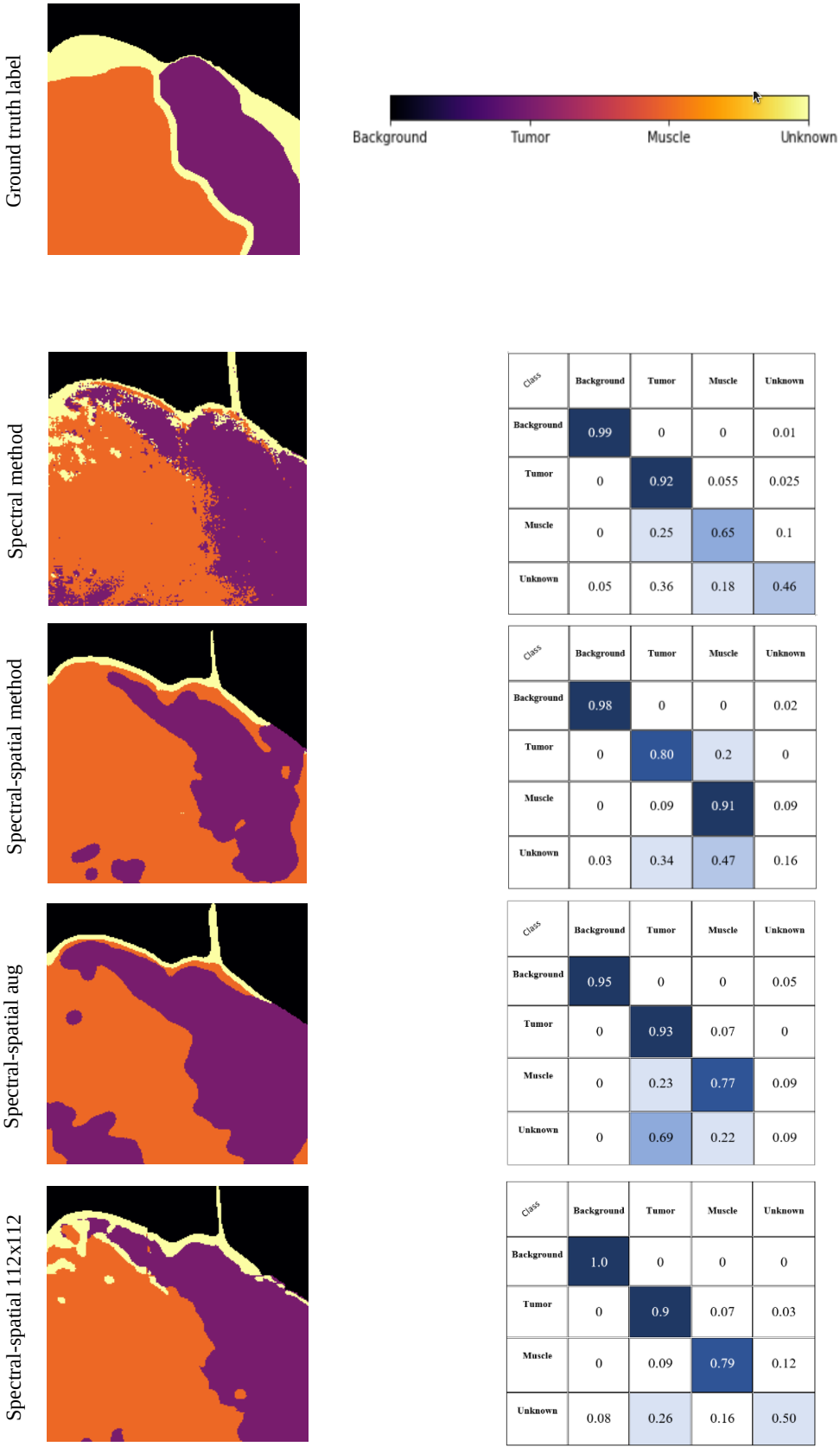


Figure 53: Representative hyperspectral image which shows tissue curvature on the periphery.

If data augmentation is introduced, it can be seen there is a further decrease in false negatives, but overall there is an increase in false positives due to the curvature of tissue along the periphery and the presence of glare pixels in the original hyperspectral image. The effect of misclassification to tumor seems rather pronounced after augmenting data, in a way that the additional data has reinforced how the network looks at glare pixels or spectra. It is also to be noticed that this sample has higher intensity glare pixels compared to the other samples. By excluding this sample from the training data set it could be possible that, the network does not explicitly learn to correctly classify the glare pixels from spatial context and hence does not generalize well on unseen data similar to this. In the final experiment with a smaller spatial size, the performance deteriorates with higher number of false negatives for tumor (misclassified as muscle). However, it generalizes better with respect to the glare pixel false positives, which could be explained in terms of the local receptive field that the neurons are able to observe.



Class	Background	Tumor	Muscle	Unknown
Background	0.99	0	0	0.01
Tumor	0	0.92	0.055	0.025
Muscle	0	0.25	0.65	0.1
Unknown	0.05	0.36	0.18	0.46

Class	Background	Tumor	Muscle	Unknown
Background	0.98	0	0	0.02
Tumor	0	0.80	0.2	0
Muscle	0	0.09	0.91	0.09
Unknown	0.03	0.34	0.47	0.16

Class	Background	Tumor	Muscle	Unknown
Background	0.95	0	0	0.05
Tumor	0	0.93	0.07	0
Muscle	0	0.23	0.77	0.09
Unknown	0	0.69	0.22	0.09

Class	Background	Tumor	Muscle	Unknown
Background	1.0	0	0	0
Tumor	0	0.9	0.07	0.03
Muscle	0	0.09	0.79	0.12
Unknown	0.08	0.26	0.16	0.50

Figure 54: Left column: Top to bottom, label for #7, predictions for the four methods (spectral method, spectral-spatial method, spectral-spatial with data augmentation, and spectral-spatial method with 112 x 112 x 164). Right column: confusion matrices for the four considered approaches.

4.1.6 Sample #7

For the final patient sample, Figure 54 displays the segmentation results and the ground truth labels, with their respective confusion matrices. In the spectral learning experiment, the segmentation reveals misclassification of muscle spectra as tumor due to the presence of glare spectra or pixels. The spectral-spatial learning shows improvement in performance by reducing the number of false positives of tumor in muscle tissue regions. For this sample, the tissue sample is smaller and hence its ROI, which means that a spatial dimension of 224 x 224 is appreciable when compared to the ROI spatial dimension (560 x 336). Therefore, only one sub-crop of the spatial size 224 x 224 can made from the ROI data cube which can comprise of majority of the tumor region represented in the ground truth. Since the prediction performance of tumor is of primary importance, this selective method of tumor region segmentation is implemented. Thus, for all the four experiments this selective spatial region is considered the label and the segmentation metrics are computed only for this region but trained on sub-crops obtained from the whole spatial region (ROI).

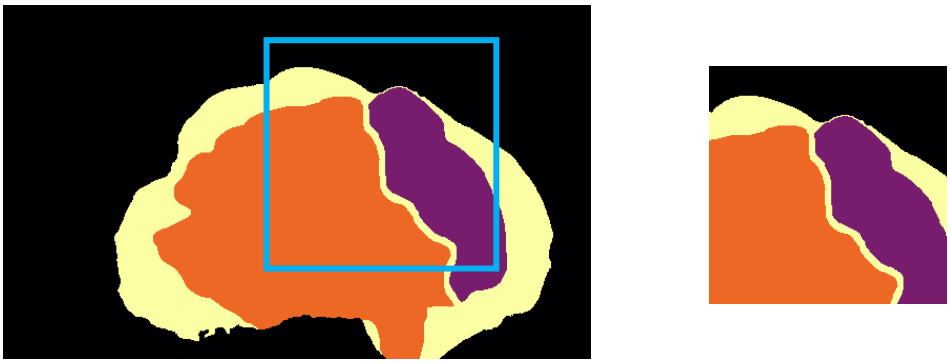


Figure 55: The original ROI spatial region of the label (left) and its smaller cropped area (right) emphasizing the tumor region.

In the third experiment, tumor false positives appear in the region of the glare pixels and the network fails to generalize them as muscle class based on spatial context. This could be the effect of removing one of the samples affected by high intensity glare. This effect subsequently reduces in the fourth experiment with a smaller spatial region of 112 x 112. This method with data augmentation, used more samples than the second and third methods, while also delivering comparable performance to them. This can also be noticed by observing the entire ROI's segmentation available for the first and fourth methods. This lends support to the reasoning that the choice of the spatial dimensions of the input must be commensurate to the local receptive field that the neurons can observe from the ROI image.

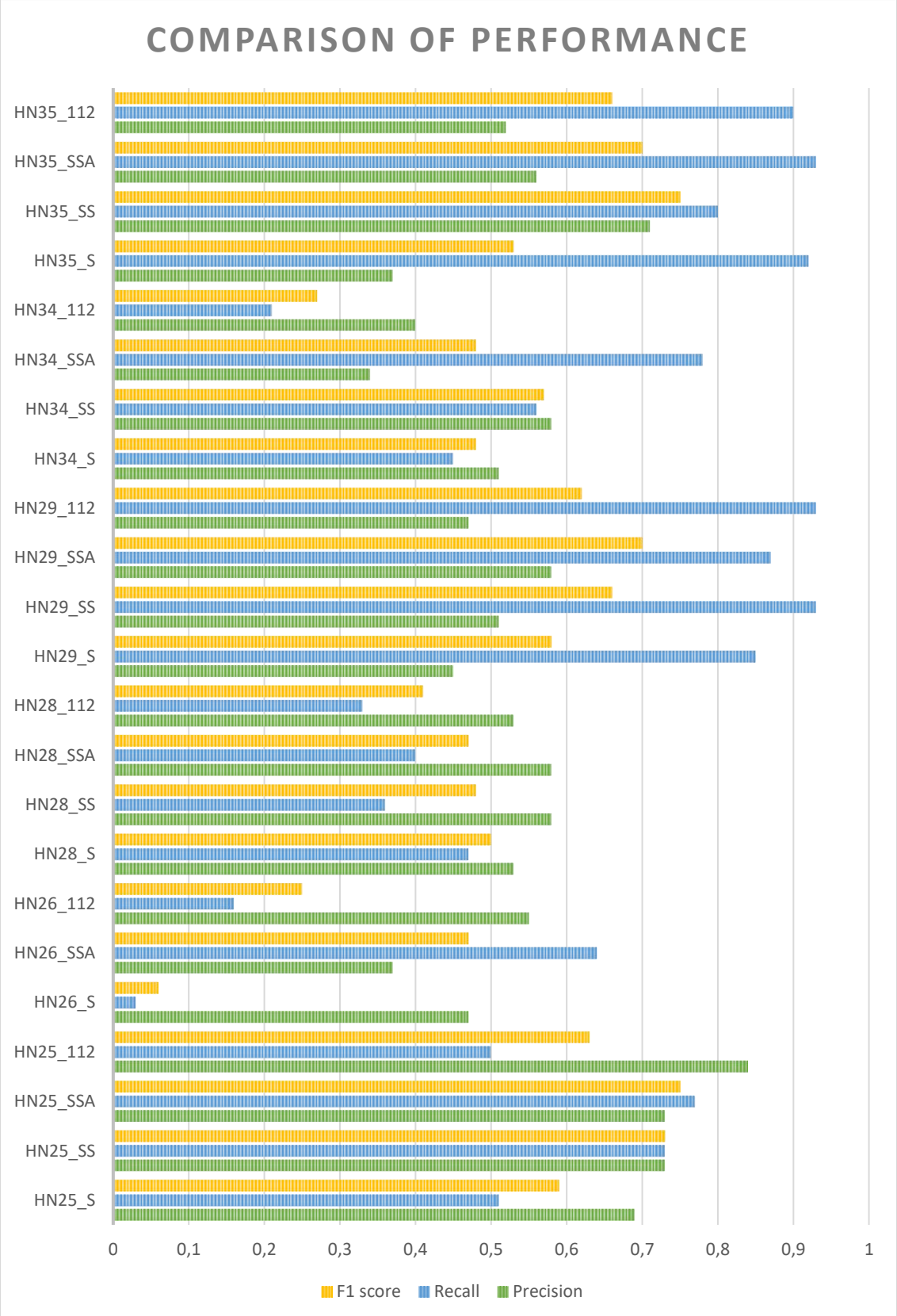


Figure 56: Comparison of precision, recall and F-1 score metrics corresponding to the testing samples across four experiments. By observing the F-1 score across all the patient samples, clearly the spectral-spatial method outperforms the spectral method. (_S, _SS, _SSA and _112 denote the experiments with spectral, spectral-spatial, spectral-spatial augmented and spectral-spatial 112x112 respectively.)

4.2 Discussion

By analyzing the segmentation results of the samples across the four experiments and their confusion matrices, we can understand how the different approaches learn features from hyperspectral data and predict on unseen data. Figure 56 shows the performance metrics precision, recall and F-1 score for all the tissue samples, across the four proposed experiments. The recall value was already available from the confusion matrices; however, we can use the F-1 score as a single metric to study how the networks in different experiments performed segmentation on unseen data, specifically for the tumor tissue class. While the deep spectral learning network with residual layers in the first experiment could learn discriminatory features from a hyperspectral data cube, a simple shallow convolutional neural network with 3-D convolutional kernel for simultaneous spectral-spatial learning improves the F-1 score on most of the samples (except #3, where there is comparable performance). Therefore, when training on limited hyperspectral data, it is necessary to exhaustively learn all the available information from the data cube instead of only the spectral information. The mean F-1 scores for all the experiments are shown in Figure 57, along with the standard deviations.

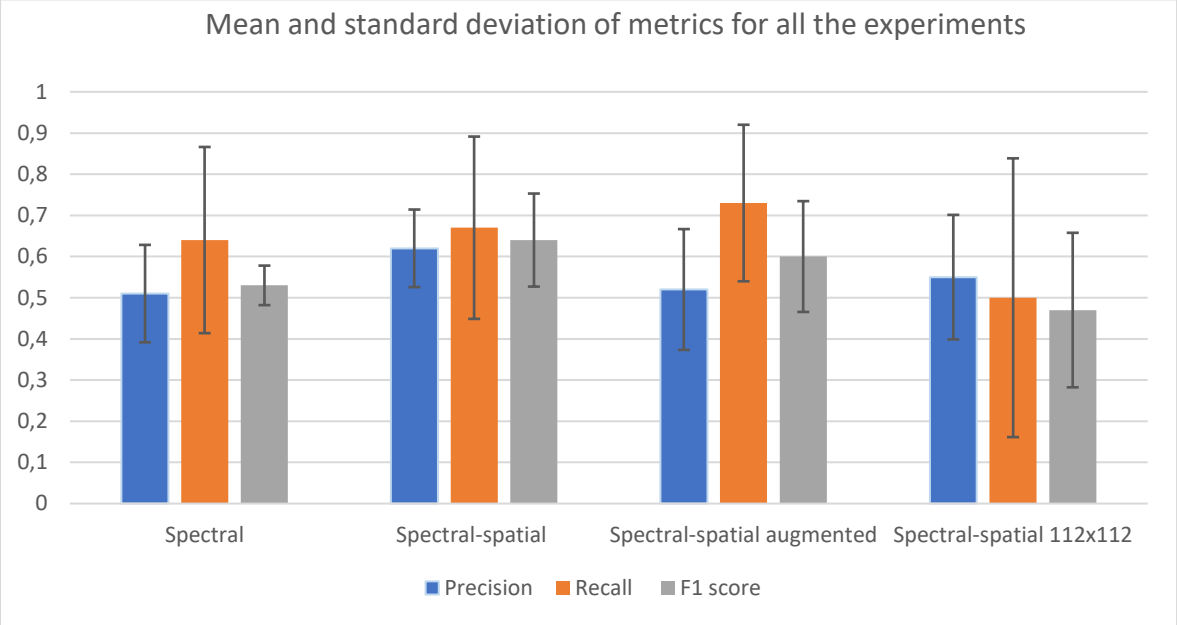


Figure 57: Bar graph showing the mean and standard deviation values of precision, recall and F-1 metrics across all testing samples, for four experiments.

By augmenting data to the training data set, the effect on segmentation performance as seen from Figure 56 is varied. For certain samples, data augmentation improves the F-1 score compared to the previous experiment. This is significant for #2, which had no tumor region predictions in the previous experiments, but by augmenting data tumor region is predicted in the final segmentation. However, an opposite effect is observed in samples like #6 and #7 which have intense glare pixels in the hyperspectral data cube. There is an increase in tumor

false positives in the glare regions which proves that while training on limited data, the generalization ability of the network to specific spatial features is affected. When augmenting data by geometrical transformations, the effect of glare pixels becomes pronounced, where the network misclassifies all spectra with high intensity as tumor, without regarding the spatial context. In the final experiment with a smaller spatial dimension of 112 x 112 on the training samples, no improvement is seen over previous two experiments with larger spatial dimension of 224 x 224. The performance is seen to vary with the spatial dimensions of the hyperspectral data cube ROI: for example, #1, #3 and #6 have larger spatial dimensions for their ROIs, while #2, #4 and #7 have smaller ROIs. By sub-cropping to a smaller spatial dimension like 112 x 112, the network is not able to capture global spatial context from the small neighborhood for these images with larger ROIs. The network with 224 x 224 spatial neighborhood - which is four times larger - performs better on all these samples. From the perspective of misclassification of glare pixels, the smaller spatial dimension performs better than the larger one, because of the choice of the earliest convolutional kernel 5 x 5 x 7, which can capture the local spatial features (instead of global spatial features) in the 112 x 112 spatial region better than a 7 x 7 x 9 on a 224 x 224 spatial region. From the table below, it can be observed that the local receptive field at the output layer of both the architectures have the same 63 x 63 area (arbitrarily ignoring the spectral dimension). Since the network views a 63 x 63 spatial area from the input image, it provides reasoning for worse performance of 112 x 112 architecture on larger ROI hyperspectral images (like #1, #3) compared to the smaller ROI images.

Table 14: Determination of receptive field for each layer of the proposed network

Layer	Spatial kernel size	Receptive field 224	Receptive field 112
Conv1	5 x 5	5 x 5	5 x 5
Maxpool	2 x 2	7 x 7	7 x 7
Conv 2	3 x 3	15 x 15	15 x 15
Conv 3	3 x 3	31 x 31	31 x 31
Upsampling	2 x 2	47 x 47	47 x 47
Conv 4	3 x 3	63 x 63	63 x 63
Final		**63 x 63**	**63 x 63**

As can be seen from Table 14, the effective receptive field is 63 x 63 in spatial dimension, which is smaller than the proposed input dimensions of 224 x 224 and 112 x 112. It was not possible to further increase the receptive field by (1) increasing the depth of the network (even with residual layers) because of non-convergence during training; (2) adding more pooling and upsampling layers due of loss of spatial resolution during these operations. Therefore, to expand the receptive field without suffering any of these limitations, dilated convolution layers can be used in place of regular convolution layers [55]. While this could be the way forward, it was not experimented extensively here as networks with dilated convolution were harder to train.

4.3 Other relevant methods

4.3.1 Noisy Spectrum

An individual spectrum in any of the available data cubes are found to be quite noisy (with lower SNR). Noise can be introduced to the signals due to many sources, mainly classified into photon noise, readout noise, dark noise and digitization noise. If we were to conceive an approach, where the network learns features from each individual spectrum with or without any spatial correlation information, it is worth investigating if any noise removal method would improve the segmentation of the data cube. In the following portion, we explore a few denoising or signal reconstruction methods to determine the influence of noise in the spectra.

4.3.2 Signal Smoothening

In order to smoothen the noisy spectral signals, various signal filtering techniques were evaluated. Among them, the Gaussian filter was applied to smoothen the spectral data of the data cubes and training was performed on the spectral-based deep learning network. This Gaussian filtering can be performed by applying a convolution kernel with a Gaussian function, with standard deviation σ of the distribution:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}}$$

The smoothening effect of Gaussian filter on two different spectra is illustrated in Figure 58. Another filter, the Savitzky-Golay filter, performs convolution on a subset of points in the spectra, with a window possessing a n-degree polynomial to fit the subset of data points. In order to smoothen the filter but to avoid losing finer spectral details, an 8^{th} degree polynomial is chosen. The smoothening effect by this filter is illustrated in Figure 58. When the smoothened spectral signals were used to train the proposed architecture, the performance deteriorated, with many false negative predictions of tumor. This could indicate that by smoothening the spectra, finer details of individual spectra, which can discriminate between tumor and other class spectra could be lost.

It was deliberated that a median filter not be used, despite its effectiveness in smoothening, since it creates unnatural spectral values that could not have been acquired by the sensor. By staying as close to the raw spectral data as possible, we can train the network to be robust to the varying inter-class separability across the tissue samples.

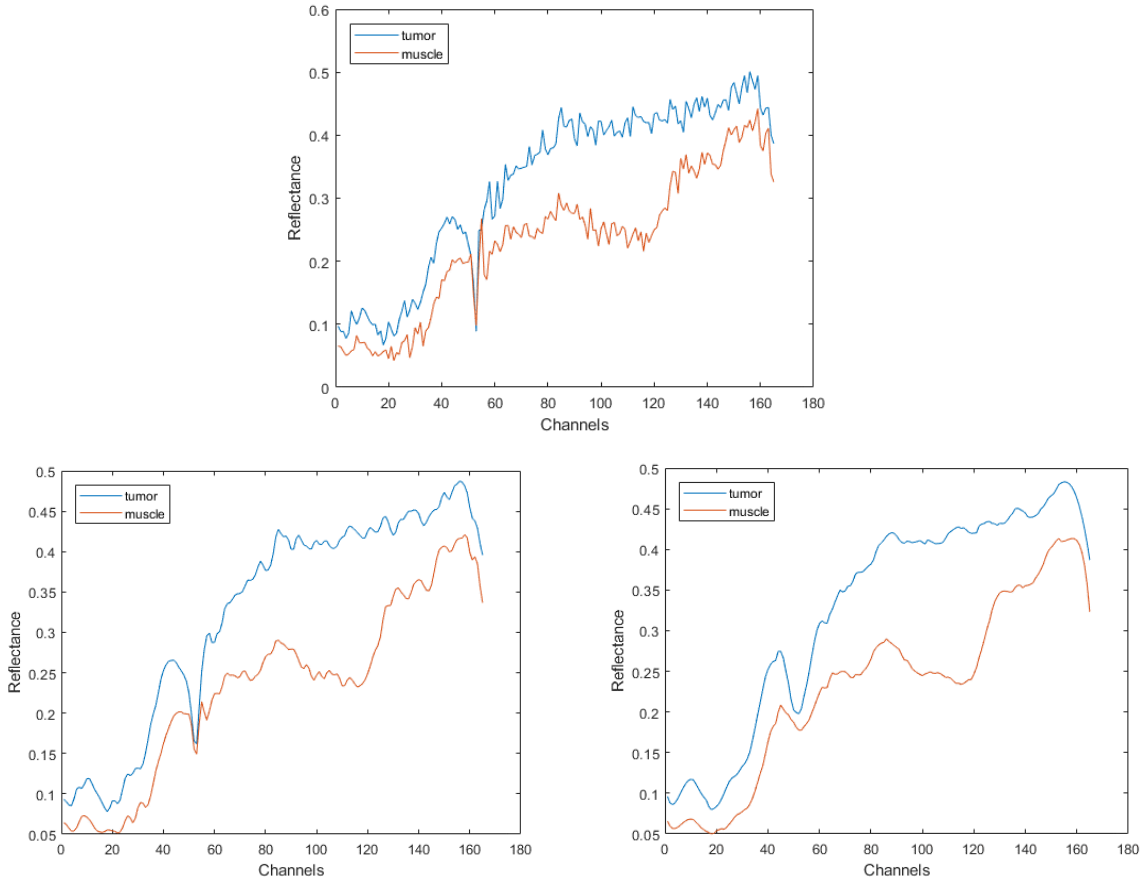


Figure 58: Plots showing the effect of filtering on noisy spectral signals. Counter clockwise from top: Graph of raw spectral signatures of tumor and muscle class; smoothening spectra using Savitzky-Golay filter; smoothening spectra using Gaussian filter.

4.3.3 Principal Component Analysis

PCA or Principal component analysis, is a statistical method that can be used to reveal the structure of a given data, set in a way which best explains the total variance of data. This can be computed by using the Singular Value Decomposition. By considering the data points along the spectral dimension as the variables, we can determine the linear combination of these variables that can account for the maximum variance in the data. These are called the principal components and it can be observed that almost 90% of the total variance can be captured by the first three principal components. By retaining these three components, we can now reconstruct the original signal by projecting them back into the original space. This can be done by multiplying the observations corresponding to the first three components, with the first three eigenvectors. By adding the mean of the original data, we get the projection of the reduced data on the original space or the reconstruction of original data corresponding to the three principal components or (90% of variance).

We can observe that this method can remove the noise present in the original data. However, by using this reconstructed data in the first architecture, there was no improvement as

anticipated. There was degradation in the prediction of tumor regions compared to using the raw noisy data. To check if loss of fine spectral features during the reconstruction method influenced this, increasing number of principal components were considered (4, 6, and 8). While this could account for 92% variance, the improvement of tumor prediction was only marginal. These tests led to the understanding that though noisy, the raw spectral data could have fine, discriminating spectral features crucial for distinguishing between tumor and the muscle class. The inter-class similarity in spectral signatures for certain samples should also be considered while investigating spectral filtering methods.

4.3.4 Non-Negative Matrix Factorization

NMF or non-negative matrix factorization is another type of factor analysis, where a non-negative matrix A (m x n) can be factorized into two non-negative matrices W (m x k) and H (k x n). This W x H factorization is a lower-rank approximation of the original matrix A, determined through an alternating least-square minimization of the residual between A and W*H:

$$\underset{W,H}{argmin}\|A - WH^T\|_F^2$$

It is required to provide an initial value to the matrices as W_0 and H_0, which can first be iteratively determined using the multiplicative update algorithm. By using the best of these values as W_0 and H_0, a set number of iterations (e.g. 1000) and lower rank k for the factorization, the data cube can be factorized into a lower rank approximation. This experiment was repeated for values of k = 3, 5 and 8. These approximations of the data cube were not adequate to predict the tumor pixels, further underlining the importance of the fine spectral features in distinguishing between tumor and the other classes. Figure 59 provides the result of NMF.

After experimenting with all the above methods to denoise the spectra in the data cubes, it was concluded that such methods do not improve the pixel-wise prediction of the hyperspectral data cubes. Therefore, it may be worth only looking at convolutional kernels of the size 1 x 1 x 5 or 1 x 1 x 7 to trade-off between learning the noisy spectra (with smaller 1 x 1 x 3 kernels) and also losing finer features to filtering or approximations.

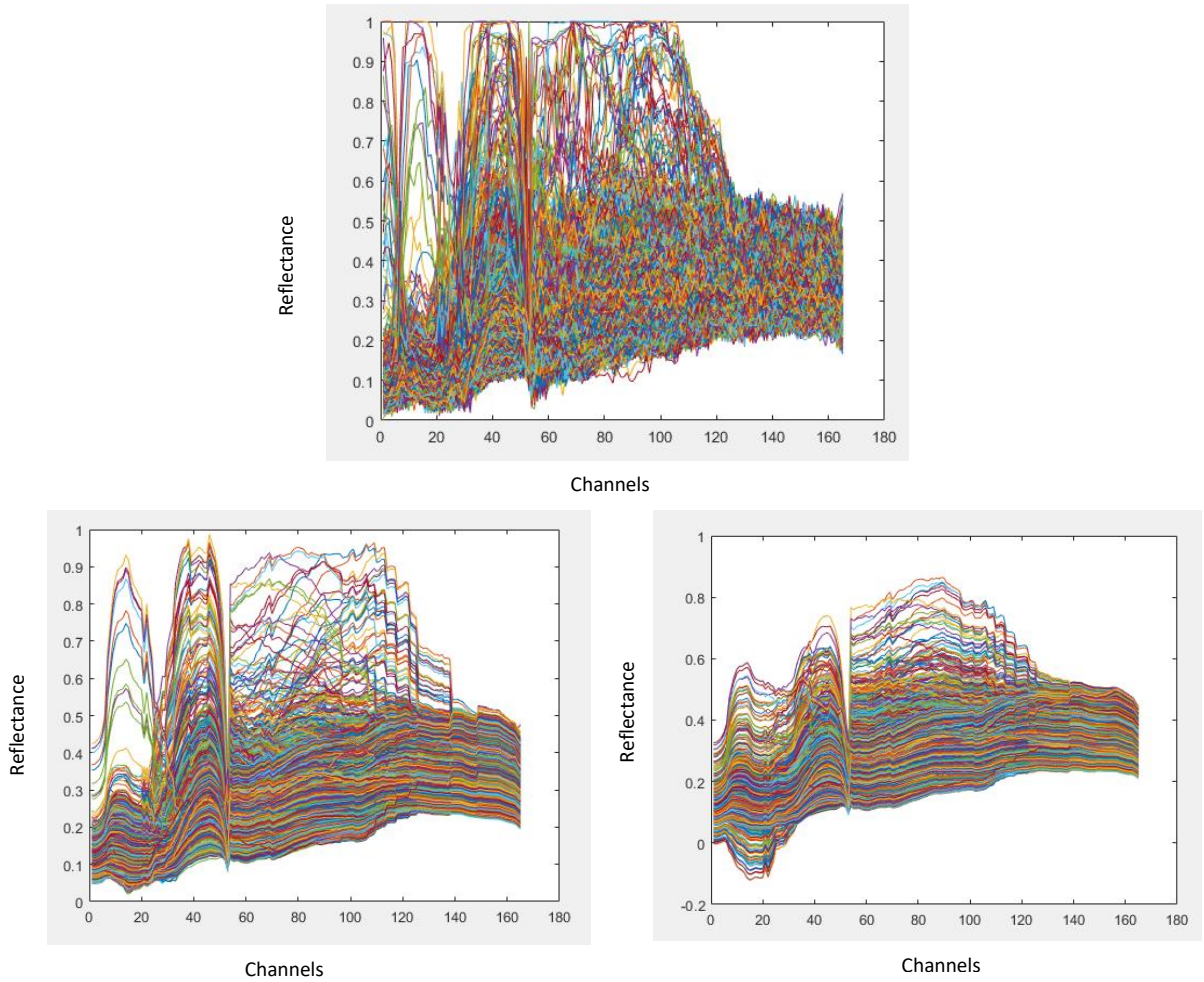


Figure 59: Plots showing the effect of approximations on raw noisy spectral signals. Clockwise from top: noisy raw spectra; spectra after NMF application; spectra after PCA application.

4.3.5 Curvature Dependence

The excised tongue tissue samples have uneven surfaces which are usually convex in nature. This is an important observation from a HSI perspective as these raised surfaces have spectral signatures that are higher in value compared to the neighboring flat or depressed areas. The consequence is, tumor affected tissue areas have higher spectral values and so do the convex areas of tissue that may or may not be tumor affected. This dependence of spectra on the tissue curvature can be remedied by the integral method proposed in previous research on medical hyperspectral image analysis. In this method, in order to make each spectrum independent of its intensity value, it can be divided by its integral or the area under the spectral curve. This method was applied on the individual hyperspectral data cubes and the following changes in spectral signatures are obtained as shown in Figure 61.

It can be seen for tissue class spectra, that the higher values spectra (corresponding to the convex areas) can be lowered and banded with the other lowered value tissue. While this can eliminate the dependence of tumor class spectra on tissue curvature (and the intensity values),

it introduces inter-class banding which bands the lower valued tumor spectra with the higher valued non-tumor class spectra like muscle and unknown.

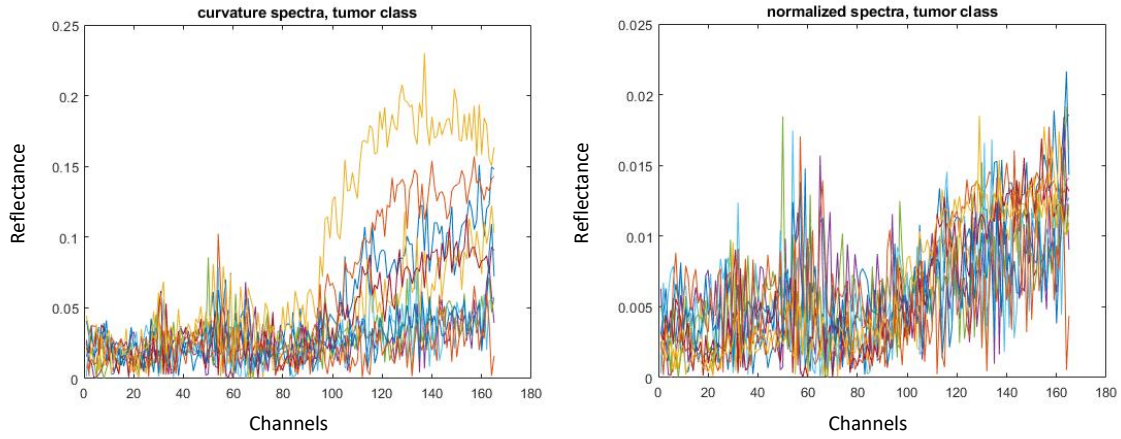


Figure 61: Plots showing the effect of integral normalization. Left: Raw spectral signature of tumor tissue; Right: Effect of diving individual spectrum by its integral or area under the curve.

This effect of curvature was confirmed in the segmentation output of the spectral approach of various samples (example #3, #4) and with introduction of spatial information in the second approach (spectral – spatial), improvement in this effect was anticipated. However, even this method was not able to generalize to new data cubes, by reducing the influence of tissue curvature on the segmentation regions.

4.4 Remaining challenges and future perspective

While this project was able to explore different aspects of configuring a convolutional neural network to segment hyperspectral images by seeking solutions to the formulated research problems and their respective sub-questions, there are still some open-ended challenges that have not found a solution in the course of this thesis.

1) **Effect of glare pixels:** Should these pixels be entirely excluded from the training data set or should they be allowed to learn to generalize in order to make the network more robust to such illumination defects?

2) **Sparse annotation**: Should the tissue classes be completely annotated for the hyperspectral images? How does the network handle sparse annotations (partially annotated tissue regions)?

3) Curvature correction: What techniques can be used to eliminate the dependence of tumor prediction on the curvature of tissue surface? Can a mapping of the tissue

thickness be determined, or can a network be trained to include the curvature during classification of pixels?

4) **Spatial features:** It is evident that inclusion of spatial features in feature learning improves performance, but can there be an efficient way of using the spatial features without losing information through downsampling and preprocessing?

While this thesis would serve as a pilot project to implement automated medical image segmentation on hyperspectral images, it important to provide some direction to future research and recommendations based on the findings of this work. More experimentation with feature-efficient architectures like the spectral-spatial network should be carried out, especially without downsampling layers. This was necessary in this thesis due to memory constraints that were posed when working with larger data dimensions, however it can be circumvented by utilizing techniques like dilated convolutions, which preserve the spatial features by only introducing holes in between pixels, allowing deeper architectures to be trained without increasing the network parameters excessively. This could also pave way for experimenting with architectures that contain skip connections similar to U-Net and ResNet, especially for the spectral-spatial approach.

It would also be interesting to investigate the correlation of spatial dimension choice on the glare pixels introducing tumor false positives. More experiments varying spatial dimensions of input data and also the convolutional kernels should be performed in order to make the network learn both finer and global spatial features. By incorporating multi-scale learning, a balanced method of learning spectral-spatial features can be formulated.

Further, sparse annotations can be explored to eliminate time spent on creation of complete ground truth labels and the problem can also be reconstituted with fewer tissue classes. It would also be worth exploring transfer learning of models, which could ease the training process. However, no particularly suitable trained model was found because most of the research on hyperspectral data prior to this thesis was on landcover classification and the learned spectral features could be too different from the features in medical hyperspectral images and fine tuning the kernel layers may not yield the desired result. It can be experimented with, which could reduce the time of configuring an architecture from scratch and make it possible to place emphasis on developing better techniques in training the network. While more data is always welcome, lack of it made this thesis even more challenging to work on!

Conclusion

This Master's thesis was a study to explore possibilities of automated image analysis using deep learning on the emerging medical imaging modality called hyperspectral imaging. Previously, image analysis tasks like classification, detection and segmentation involved feature engineering steps with medical domain expertise to analyze the medical images. With deep learning, a supervised learning method to automatically learn the features from the image can be developed. The research problem was to develop a proof of concept for a non-invasive, automatic segmentation tool that can assist surgeries. The following research question should be answered to draw conclusions from this study.

"Can a convolutional neural network perform tissue segmentation on limited patient data?"

Two different approaches to feature learning were proposed: spectral and spectral-spatial features. For each of these approaches, different architectures were experimented with, leading to two different architectures. Based on these two architectures, four experiments were devised according to the input dimensions of the image data. After training the networks for these experiments, pixel-wise segmentation images were generated (predictions) and based on the F-1 metric, it can be concluded that learning both spectral and spatial information would improve segmentation performance. Within spectral-spatial method, the basic architecture (224 x 224 x 164) produces a mean F-1 score of 0.64 and with data augmentation a mean F-1 score of 0.6. This outperforms experiments with smaller input dimensions (112 x 112 x 164) and with only spectral features. Thus, even with limited patient data, networks which can generalize on new patient data can be crafted.

Appendix I

Spectral reconstruction using PCA

In order to understand if the noise affecting the individual spectrum can have influence on classification performance by the spectral-only network proposed in Chapter III, different methods of spectrum approximation were performed. Principal component analysis or PCA was among the considered approaches to find an approximation of the original hyperspectral image data. The PCA of this data was computed using the Singular Value Decomposition (SVD) method, to decompose the matrix X of the dimension M x n into $X = U\Sigma V^T$. By considering the first k dimensions of the reduced U space (or k principal components which are the columns of $U\Sigma$) and multiplying with the corresponding reduced dimension matrices U and Σ. By adding back the mean vector of the original matrix, the reconstruction of the hyperspectral data can be obtained, based on the chosen number of first principal components.

Input: M x n matrix X, reshaped from a l x m x n hyperspectral image matrix

Output: Reconstructed X, based on a lower rank approximation

Perform SVD $\rightarrow$ $X = U\Sigma V^T$

Select first k Principal Components or columns of $U\Sigma$

Multiply matrices $\rightarrow$ $\hat{X} = U_{:,1:k}\, \Sigma_{1:k,1:k}\, V^T_{:,1:k}$

Reconstructed $\hat{X}_{recon} = \hat{X} + \mu$

Reshape M x n to l x m x n again

The matrix reconstruction is performed for different number of first principal components like 3, 4, 6 and 8. The total variance represented by these principal components ranges from 90 to 92%, the reconstructions are not adequate to improve the performance over that of the original noisy spectra.

Abbreviations

HSI	Hyperspectral imaging
VIS	Visible
NIR, IR	Near-Infrared, Infrared
UV	Ultraviolet
AOTF	Acousto-optical tunable filter
ICA	Independent component analysis
PCA	Principal component analysis
LDA	Linear discriminant analysis
SVM	Support vector machines
CNN	Convolutional neural network
NMF	Nonnegative matrix factorization
BLDE	Balanced local discriminant embedding
LR	Logistic regression
k-NN	k-nearest neighbor
DTC	Decision tree classifier
NAPDH	Nicotinamide adenine dinucleotide phosphate

References

[1] T. J. Malthus and P. J. Mumby, “Remote sensing of the coastal zone: An overview and priorities for future research,” *Int. J. Remote Sens.*, vol. 24, no. 13, pp. 2805–2815, 2003.

[2] A. A. Gowen, C. P. O’Donnell, P. J. Cullen, G. Downey, and J. M. Frias, “Hyperspectral imaging – an emerging process analytical tool for food quality and safety control,” *Trends Food Sci. Technol.*, vol. 18, no. 12, pp. 590–598, 2007.

[3] G. Lu and B. Fei, “Medical hyperspectral imaging: a review,” *J. Biomed. Opt.*, vol. 19, no. 1, p. 010901, 2014.

[4] P. W. Yuen and M. Richardson, “An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition,” *Imaging Sci. J.*, vol. 58, no. 5, pp. 241–253, 2010.

[5] Q. Li, X. He, Y. Wang, H. Liu, D. Xu, and F. Guo, “Review of spectral imaging technology in biomedical engineering: achievements and challenges,” *J. Biomed. Opt.*, vol. 18, no. 10, p. 100901, 2013.

[6] G. Hughes, “On the Mean Accuracy of Statistical Pattern Recognizers,” *IEEE Trans. Inf. Theory*, vol. 14, pp. 55–63, 1968.

[7] L. Zhang, L. Zhang, D. Tao, and X. Huang, “On combining multiple features for hyperspectral remote sensing image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 879–893, 2012.

[8] Q. Wang, J. Lin, and Y. Yuan, “Salient Band Selection for Hyperspectral Image Classification via Manifold Ranking,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 27, no. 6, pp. 1279–1289, 2016.

[9] Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, “Multiple Spectral-Spatial Classification Approach for Hyperspectral Data,” *Geosci. Remote Sensing, IEEE Trans.*, vol. 48, no. 11, pp. 4122–4132, 2010.

[10] Z. Zhong, J. Li, Z. Luo, and M. Chapman, “Spectral-Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, 2018.

[11] Y. Chen, Z. Lin, X. Zhao, S. Member, G. Wang, and Y. Gu, “Deep Learning-Based Classi fi cation of Hyperspectral Data,” *Ieee J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, 2014.

[12] W. Li, G. Wu, F. Zhang, and Q. Du, “Hyperspectral Image Classification Using Deep Pixel-Pair Features,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, 2016.

[13] Y. Chen, X. Zhao, S. Member, X. Jia, and S. Member, “Spectral – Spatial Classification of Hyperspectral Data Based on Deep Belief Network,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, 2015.

[14] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, “Deep Feature Extraction and

Classification of Hyperspectral Images Based on Convolutional Neural Networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, 2016.

[15] W. Zhao and S. Du, "Spectral-Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, 2016.

[16] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, 2017.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[18] L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised spectral-spatial feature learning via deep residual conv-deconv network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 391–406, 2018.

[19] L. V Wang and H. I. Wu, "Introduction," in *Biomedical Optics: Principles and Imaging*, Hoboken: John Wiley & Sons Inc., 2009, pp. 1–15.

[20] J. A. Freeberg, J. L. Benedet, C. MacAulay, L. A. West, and M. Follen, "The performance of fluorescence and reflectance spectroscopy for the in vivo diagnosis of cervical neoplasia; point probe versus multispectral approaches," *Gynecol. Oncol.*, vol. 107, no. 1, pp. S248–S255, 2007.

[21] D. G. Ferris *et al.*, "Multimodal hyperspectral imaging for the noninvasive diagnosis of cervical neoplasia," *J. Low. Genit. Tract Dis.*, vol. 5, no. 2, pp. 65–72, 2001.

[22] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2017," *CA. Cancer J. Clin.*, vol. 67, no. 1, pp. 7–30, 2017.

[23] K. Masood, N. Rajpoot, K. Rajpoot, and H. Qureshi, "Hyperspectral Colon Tissue Classification using Morphological Analysis," *Int. Conf. Emerg. Technol.*, no. November, pp. 735–741, 2006.

[24] M. Maggioni *et al.*, "Hyperspectral microscopic analysis of normal, benign and carcinoma microarray tissue sections - art. no. 60910I," *Opt. Biopsy VI*, vol. 6091, no. d, pp. I910–I910, 2006.

[25] E. J. M. Baltussen *et al.*, "Real-time tissue classification using hyperspectral imaging ; a way towards smart laparoscopic colorectal surgery," vol. 24, no. 1, 2019.

[26] Z. Liu, H. Wang, and Q. Li, "Tongue tumor detection in medical hyperspectral images," *Sensors*, vol. 12, no. 1, pp. 162–174, 2012.

[27] N. Bedard *et al.*, "Multimodal snapshot spectral imaging for oral cancer diagnostics: a pilot study," *Biomed. Opt. Express*, vol. 4, no. 6, p. 938, 2013.

[28] D. Hattery, M. Hassan, S. Demos, and A. Gandjbakhche, "Hyperspectral imaging of Kaposi's Sarcoma for disease assessment and treatment monitoring," *Appl. Imag. Pattern Recognit. Work. 2002 Proc.*, vol. 31, p. 124, 2002.

[29] V. Zheludev *et al.*, "Delineation of malignant skin tumors by hyperspectral imaging using diffusion maps dimensionality reduction," *Biomed. Signal Process. Control*, vol. 16, pp. 48–60, 2015.

[30] M. Nathan, A. S. Kabatznik, and A. Mahmood, “Hyperspectral imaging for cancer detection and classification,” *2018 3rd Bienn. South African Biomed. Eng. Conf. SAIBMEC 2018*, pp. 1–4, 2018.

[31] S. V. Panasyuk *et al.*, “Medical hyperspectral imaging to facilitate residual tumor identification during surgery,” *Cancer Biol. Ther.*, vol. 6, no. 3, pp. 439–446, 2007.

[32] K. J. Zuzak, S. C. Naik, G. Alexandrakis, D. Hawkins, K. Behbehani, and E. H. Livingston, “Characterization of a near-infrared laparoscopic hyperspectral imaging system for minimally invasive surgery,” *Anal. Chem.*, vol. 79, no. 12, pp. 4709–4715, 2007.

[33] H. Akbari, Y. Kosugi, K. Kojima, and N. Tanaka, “Detection and analysis of the intestinal ischemia using visible and invisible hyperspectral imaging,” *IEEE Trans. Biomed. Eng.*, vol. 57, no. 8, pp. 2011–2017, 2010.

[34] L. Ma *et al.*, “Deep learning based classification for head and neck cancer detection with hyperspectral imaging in an animal model,” no. March 2017, p. 101372G, 2017.

[35] M. Halicek *et al.*, “Deep convolutional neural networks for classifying head and neck cancer using hyperspectral imaging,” *J. Biomed. Opt.*, vol. 22, no. 6, p. 060503, 2017.

[36] M. Halicek *et al.*, “Optical biopsy of head and neck cancer using hyperspectral imaging and convolutional neural networks,” in *Optical Imaging, Therapeutics, and Advanced Technology in Head and Neck Surgery and Otolaryngology 2018*, 2018.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “AlexNet,” *NIPS*, 2012.

[38] H. Zhu, B. Chu, Y. Fan, X. Tao, W. Yin, and Y. He, “Hyperspectral Imaging for Predicting the Internal Quality of Kiwifruits Based on Variable Selection Algorithms and Chemometric Models,” *Sci. Rep.*, vol. 7, no. 1, p. 7845, 2017.

[39] H. Wang and B. Raj, “On the Origin of Deep Learning,” pp. 1–72, 2017.

[40] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.

[41] L. D. Le Cun Jackel, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, B. Le Cun, J. Denker, and D. Henderson, “Handwritten Digit Recognition with a Back-Propagation Network,” *Adv. Neural Inf. Process. Syst.*, pp. 396–404, 1990.

[42] M. A. Nielson, *Neural Networks and Deep Learning*. Determination Press, 2015.

[43] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8689 LNCS, no. PART 1, pp. 818–833, 2014.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.

[45] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” Nov. 2015.

[46] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes*

Bioinformatics), vol. 9908 LNCS, pp. 630–645, 2016.

[47] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” Feb. 2015.

[48] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How Does Batch Normalization Help Optimization? (No, It Is Not About Internal Covariate Shift),” *CoRR1*, 2018.

[49] S. Ruder, “An overview of gradient descent optimization algorithms,” Sep. 2016.

[50] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” pp. 1–8, 2015.

[51] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, “The importance of skip connections in biomedical image segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10008 LNCS, pp. 179–187, 2016.

[52] A. Abdulkadir, S. S. Lienkamp, and O. Ronneberger, “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation ¨.”

[53] F. Milletari, N. Navab, and S.-A. Ahmadi, “V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation,” pp. 1–11, 2016.

[54] T. M. Quan, D. G. C. Hildebrand, and W.-K. Jeong, “FusionNet: A deep fully residual convolutional neural network for image segmentation in connectomics,” 2016.

[55] W. Luo, Y. Li, R. Urtasun, and R. Zemel, “Understanding the Effective Receptive Field in Deep Convolutional Neural Networks,” no. Nips, 2017.