Cancer Classification in Human Brain and Prostate Using Raman Spectroscopy and Machine Learning

by

Jeremy Pinto

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2017

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Real-time assisted classification of cancerous and healthy human tissue is useful to surgeons since visual classification of cancer boundaries is almost impossible to the naked eye. Raman spectroscopy can be used to quantify the inelastic scattering of light in molecules by analyzing the interaction of photons with the vibrational modes of biological tissue. Raman spectroscopy can therefore serve as a tool to uniquely identify the presence of certain types of cells and their respective pathologies. In particular, Raman spectroscopy can be used to detect cancer cells in-vivo in affected human tissue by using various machine-learning algorithms. We showed that preprocessing steps have a significant impact on classification results and that the IModPoly algorithm performed similarly to the Zhang algorithm however the IModPoly algorithm had signficantly faster runtimes, which is more suitable for real-time classsification. We studied the performance of different classifiers on Raman spectrums acquired from human brain and prostate. We compared the performance of support vector machines, convolutional neural networks and multi-layer perceptrons. We've shown that a convolutional neural network and support vector machines have similar performance metrics when applied to pre-processed Raman spectrums acquired from human prostates when using a K-fold cross validation scheme, with mean ROC AUC 0.941 ± 0.017 and 0.943 ± 0.018 respectively, and that these outperform the MLP with an AUC of 0.935 ± 0.021. We show that data-reduction through PCA and AutoEncoders allow for similar, but overall worse, classification performance through data reduction of up to 2.5 times and that using the raw input as a feature space results overall in higher AUC across classifiers. On the smaller brain dataset, we found that the SVM outperformed the ConvNet and MLP with respective AUCs of 0.955 ± 0.029, 0.941 ± 0.036 and 0.846 ± 0.130. Using ConvNets and K-fold cross-validation, we were able to achieve an average accuracy, sensitivity and specificity of 0.921, 0.785, and 0.947 on the prostate dataset and 0.891, 0.944 and 0.825 respectively on the brain dataset. We show that classification metrics drop significantly when using a leave-one-patient-out approach compared to K-fold and show examples of clustering among patients within datasets, suggesting that data collection ensuring more uniform signal collection is of higher priority for robust performance over the choice of classifiers. We studied the use of transfer learning from one dataset to another, and showed limited increase in performance when training the classifier on the prostate dataset before applying it to the brain dataset. We showed a clear distinction using t-SNE in the feature space of brain and prostate datasets, demonstrating the clear biological differences that can be captured using Raman spectroscopy. Future work needs to address the shortcoming of the leave-one-patient-out approach compared to K-fold and the apparent clustering of patient data. It is imperative to determine if the source of clustering is inherent to patients or a result of bias in current protocols. The possibility of calibrating or fine-tuning data in real-time during clinical procedures should also be explored in future work as well as more refined preprocessing procedures.

Acknowledgements

Writing a thesis is challenging and having the right support is critical. I would like to thank my supervisor, John Zelek, for guiding me successfully through the journey of grad school and for being both a friend and a supervisor. I would like to thank JZ corner and all of the people in sleazy4 who made the grad school experience much more enjoyable on a daily basis. I would like to thank ODS Medical Inc. for providing me with data and support along the way. I would like to thank my brother who made the collaboration with ODS possible in the first place. Finally, I would like to thank my family, who have always supported and encouraged me to pursue a path in a academia. Much of what I have accomplished thus far would not have been possible without their unconditional love and support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Cancer is currently the number one cause of death worldwide. The World Health Organization estimates that in 2015, 8.8 million deaths were related to cancer and accounted for nearly 1 in 6 deaths globally [48]. In 2012, prostate cancer was the fourth most frequently diagnosed cancer, and accounted for approximately 8% of cancer diagnosis worldwide [21]. In 2015, 26% of all cancer cases diagnosed in males in the United States were related to prostates and resulted in over 27 000 deaths [54]. Cancer in the brain accounted for nearly 2% of cancer diagnoses worldwide [21]. In 2012, it was diagnosed at a rate of 250 000 annually worldwide and caused approximately 189 000 deaths [2]. Intensive amounts of research have been dedicated to find better treatments and improve survival and remission rates for those diagnosed with various forms of cancer.

In this thesis, we explore real-time cancer diagnosis of cells in-vivo in the human body using Raman spectroscopy, specifically in the human brain and prostate. This knowledge is of particular importance in the context of surgical tumor removal. When operating on human tissue to remove tumors, surgeons know the location of tumors prior to surgery using non-invasive imaging techniques. Despite the usefulness of the pre-operative imaging techniques, during surgery is is not always clear where the strict border of the tumour is versus where healthy tissue is. When removing the bulk of a brain tumor, for example, a surgeon currently has no way of knowing in real-time exactly where the cancer cell borders are and must instead rely solely on pre-operative images as a guide. This is of critical importance since the volume of residual cancer cells after surgery directly impacts survival rates [30]. Removal of healthy brain tissue can cause deterioration of basic motor skills (impaired memory, vision etc.). Visually, this distinction is almost impossible to make with the naked eye. This motivates the need for a diagnostics tool to be used by surgeons in-vivo in real-time during surgery, allowing surgeons to probe localized and specific regions of various tissues for cancer cells and aiding them in their decision making when removing tissue.

Raman spectroscopy for cancer detection in human tissue in-vivo is an emerging technology and has shown promise in previous studies [28] [29] [2] [26]. Raman scattering occurs when photons interact inelastically with molecules in a specific medium. The Raman spectrum of a tissue consists in the frequency response of the inelastic scattering of monochromatic light (generally from a laser) resulting from the interaction with vibrational modes in biological molecules. It has been shown that the subtle Raman spectrums can be used as a type of cellular fingerprint and that it is possible to classify pathology of tissues based on the response of the cells to incident light [26] [2] [28]. Advancement in

laser and computing technology allows affordable and real-time measurements to be made by surgeons during surgery and offer a non-invasive diagnostics tool.

We focus our study on data acquired from human prostates and brains. A handheld probe which consists of a near-infrared spectrum-stabilized laser operating at a wavelength of 785 nm capable of single-point submilimeter precision with a circular laser spot of radius 0.2 mm^2 was used for data collection. For the prostate dataset, Raman signals originating from prostates of 32 patients following radical prostatectomy are studied. For each patient, a number of samples are measured at various regions of interest on the affected tissue. Samples are measured directly on the prostates and histopathology results of the areas under study are used as labels for classification. The brain dataset comprises of signals measured from 12 distinct human brains in-vivo by a neurosurgeon during surgery. Raman spectrums were collected from various regions of interest during surgery and the same sampled locations were sent to histopathology for labelling. The equipment used for the brain dataset and setup is very similar to the setup used for the prostate dataset.

By working on the data provided by ODS Medical INC., we seek to provide a deeper insight in to the processing steps and machine-learning algorithms used to identify cancer in real-time, in-vivo. We focus our attention on the classification of already acquired signals in each dataset. Particularly, we are interested in studying the preprocessing steps involved prior to classification as well the strengths and drawbacks of different classifiers, such as support vector machines (SVM), multi-layer perceptrons (MLP) and convolutional neural networks (ConvNet). We are the first, as far as we know, to explore using ConvNets as a classifier on data acquired in-vivo in humans from Raman spectroscopy. We also explore the use of data reduction through autoencoders (AE) and principal component analysis (PCA) and the implications of data reduction. We study different cross-validation (CV) methods and metrics to evaluate the performance of the various classifiers and the various architectures they may have. We aim to also study the effectiveness of a leave-one-spectrum-out (LOSO) approach traditionally used in literature compared to other forms of cross-validation such as K-fold cross-validation and leave-one-patient-out (LOPO) schemes. We are particularly interested to understand how networks perform when they have not encountered signals from a patient in the dataset. One problem with our current Raman spectroscopy datasets is that they are very limited in quantity, due to the very nature of the procedure. We explore transfer learning and fine-tuning by using the information from one tissue and applying it to another and study the effect of data augmentation to seek ways of improving classification from limited datasets.

1.1 Contributions to the field

We focus our attention on data gathered from human brains and prostates in-vivo using Raman spectroscopy. Prior studies done in this field have shown that MLP could outperform boosted trees algorithms in brain cancer classification task with accuracy $\geq 90\%$ [29] [30]. Other studies have demonstrated that SVM combined with PCA could yield sensitivity and specificity results $\geq 85\%$ [62]. One major challenge in the classification task is the limited quantity of available data. We contribute to the field in the following ways:

- We show that preprocessing steps have a significant impact on classification results and present those that are best suited for real-time Raman spectroscopy

- We show that SVM and ConvNet consistently outperform MLP on our datasets
- We explore the use of data augmentation on our datasets and show negligible improvements
- We explore the use of dimensionality reduction using PCA and AE and show that while PCA is better suited than AE, classification results are optimal on the original feature space
- We explore transfer learning, and show that there is marginal improvement in using one organ to fine-tune the weights of a classifier on a different organ
- We show evidence of clustering within patients in our datasets suggesting that metrics acquired using LOSO in literature can be inherently biased and overly optimistic and that LOPO should be used to avoid bias

One important result is that classification metrics across different classifiers did not vary significantly. This suggests that the separability of the data is inherently limited by artifacts in the data such as noise, variability across patients etc. This suggests that future work should be concentrated on ensuring as little variability as possible in data gathering protocols across patients, and on more refined preprocessing steps. Obtaining more data is also critical since our datasets are currently limited in size and additional data could allow the training of more robust classifiers.

Chapter 2

Background Information

In this chapter, background information on fundamental theory related to Raman spectroscopy and classification techniques is provided to help understand the classification of cells using Raman spectrums in the human body. This chapter gives a high-level overview of the many concepts explored in this thesis and should be considered as a condensed summary of the relevant information needed to the reader prior to reading further in to the thesis. Sources are provided should the reader want to explore further on given topics.

2.1 Spectroscopy

Spectroscopy studies radiation as a function of its wavelength, and can be used to study all types of electromagnetic radiation [55]. A classic example, as demonstrated by Netwon in the 17th century, would be the use of a prism to separate a polychromatic source of lighting such as the light from the sun in to its different wavelength components [53]. This simple experiment demonstrates that white light is actually made up of a continuous spectrum of colors. In the 19th century, it was shown that colors from the sunlight were not distributed evenly. In fact, different sources of lighting exhibit different color distributions based on the nature of the source of light emitting them. Thus, light will scatter differently depending on the medium it originates from based on its atomic and molecular makeup. This guiding principle has allowed the study of atomic and molecular structures of microscopic objects. In this sense, spectroscopy has had a tremendous impact on the advancement of molecular physics, chemistry and molecular biology [16]. Infrared (IR) spectroscopy, for example, deals with the vibration of atoms and molecules in the IR region of the electromagnetic spectrum. By analyzing the absorption pattern as a function of wavelength, it is possible using IR spectroscopy to identify different chemicals and molecules that make up a material, since different molecules will have different resonant frequencies that will absorb at characteristic wavelengths. These resonant frequencies are generally those which match the vibrational frequencies of a molecule and are related directly or indirectly to their underlying structure. [50]

2.1.1 Raman spectroscopy

When electromagnetic radiation interacts with matter, the energy of the radiation $E = hv$, where h is Planck's constant and v is the frequency of the incident photon, can be

be absorbed, transmitted or scattered by the molecule it interacts with. If the photon collides elastically with the molecule, there is no energy transfer. The scattered photon has the same wavelength as the incident photon and the rotational and vibrational energy of molecules remain unchanged in this process. This process is known as Rayleigh Scattering. In the case of inelastic scattering, the incident photon may gain or lose energy based on the interaction with a molecule, which would itself lose or gain energy such as to not violate the conservation of energy laws of thermodynamics. For an incident photon of energy hv_0 and scattered energy hv_m, the change in energy absorbed or transmitted by the molecule, ΔE, will be:

$$\Delta E = hv_o - hv_m$$

This shift in energy, which will result in a shift in frequency (and thus in wavelength) of the scattered photon, is a process known as Raman Scattering and was first experimentally observed by Indian physicist Sir Chandrasekhara Venkata Raman in 1928. The changes in energy will appear at wavelenghts corresponding to the vibrational and rotational energy levels of electrons in the molecules the photons are interacting with. When $\Delta E > 0$, the molecule gains energy and the scattered photon has a longer wavelength (lower frequency and lower energy) than the incident photon, and when analyzed by a spectrometer, will give rise to what is known as a Stokes line or Stokes shift on the spectrogram. In the case where $\Delta E < 0$, the molecule looses energy and the scattered photon will have a shorter wavelength. This is known as an anti-Stokes shift and occurs when the molecule is not originally in its ground energy-state. When the photon interacts with the molecule, the molecule is brought to a higher, unstable energy state (a virtual state), and can then drop back to its ground energy state, scattering the photon with higher energy than it originally had. When considering vibrational quantum energy states for a molecule, a Raman transition can only shift by $\Delta v = \pm 1$ [13]. Figure 2.1 demonstrates a Jablonski diagram summarizing the different processes of elastic and inelastic scattering.

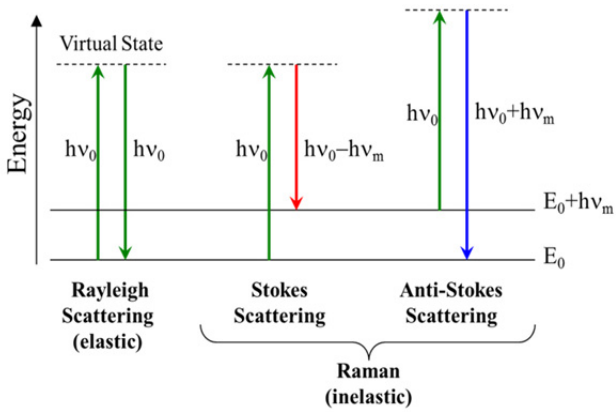


Figure 2.1: Example of elastic (Rayleigh) scattering, and inelastic (Raman) scattering with energy shifts. Image source [6]

Scattered photons due to Raman scattering can then be collected by a spectrometer and viewed as an intensity spectrum. The intensity will vary as a function of the Raman shift, i.e. the frequency change of the incident light, which is typically expressed in units of cm^{-1}. Since incident light is statistically likely to interact with all of the molecules present, the Raman spectrum collected will typically be representative of a combination of all of the

molecules' vibrational modes. Since each molecular type is unique in its set of vibrational modes, Raman spectroscopy can therefore serve as a tool to uniquely identify the presence of certain molecules. The Raman spectrum of each molecular type will generally consist of a series of peaks corresponding to the vibrational frequencies of those molecules [26]. Figure 2.2 gives an example of the molecular fingerprint of cholesterol, a prominent molecule in the human body. The characteristic Raman peak of cholesterol appears at around 1400 cm^{-1}, and corresponds to the CH_2 and CH_3 vibrational modes of the molecule. If a tissue under study consists in part of cholesterol, it would thus be possible to identify certain of its characteristic peaks.

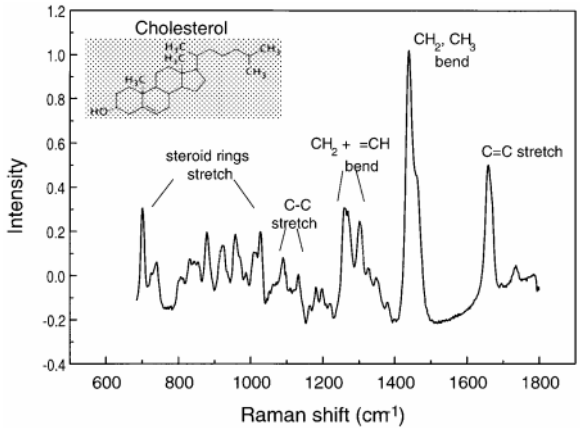


Figure 2.2: Example of a Raman spectrum for cholesterol. Different peaks correspond to the different excitation modes of chemical bonds in the molecule. Image source [26]

While Raman spectroscopy is useful as an identifier of single molecules by identifying prominent peaks, it can also be applied to biomedical problems such as the study of organic tissue. When considering a tissue, one can model the Raman response as a combination of the response of many complex molecules, and can provide insight into important biochemical changes due to diseases. If a disease causes changes in the molecular makeup of cells, Raman spectroscopy can be used to detect these subtle changes which might otherwise be invisible to microscopes. It is also possible to identify constituents of cells in tissue samples using Raman spectroscopy [26].

2.1.2 Fluorescence

Fluorescence is a radiative process in which the energy of a photon is absorbed by a molecule if the energy of the photon matches the energy difference between states of the electrons' energy levels in a molecule. Once in its excited state, the molecule can re-emit some of the energy as fluorescence. This process is different from Raman scattering in that the photon energy is completely absorbed by the molecule and not directly scattered. That is, fluorescence is a resonant process while Raman scattering is non-resonant. Raman scattering is an almost instantaneous process relative to fluorescence [65]. Autofluorescence is a process by which biological tissue emits light naturally through fluorescence and both autofluorescence and Raman scattering can occur simultaneously [46]. When considering Raman spectroscopy applications, it is thus important to have a meaningful way to separate between signals due to autofluorescence and Raman spectroscopy. The algorithms used to achieve this separation will be further explored in the Methodology chapter.

2.2 Classifiers

A classifier is a type of algorithm which allows classification of data into different categories [36]. For example, when looking at pictures of cats and dogs, a classifier might be tasked with distinguishing between 'cat' and 'dog' correctly. In the context of this thesis, classifiers are used to classify Raman spectrums basesd on their experimentally determined pathologies. There are many different types of classifiers that exist and choosing a specific classifier for a given task is not trivial. In the context of this thesis, three particular classifiers will be introduced and compared, Support Vector Machines (SVM), Multi-layer Perceptrons (MLP) and Convolutional Neural Networks (ConvNet). These aforementioned classifiers fall under the category of supervised machine learning, which means that ground-truth labels accompany all of the different spectrums to be classified. The classifiers are first provided with a training set of examples accompanied with labels to train on. The various classifiers' performances are then copmared using metrics evaluated on an independent labelled test set [36]. Validation methods and metrics will be discussed in the methodology chapter of this thesis.

2.2.1 Support Vector Machines

Support vector machines are used in supervised learning. The method consists of finding hyperplanes in a feature space that establish the optimal decision boundaries between classes, such that the margin of separation between classes is maximized [5]. In a simple 2-feature example, a line can be used as a separation boundary between classes. If the data is linearly separable, there exists a line that that maximizes the margin of error, with points directly on that line known as support vectors [5]. An example of this is shown in Figure 2.3.

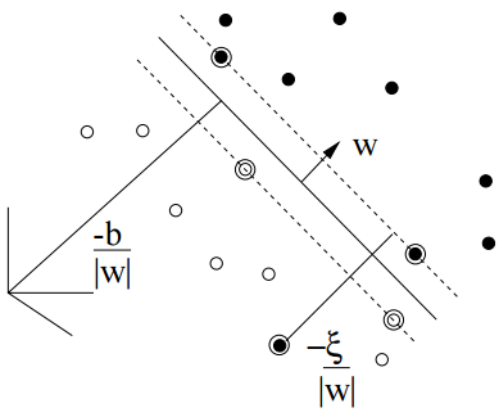


Figure 2.3: Example of a decision boundary for an SVM in a 2-feature space. The black and white dots correspond to 2 different classes, and the circled dots represent the support vectors on the optimal line. Image source [5]

The mathematical formulation for support vector machines is as follows: consider a 2-class classification problems with features x such that each sample x_i has an associated

label y_i such that $y_i \epsilon$ {-1,1} . The algorithm searches for an optimal set of weights w and a bias term b such that the distance between support vectors and the optimal hyperplan (the margin) is maximized. In a linearly separable case, this can be formulated as setting the constraints

$$\begin{aligned} \mathrm{x}_i \cdot w + b \geq 1 \text{ if } y_i = +1 \\ \mathrm{x}_i \cdot w + b \leq 1 \text{ if } y_i = -1 \end{aligned} \tag{2.1}$$

which is equivalent to solving

$$y_i(x_i \cdot w + b) - 1 \geq 1 \ \forall \ i \tag{2.2}$$

It can be shown using this formulation that the margin of separation is $1/\|w\|$. Thus, to maximize the separation between classes, it is necessary to minimize $\|w\|^2$. This can be achieved using unconstrained Lagrange multipliers, turning the problem in to a convex quadratic programming problem to allow to solve for non-linear cases [5]. In the case where the data itself is non-separable in the feature space given (i.e. if there is no hyperplane that can separate the data in the gieven feature space), soft margins are used to accept classification error as a possiblity and this is done by introducing positive slack variables [36]. It is also possible to map the data in to a transformed feature space and turn the original non-linear separation into a linearly separable space in the new feature space [36]. Once an SVM has been fitted to training data, the class of an unknown sample is determined by evaluating where it lies with respect to the optimal hyperplanes, which can be done by evaluating $sgn\ (w \cdot x + b)$.

SVMs have been shown to be a useful classifier for disease diagnosis using Raman spectroscopy. It has been used to diagnose and predict castration-resistant prostate cancer in patients with prostate cancer [62]. It was also used to diagnose and identify glycated hemoglobin levels in-vivo [61].

2.2.2 Multi-layer Perceptron

Multi-layer Perceptrons (MLP) are a set of classifiers which seek to map an input $\mathbf{x}$ to an output $\mathbf{y}$, which could be related by some function $f^*(\mathbf{x})$=$\mathbf{y}$, by approximating a function f such that $f(\mathbf{x};\theta) \approx \mathbf{y}$, where θ is a set of parameters. The goal of the MLP is to find an optimal solution to f by optimizing parameters θ by using a feedforward approach [24]. A feedforward approach maps all of the inputs $\mathbf{x}$, to the output $\mathbf{y}$, through one forward pass, but does not feedback the opposite way, from $\mathbf{y}$ to $\mathbf{x}$. MLP are a type of Artificial Neural Network (ANN), which have been inspired and modeled around the way axons in biological brains are connected [22]. Figure 2.4 shows an example of what an MLP might be sketched up to look like.

The hidden layer can be composed of multiple layers stacked in series, hence the name 'Multi-layer Perceptron'. Each layer can be considered as an independent function mapping, such that f(x) can be modeled as a series of functions g, h, i such that $f(x) = i(h(g(x)))$. The input, $\mathbf{x}$, is thus fed through the first layer, g, and iteratively through all layers until the final output layer which approximates a value for y. The intermediate layers are each referred to as hidden layers and each layer can have its own width which is directly related to its dimensionality. The number of hidden layers is referred to

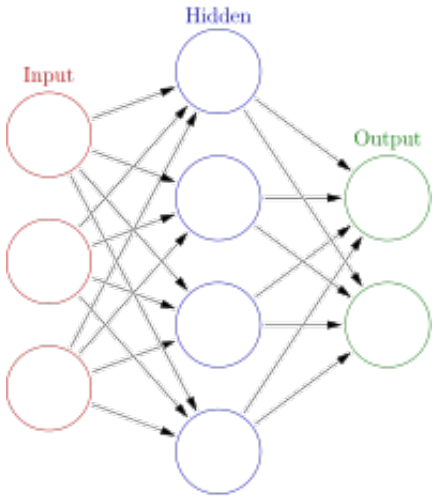


Figure 2.4: Example of a Feedforward Multi-Layer Perceptron. The input nodes are connected to the output nodes by a series of parameters and map a certain function f*. Image source [23]

as the depth of the network. An MLP is considered to be a fully-connected network since each node of every layer is connected to each node in the following layers.

For each hidden layer, a set of weights, w, and biases, b, are learned by the network in order to optimize its performance. Therefore, a layer $g(x; w, b)$ would learn a linear mapping such that $g(x) = x^T w + b$. It can be shown that when only linear mappings are used, any multi-layer perceptron could be reduced to a single layer perceptron. What makes multi-layer perceptrons especially useful is that they allow to insert non-linear functions ϕ in to our models, known as activation functions, between each layer. We can therefore represent each layer as a non-linear mapping $g(x) = \phi(x^T w + b)$. The collection of non-linear functions allows MLPs to approximate extremely non-linear functions [22]. There are many different models for activation functions. A commonly used activation function in MLP is the rectified linear activation function (ReLu) which is defined as

$$ReLu(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \tag{2.3}$$

MLPs are used in the context of supervised learning, such that each input x is accompanied by a target label y. A loss function $\mathbf{L}$ is defined as a measure to quantitatively assess how well a network estimates a function f^*, and is usually averaged over all of the training samples the network has seen such that $\mathbf{L} = \frac{1}{N}\sum_i L_i$ where N is the number of training examples seen by the network. The loss function is typically defined such as to make its derivative numerically easy to compute. The way to define the loss function for a particular problem is generally task-dependent. One type of loss commonly used for labeled data belonging to different categories is categorical cross-entropy. It is defined as

$$\mathbf{L} = -\sum_i y'_i \log y_i \tag{2.4}$$

where y_i is the predicted output of the network and y'_i is the ground truth. In the case where the labels are categorical, y'_i is typically represented as a one-hot vector of dimension 1xM

where M is the number of labels. Each position in y_i' corresponds to a specific class, and the network learns to predict the probability of an input x belonging to each class through its prediction y_i. Since the output y_i should be a probability distribution, it should sum to 1 and all values should be greater than zero. It should attribute the highest value to the most likely class. A softmax function σ is typically applied to the output layer of the MLP in order to achieve this, which is defined as

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{M} e^{z_k}} \ , \ j = 1, .., M \tag{2.5}$$

Once the loss function has been defined, the aim of the MLP is to minimize the loss function by finding optimal parameters to f. The network is trained with a series of mini-batches (sampled at random from the training set) of labeled training data. At the end of every mini-batch, the loss is calculated and weights are updated by an optimization method known as stochastic gradient-descent using the backpropagation algorithm. The process is said to be stochastic because of the random nature of the mini-batch selection. During backpropagation, the gradient of the loss function is calculated with respect to the parameters of the network and gradient-descent updates the weights and biases of the MLP at every iteration. The process is repeated until terminated by a user-defined criteria (usually when the loss of training and validation sets reach a minimum) [47].

Apart from the numerous parameters θ that need to be optimized in the network, there are a variety of hyper-parameters that can be tweaked to improve the performance of a neural network. For example, these hyper-parameters can include the number of hidden layers, the number of hidden units per layer, the activation function used, the weight initialization, etc. [3].

2.2.3 Convolutional Neural Networks

Convolutional neural networks (ConvNets) are a specific type of neural network and share very similar properties to MLPs. Most of the terminology and methods discussed in the MLP section are very similar to those associated to neural networks. The main difference between a ConvNet compared to an MLP is that an MLP is a fully-connected model, such that each node in a layer is connected to the nodes in the following layer through learned weights, whereas a ConvNet has filters of much smaller size that are convolved with each previous layer to make up the following layer. ConvNets have been ubiquitous in machine learning since 2012, when a ConvNet outperformed by a significant margin all other classifiers at the ILSVRC competition at the task of image recognition. A deep ConvNet, AlexNet, was used and consisted of 5 convolutional layers and 2 fully connected layers [37]. Since then, they have shown promise in many classification problems such as digit recognition, face recognition, speech recognition etc. [40].

For a 1-dimensional discrete signal, as is the case in Raman spectroscopy, a convolution is defined by equation 2.6:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \tag{2.6}$$

In equation 2.6, $h[n]$ is considered to be a filter of finite size. The goal of the ConvNet is to learn the weights associated to different kernels and feed that information forward through several layers in order to reach the targeted output. This is done through stochastic gradient descent as was dicsussed in the MLP section. In highly-correlated input spaces, such as the data obtained in Raman Spectrumxs samples, local correlations can be exploited to classifying signals more accurately by forcing the extraction of features to a local space and imposing locality to the field of hidden units [39].

2.3 Dimensionality Reduction

Dimensionality reduction is a means by which the feature space of some input space is reduced in a way that maintains important information from the samples under consideration. For example, if we consider N samples, each of respective size 1xM, dimensionality reduction would seek a transformation such that all N samples would be reduced to a dimension of 1xK such that $K < M$. We will explore how to transform a given signal in a different space to reduce its dimension. There are different ways in which dimensionality reduction can be achieved. One can employ feature selection, in which specific features are retained from an input space and others ignored. Subsampling would be a good example of feature selection. Another method, feature extraction, seeks to learn new representations of the original feature space through some transformation [8]. Dimensionality reduction can allow the use of simpler network architectures, involve the training of less weights and limit the potential for overfitting. It can also reduce run-time during classification and avoid fitting on noise models.

The advantage of using either PCA or AE for dimensionality reduction is that they are both unsupervised learning methods and can thus be trained on unlabeled data that can not be classified by histopathology. Histopathology is time-consuming and expensive compared to sampling measurements.

2.3.1 Principal Component Analysis

Principal Component Analysis (PCA) is a form of unsupervised feature extraction. It is considered unsupervised since it does not require any labeling of the data prior to feature extraction. PCA attempts to reduce the number of features representing a certain input while maximizing the variance of the dataset. This is done by searching for the dimensions, i.e. the principal components, which offer maximum variation of the data [51]. Mathematically, PCA can be defined as follows: considering a matrix $\mathbf{X}$ of zero-mean, with I rows and J columns, PCA seeks to represent the data from $\mathbf{X}$ to a linear projection $\mathbf{t_i}$ such that

$$\mathbf{t_i} = \mathbf{X}\mathbf{w_i} \tag{2.7}$$

Where $\mathbf{w_i}$ is a set of p weights such that each $\mathbf{t_i}$ is defined by its associated $\mathbf{w_i}$ and $\|\mathbf{w_i}\| = 1$. PCA seeks to find values for $\mathbf{w_i}$ such as to maximize the variance of $\mathbf{t_i}$, var($\mathbf{t_i}$) [4]. Mathematically, this can be formalized as trying to solve

$$\text{argmax var}(\mathbf{t_i}) = \text{argmax } \mathbf{t_i^T}\mathbf{t_i} \tag{2.8}$$

Using equation 2.7, we therefore need to solve

$$\text{argmax } \mathbf{w_i^T X^T X w_i} \tag{2.9}$$

It can be shown that the maximum value of 2.9 is obtained by setting $\mathbf{w_i}$ as the eigenvector corresponding to the largest eigenvalue of 2.9 [4]. By finding $\mathbf{t_i}$ where i=1, we can find the first principal component. Subtracting $\mathbf{t_i w_i}$ from $\mathbf{X}$ from the remaining values, we can iteratively find all the remaining components that maximize the variance successively. The amount of total components to use are problem-dependant and depend on the variance of the datasets considered. One main advantage of PCA is that it is a linear representation of the data and can sometimes facilitate data representation and visualization [60].

2.3.2 Autoencoders

An autoencoder is a neural network that learns how to output a copy of the input it was provided with. While the output of an autoencoder might be of little importance since the input is already known, it can be designed in an hour-glass shape such that the waist of the hour-glass becomes a condensed representation of the original input and contain as much information as possible from the original input. This is represented in Figure 2.5. A typical autoencoder passes the input x through a series of encoder functions f(x) and decoder functions g(x) while trying to minimize the loss L(x,g(f(x)). The output f(x) can be designed to have a much smaller dimensional space than x, leading to dimension reduction.

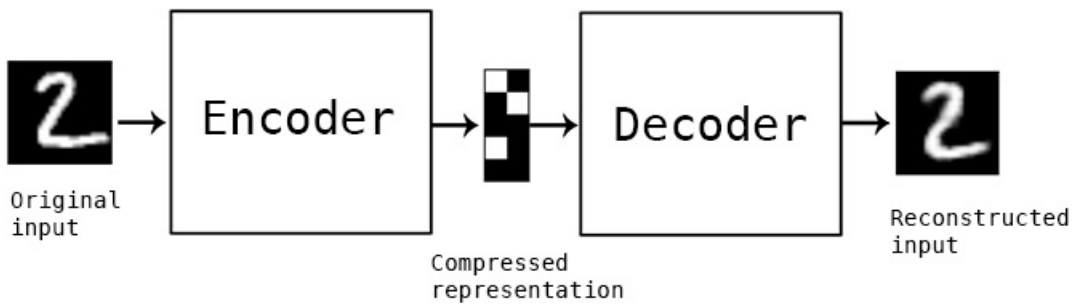


Figure 2.5: Example of an autoencoder. The original input is compressed by the encoder function to a representation of smaller dimension learned by the network. The decoder then reconstructs the image to minimize the reconstruction error. Image source [33]

It can be shown that when an autoencoder consists of a single layer, is linear and the loss function used is the mean-squared error, an autoencoder can be used to approximate PCA [24]. Autoencoders with nonlinear encoder functions and nonlinear decoders can learn more powerful nonlinear forms of PCA and can result in better dimension reduction representations [24].

2.3.3 Conclusion

We have presented in this chapter the core information needed to understand the fundamentals behind classification of Raman spectrums. We have explored the very nature of

Raman spectroscopy and the competing phenomena that are observed when measuring Raman spectrums. We have summarized the important theory behind classifiers that we will be using to classify our signals and the different dimensionality reduction algorithms that we will be using throughout this thesis. This section is meant to serve as a high-level overview of the different theories involved.

Chapter 3

Related Work

Raman spectroscopy as a diagnostics tool has been explored extensively in many previous works [28][2][26]. Research in the field has been facilitated and accelerated by the technological advancement of commercially available and powerful lasers and spectrometers over the past few decades [26]. The advancement of computing power has also lead the path to much more sophisticated statistical analysis tools allowing the processing and interpretation of complex and highly correlated data such as a Raman spectrums, paving the way to more efficient diagnostics. This section will highlight related works and studies that have been done in the past concerning Raman spectropscopy with a particular attention to diagnostics applications and its use in clinical settings, in-vivo and ex-vivo.

3.1 Raman spectroscopy and Disease detection

Many aspects of Raman spectroscopy make it an attractive solution as a real-time diagnostics tool. Firstly, it does not require exogenous contrast agents as would be required in other imaging techniques like in Magneic Resonance Imaging and PET-scans [28]. Advances in the technology in the instrumentation in Raman spectroscopy allow faster acquisition times (in the milliseconds) paving the way for real-time diagnostics [28] [2]. Advances in edge-computing devices allow fast processing of complex signals needed to discern different diseases. This is of particular use in in-vivo applications where surgeons might require real-time feedback to make important medical decisions. Spectrometers and lasers are small enough that they can be housed in mobile units and edge-computing devices are powerful enough to work offline, facilitating deployment in operating rooms. Handheld ergonomic probes adapted for Raman spectroscopy have already been tested and developed [30].

3.1.1 Brain

Brain cancers are diagnosed at a rate of 250 000 annually worldwide and cause approximately 189 000 deaths [2]. Current operating procedures involve targeting of the tumor using magnetic resonance imaging (MRI) and neurosurgical microscopes. Once located, a neurosurgeon then surgically removes the tumor from patients. One problem that typically arises in current procedures is that it is not possible to diagnose at a cellular level the boundaries of the tumor. While finding and removing the majority of the tumor is a well-established process, it is visually impossible to distinguish in real-time which neighboring

cells are cancerous or healthy with current pre-operative imaging modalities. This can lead to residual cancerous cells left behind after surgery which can in turn lead to relapse post-surgery and affect survival rates of patients. The caveat is that removing healthy brain-tissue unnecessarily can impact basic motor functions in patients' brains. It is thus necessary to find a means to diagnose cells in real-time in the brain for surgeons to guide their decision making process.

Extensive work has been done in brain cancer detection using Raman spectroscopy. Many studies have looked at specific applications in rodents and in ex-vivo brain tissue. Mizuno et al. have shown that grey and white matter could be differentiated in rats using near infrared Raman spectroscopy using a 1064 nm excitation wavelength [45]. This paved the way for studying brain tissue contents in-vivo using Raman spectroscopy by showing that common spectral bands could correspond to the tissue's molecular structure. Koljenovic et al. showed that adajcent brain structures could be differentiated using Raman spectroscopy in pig brains. They analyzed the spectrums from 7 sliced pig brains ex-vivo The pig brains were sliced and a 719 nm laser was used. The spectrums were reduced using PCA and hierarchical cluster analysis combined with a euclidean distance measurement were used to distinguish between various brain tissues [35]. In Kast et al. [31] it was shown that Raman spectroscopy could distinguish healthy grey matter and white matter from glioblastomas (cancer) in human brains ex-vivo. They used a 785 nm laser and analyzed spectrums of 40 frozen brain tissues with a particular focus on 3 bands of the Raman spectrum [31].

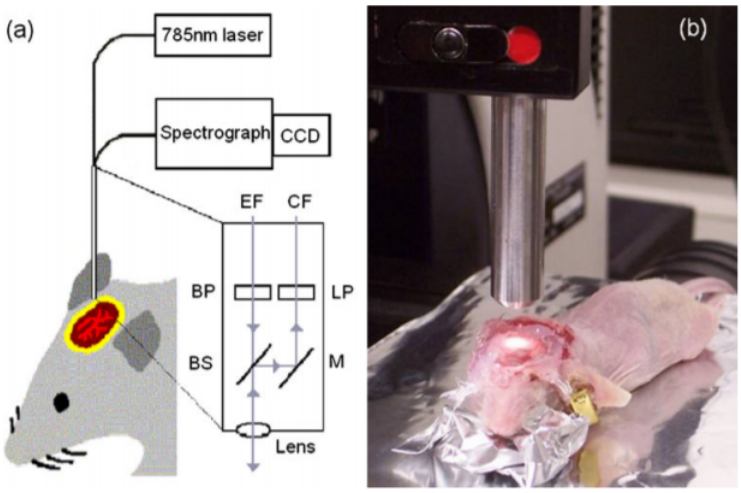


Figure 3.1: Setup used by Kirsch et al. to detect tumours in mice in-vivo. Image source: [34]

In Kirsch et al, it was shown as proof of concept that Raman spectroscopy could be used in-vivo in mice to detect brain tumors. A 785-nm laser was used to probe a square area of the brain of a mouse was probed. Using a K-means clustering algorithm, a map of the tumorous regions was drawn and overlayed on top of the corresponding brain structure and shown to correspond to the tumor location in the mice brain as shown in 3.2 [34]. Figure 3.1 shows the setup that was used in the study.

In Jermyn et al., an intra-operative Raman spectroscopy probe is used in-vivo to distinguish healthy cells from cancer cells, and reported a sensitivity and specificity of >90

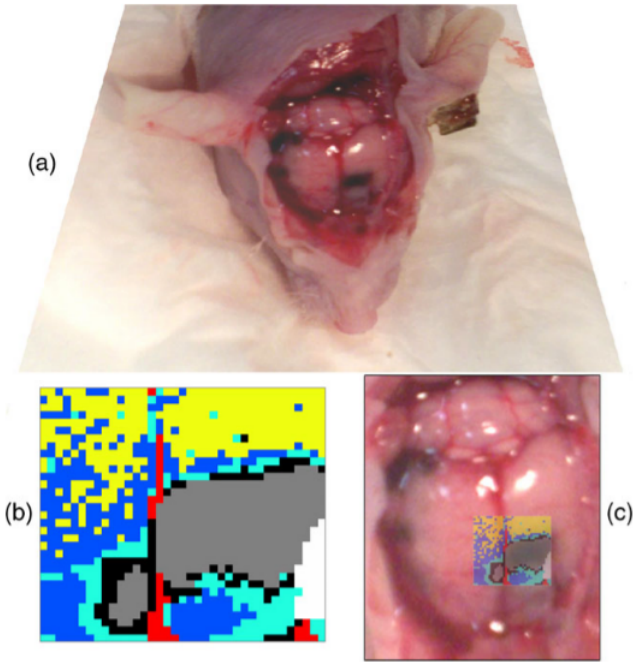


Figure 3.2: In figure (a), the mouse brain is shown and the tumor is visible to the naked eye. Raman spectrums are taken over the area of the brain and a color map is drawn using the k-means clustering algorithm to find the tumor (b). This is overlayed to the original image in (c). Image source: [34]

% using a leave-one-spectrum-out validation. The study was conducted on 17 patients with World Health Organization (WHO) grade 2-4 gliomas over 161 samples. The samples were classified in to three categories ; normal brain (no cancer cells present), normal brain infiltrated with invasive cancer cells ($<$ 90% cancer cells present) and dense cancer ($>$ 90% cancer cells present). A surgeon uses a handheld probe equipped with a Raman spectrometer to acquire signals in-vivo during surgery. Samples are sent to histopathology for labelling [30]. Figure 3.3 shows an example of the probe being used in humans in-vivo.

Using a boosted trees algorithm, they reported an AUC of 0.96 and an accuracy of $>$ 90% using a leave-one-specturm-out approach [30]. In a subsequent study on the same dataset, it was shown that an MLP performed much better than the boosted tree algorithm when taking light artifacts in to consideration from within the operating room. It was shown that using a leave-one-spectrum-out cross-validation (CV) technique, the MLP achieved accuracy, sensitivity and specificity of 90%, 91% and 89% respectively compared to boosted trees which achieved 71%, 84% and 51%. The boosted trees algorithm was severely impacted by ambient light effects and was shown to not be a suitable candidate as a classifier for intraoperative procedures. An MLP consisting of 2 layers was used with a sigmoidal activation function between the layers, shown in Figure 3.4 [29].

It was further shown in the same study that MLPs outperformed the boosted trees algorithm when classifying grey matter from white matter in calve brains acquired ex-vivo. 3 calf brains were analyzed and a total of 330 spectrums were collected. In this case, all spectrums from a given brain were used as a test set while the other spectrums were used as training sets. Once again, when taking ambient lights in to consideration, the MLP outperformed the boosted trees algorithm. The MLP achieved grey matter detection accuracy and white matter detection accuracy of 98% and 97% respectively, compared to

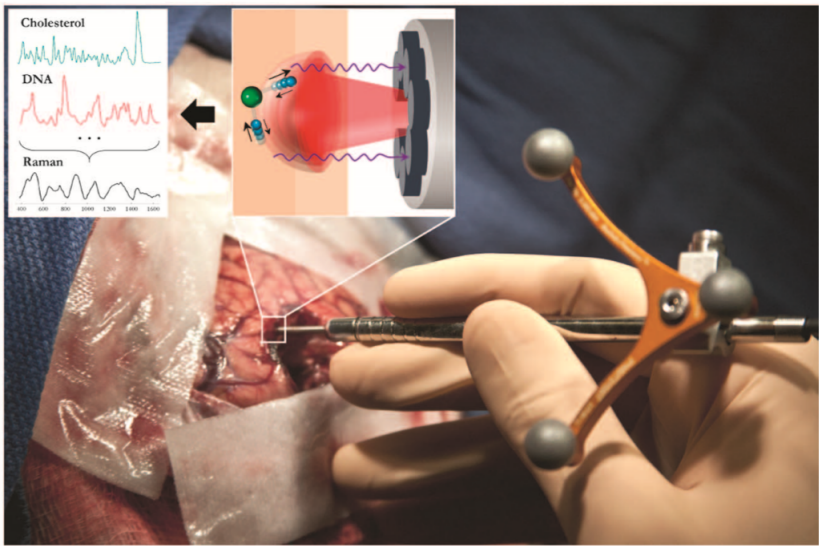


Figure 3.3: Example of the probe which can be used in real-time by surgeons. Image source [17]

89% and 81% for boosted trees.

3.1.2 Prostate

Prostate cancer is the most common cancer in males and the second deadliest cancer worldwide [9]. 1 in 10 cancer related deaths in males is a result of prostate cancer complications, and 1 in 6 males will be diagnosed with prostate cancer in their lifetime [54] [18]. The advent of prostate specific antigen (PSA) testing, a test allowing to measure the amount of PSA in blood, allows for earlier detection and treatment of prostates. There are various possible treatments for patients diagnosed with prostate cancer depending on a physician's risk assessment, including active surveillance, radiation therapy, hormone therapy, and surgery [56]. In the case that surgery is needed, one procedure available is radical prostatectomy (RP) [18]. In such a procedure, the entire prostate gland and seminal vesicles are removed. It is considered a minimally invasive approach, and it is approximated that 40% of RP are robot-assisted [18]. Despite being minimally invasive, RP is not without risk and can lead to erectyle dysfunction and urinary incontinence [63].

During RP, the surgeon seeks to remove the entirety of the prostate while preserving surrounding nerves and tissues. The prostate is then further studied ex-vivo by histopathologists to determine whether the cancer cells were confined to the prostate (negative positive margin) or whether cancer may have spread to surrounding tissue (positive surgical margin). Figure 3.5 illustrates the difference between the two scenarios. In the case of positive surgical margin, further compmlications are expected and can lead to patient relapse. The surgeon thus has to cut as much surrounding tissue as possible to ensure no cancer cells remain, however cutting sensitive surrounding tissue could lead to other complications like erectile dysfunction. The time between surgery and histopathology is considerable and the feedback is not instantaneous and can take from hours to days or weeks. Patients with a positive surgical margin have to therefore worry about additional treatment following RP. RP is believed to occur in 1 in 5 patients and is strongly correlated to disease recurrence. Positive surgical margin is associated to a 55% survival rate over 10 years. Histopathology

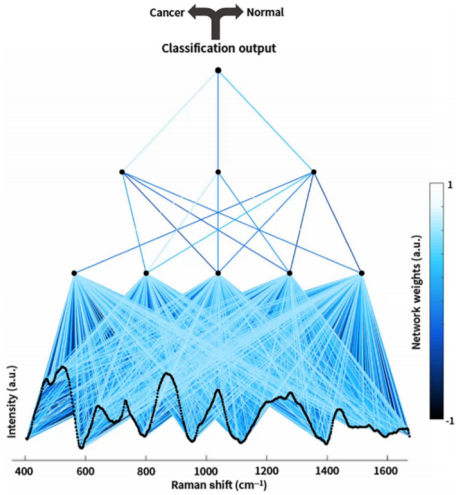


Figure 3.4: Example of the MLP used in Jermyn et al. in the study of cancer detection in human brains. Image source: [29]

is expensive and lengthy, and other imaging modalities, such as ultrasounds and MRI, do not provide adequate sensitivity and specificity for proper diagnosis of the cancer boundary. Raman spectroscopy has shown high potential for real-time in-vivo margin assessment in prostates, and is of particular interest because the technique is small enough to allow to it to be retrofitted to current surgically-assisting robots and does not cause any nerve damage [32].

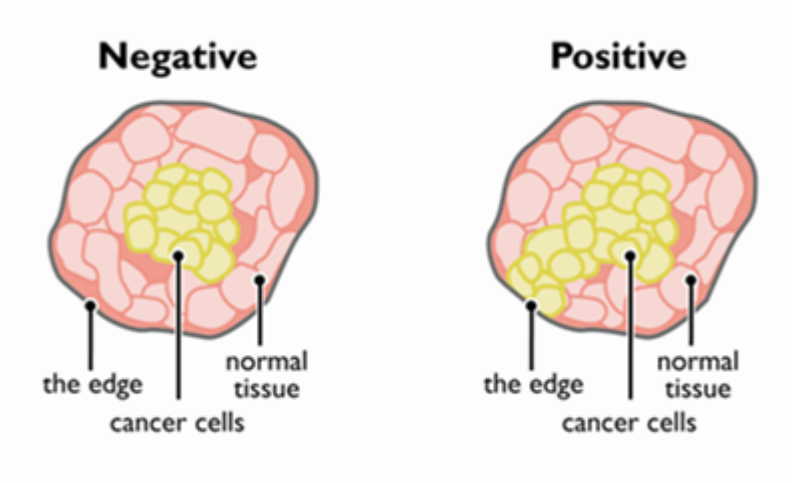


Figure 3.5: On the left, the cancer is completely removed (negative surgial margin). On the right, cancer is not localized completely within the prostate and cancer recurrence is very likely. Image source: [7]

Wang et al. have demonstrated that Raman spectroscopy can be used to diagnose castration-resistant prostate cancer ex-vivo. They show that by using a leave-one-spectrum-out cross-validation on 50 patients, they could achieve a sensitivity of 88.2% and specificity of 88.9%. They used a combination of PCA and SVM to achieve separation [62]. Crow et al. have shown that they were able to determine pathologies within the prostate by using a least-squares fit approach. They did this by analyzing the spectra of pure biochemical constituents. They used their method on 34 prostates for a total of 381 spectrums, and used

constituents such as cholesterol, DNA, and collagen. Their acquisition was done ex-vivo with acquisition times of 20 seconds. The acquired spectrums were then approximated as a sum of its biochemical constituents. They have shown that cancerous prostates show an increase in DNA content as the tissue progressed from normal to malignant, which is consistent with tumor diagnostics. They also showed an increase in cholesterol levels when evolving from benign to malignant [59].

3.1.3 Conclusion

This section highlights past and present state-of-the-art approaches in the field for using Raman spectroscopy as a real-time cancer diagnostics tool. We have highlighted the work done primarily on brain and prostates as this is the subject matter of this thesis. Advancement in technology behind lasers, spectrometers and computation power pave the way for real-time diagnostics. We seek to contribute to the field by presenting our results on data acquired on human brain and prostates.

Chapter 4

Methodology

This chapter outlines the methodology and procedures used in the classification of Raman spectrums sampled from human prostates in the context of this thesis work. The key steps involved are data collection, data processing and classification. The data collection involves the collection of signals ex-vivo directly from human prostates after radial prostectomy and in-vivo during surgery for brain, the design and implementation of a system to collect and store the relevant spectral data for later processing, as well as the classification by histopathology of the different samples collected to be used as ground truth for classification. Data processing involves all the pre-processing and post-processing steps taken before actual classification of the signal. This involves noise subtraction, machine calibration, autofluorescence removal, signal smoothing and normalization of the signals. Classification looks at the different means by which to classify the data by comparing the performance of various classifiers. This involves using different validation schemes to study the robustness of classifiers and to justify their use in real-time clinical settings.

4.1 Data Collection

Collecting the data in an efficient and methodical way is a critical first step. If the signals collected are corrupted by ambient factors this can make the classification problem more difficult. Another problem could be a low signal to noise ratio due to short exposure times, however considerations must also be taken to ensure that the system can operate in real-time and avoid longer exposure times. Health considerations also need to be taken into account such as to not affect tissues that are exposed to the laser. It is thus important that measurements be taken in a relatively small time frame and that the signal to noise ratio be kept high [28].

4.1.1 Handheld Raman spectroscopy Probe

The handheld probe used consists of a near-infrared spectrum-stabilized laser operating at a wavelength of 785 nm capable of single-point submilimeter precision with a circular laser spot of radius 0.2 mm^2. This corresponds to a tissue area of 0.5 mm diameter. The depth the laser can reach is of approximately 1 mm and the acquisition time is of 0.2 s [30]. Fiber optic cables relay the signal to a charged coupled device (CCD) array and the spectrum

can be collected and analyzed in real-time. The CCD used has a spectral resolution of 1.6 to 2.1 cm^{-1} and operates in the range of 381 to 1653 cm^{-1}. The Rayleigh scattering of the laser on the tissue can be several orders of magnitudes higher than the Raman scattering of the signal. In order to filter out most of the Rayleigh scattering as well as the inelastic scattering resulting from the instrument itself which both contaminate the Raman signals of interest, the probe was fitted with micrometer-scale in-line filters. A combination of a band-pass filter and low-pass filter were used to attenuate the signal resulting from the elastic scattering [30]. A schematic of the system used is shown in Figure 4.1.

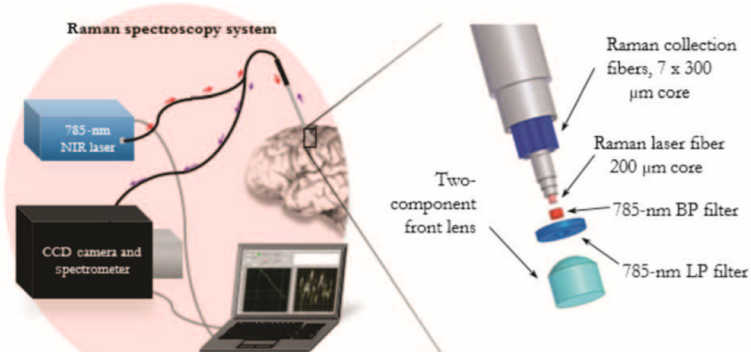


Figure 4.1: Schematic of the laser and CCD setup used for the collection of Raman spectrums. The 785-nm laser is gathered on to a CCD and the data is collected on a server for later processing. [30]

The laser is embodied in a solid frame (the probe) which can be handheld by a surgeon for real-time use. The probe uses 7x300μ m cores. Those, along with the lenses to converge the laser light and filters, are housed inside a 2.1 mm stainless steel needle tube for the surgeon to guide during measurements. Figure 4.2 shows the probe used by a surgeon during brain-tumor removal.

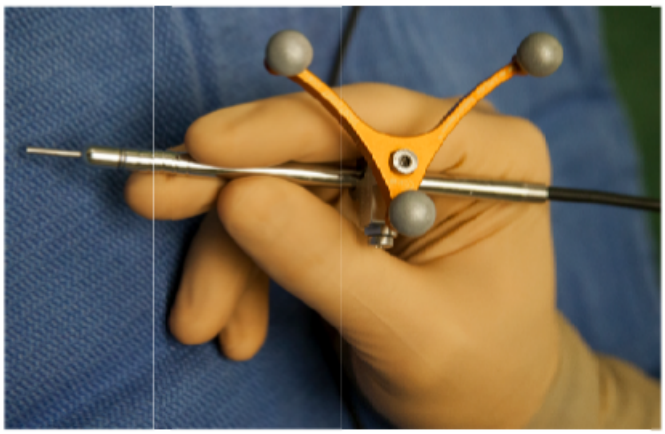


Figure 4.2: Example of the probe which can be used in real-time by surgeons. Image source [17]

4.1.2 Histopathology

Tissue sampling and labelling is a critical step in the classification of Raman spectrums. This process is especially difficult in the case of in-vivo applications since the ground-truth

labelling is usually done externally in a separate laboratory dedicated to histopathology assessment of tissue. This makes it difficult to correlate a measurement directly taken in-vivo to a measurement taken ex-vivo by the pathologists. Part of this difficulty arises from the size of the probe used and the error that can be attributed to the position of the probe and the subsequent position of the biopsy tool. It is also difficult to determine exactly how deep the laser penetrates within a given tissue and therefore how much sampling depth and volume should be removed from the tissue under consideration by pathologists. Another factor which is important to consider is the heterogeneity within tissue sample [28]. It is possible for a given measurement to contain signals from cells which are healthy and cancerous simultaneously. Since protocols in clinical pathology generally report the worst assessment of a conglomeration of cells, it is possible for a sample to come back labelled as cancerous even though it represents a very small fraction of the Raman Signal, and could be considered a false negative by the classifier. With respect to gathering labelled data for a real-time Raman spectrum classifier, one alternative is to make ex-vivo measurements on fresh tissue samples within a few hours of removal [28].

The tissues sent to pathology are classified into different categories based on the tissue under examination. For example, when considering prostate tissue, a Gleason score is usually given for a sample based on the assessment done by the pathologist. Scores indicate the severity of the detected cancer and are used as a referential prognosis. Gleason scores can range between 2 and 10. Lower Gleason scores are associated with a healthier prognosis [25].

4.2 Data Processing

Data processing is critical when analyzing Raman spectrums. This is mainly due to the fact that the Raman portion of the signal is generally several orders of magnitude smaller than other portions of the signal such as autofluorescence and elastic scattering. It is also necessary to ensure that all the equipment used is calibrated regularly. Environmental factors, such as ambient lights, temperature, etc. as well as instrument-dependent artifacts can lead to different results when comparing spectrums [17] [11]. It is particularly necessary to have a baseline accross instruments in the case of deploying this system across many operation rooms. This section will outline the steps taken with regards to data processing prior to the classification of signals. This includes machine calibration, autofluorescence estimation and removal as well as signal smoothing and rescaling. A sketch of the pipeline used is represented in Figure 4.3. All data processing was done using MatLab and Python.

4.2.1 Machine Calibration

Before taking any measure, it is imperative to calibrate the spectrometer according to the guidelines provided by the National Institute of Science and Technology (NIST). A measurement of acetaminophen is used to calibrate the x-axis of the spectrometers as described in [17]. The relative intensity correction is then performed to correct for the system response using a measurement of a NIST 2241 Standard Reference Material (SRM). The SRMs have the property that they emit a broadband (200 $cm^{-}1$ to 4000 $cm^{-}1$) luminescence pattern when shined on by the Raman excitation laser. Their response should be smooth, photostable and have low absorbance in the range of the exciation laser to avoid

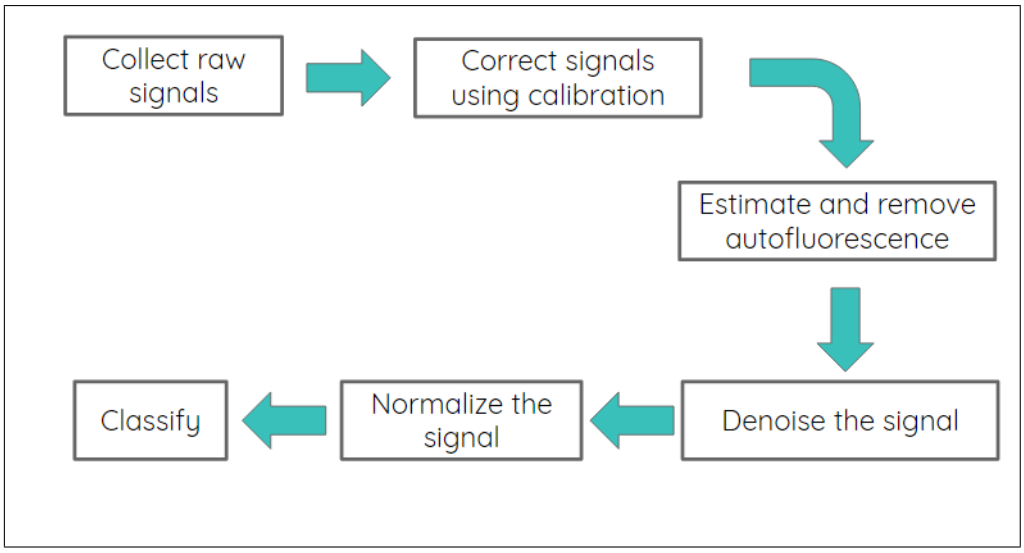


Figure 4.3: General outline of the data processing pipeline used for the processing of Raman signals prior to classification.

temperature dependent responses. The spectrum obtained with the SRM on our machine is used as a baseline and compared to NIST standards to correct the impulse response of the device which could vary based on temperature, lighting conditions, etc. The baseline is corrected such that the curve generated by our spectrometer be identical to the curve proposed by the polynomial fit proposed by NIST by satisfying equation 4.1 [NIST]

$$C(\Delta\nu) = I(\Delta\nu)/S_{bench}(\Delta\nu) \tag{4.1}$$

where $C(\Delta\nu)$ is the correction curve, $S_{bench}(\Delta\nu)$ is the raw signal obtained from the benchmarked material and $I(\Delta\nu)$ is the known benchmarked polynomial response provided by NIST. Formally, $I(\Delta\nu)$ is defined as

$$I(\Delta\nu) = A_0 + A_1(\Delta\nu)^1 + A_2(\Delta\nu)^2 + ...A_5(\Delta\nu)^5 \tag{4.2}$$

where $A_0...A_5$ are defined by NIST in [11]. All subsequently measured Raman spectrums $S_{meas}(\Delta\nu)$ are then intensity corrected using the correction curve obtained in 4.1 such that the signals obey equation 4.3

$$S_{corr}(\Delta\nu) = C(\Delta\nu) \cdot S_{meas}(\Delta\nu) \tag{4.3}$$

4.2.2 Data Acquisition

When acquiring the data, the probe collects multiple samples in each region, or samples from multiple regions. Multiple Raman spectrums are recorded with an integration time of 0.05 s and are averaged to reduce noise. A measurement is also taken with the laser off to use as a background noise model at every location. The subsequent measurement used consists of the mean of all signals with the laser on minus the background measurement to correct for ambient factors.

4.2.3 Autofluorescence estimation and removal

Autofluorescence is the process by which biological tissue will emit fluorescence as a result of ambient light sources. When analyzing Raman spectrums to classify cancer cells in the context of this thesis, we want to focus primarily on the Raman response of cancer cells. We thus need to remove the autofluorescence from the signal in a systematic manner across spectrums. Since fluorescence is typically orders of magnitude higher than autofluorescence, a typical approach consists of fitting a model to the background and removing it from the original signal such that the more subtle Raman variations are emphasized. We present in this section algorithms which are used for autofluorescence removal prior to classification.

4.2.4 Zhang fit

The first method explored, which we will refer to as the "Zhang Fit", is based on the method proposed by Zhang et al. in [64]. Consider an original signal $\mathbf{x_i}$ of length 1xm. We are seeking to approximate an autofluorescence signal, $\mathbf{z_i}$, also of length 1xm, such that the Raman portion of the signal, $\mathbf{r_i}$, can be approximated as:

$$\mathbf{r_i} = \mathbf{x_i} - \mathbf{z_i} \tag{4.4}$$

The method consists of seeking for a $\mathbf{z_i}$ which minimizes a cost function, Q, which takes in to account the fidelity of $\mathbf{z_i}$ to $\mathbf{x_i}$ and the roughness of $\mathbf{z_i}$. The fidelity of the signal, S, is defined as the sum of squared error of $\mathbf{z_i}$ and $\mathbf{x_i}$ such that:

$$S = \sum_{i=1}^{m} (\mathbf{x_i} - \mathbf{z_i})^2 \tag{4.5}$$

The roughness of the estimated signal, R, seeks to penalize sharp increases in $\mathbf{z_i}$, and can be thought of as a regularizer that forces $\mathbf{z_i}$ to adopt a smooth behaviour by penalizing curvature. R can be expressed as:

$$R = \sum_{i=2}^{m-1} (\mathbf{z_i} - \mathbf{z_{i-1}})^2 \tag{4.6}$$

Combining equations 4.5 and 4.6, and introducing a parameter λ, we can write Q as:

$$Q = S + \lambda R = ||\mathbf{x_i} - \mathbf{z_i}||^2 + \lambda ||\mathbf{D}\mathbf{z_i}||^2 \tag{4.7}$$

λ can be user-defined and can be thought of as how much the fit should prioritize fidelity to the original signal as opposed to smoothness. $\mathbf{D}$ acts as the derivative to the identity matrix such that D can be represented as:

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & 0 & ... & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & ... & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & ... & 0 \\ 0 & ... & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & ... & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & ... & 0 & -1 & 1 \end{bmatrix} \tag{4.8}$$

Since we are looking to minimize Q in 4.7, we are looking to find a solution to

$$\frac{\partial Q}{\partial z_i} = 0 \tag{4.9}$$

Applying 4.9 to 4.7, we get

$$(\mathbf{I} + \lambda \mathbf{D}\mathbf{D}')\mathbf{z_i} = \mathbf{x_i} \tag{4.10}$$

where we consider $\mathbf{I}$ to be the identity matrix. Thus, to find $\mathbf{z_i}$, we simply need to solve

$$\mathbf{z_i} = (\mathbf{I} + \lambda \mathbf{D}\mathbf{D}')^{-1}\mathbf{x_i} \tag{4.11}$$

We apply this method iteratively, such that after a first iteration from solving 4.11, we define a binary mask vector $\mathbf{w_i}$, such that values of $\mathbf{w_i}$ are either 0 or 1. We use a rule that

$$\mathbf{w_i} = \begin{cases} 1 & \text{if } \mathbf{x_i} < \mathbf{z_i} \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

and define a matrix $\mathbf{W}$ which has $\mathbf{w_i}$ along its main diagonal. We then redefine the fidelity parameter S, to consider the binary mask such that

$$S = \sum_{i=1}^{m} \mathbf{w_i}(\mathbf{x_i} - \mathbf{z_i})^2 \tag{4.13}$$

This leads to a new expression for Q, and therefore solving for $\frac{\partial Q}{\partial z_i} = 0$ now yields

$$\mathbf{z_i} = (\mathbf{W} + \lambda \mathbf{D}\mathbf{D}')^{-1}\mathbf{W}\mathbf{x_i} \tag{4.14}$$

Note from 4.14 that the λ term can also be user-defined and can be different from the λ value in 4.7.

Let us look at an example of this autofluorescence removal in practice. Figure 4.4 shows an example of a Raman Spectrum after being corrected using the NIST reference. We use the first iteration of the Zhang algorithm that we have described and find the relevant binary mask.

We then use the values from the binary mask to re-approximate $\mathbf{z_i}$ to obtain our autofluorescence approximation. Finally, we subtract $\mathbf{z_i}$ from $\mathbf{x_i}$ and recuperate the Raman portion of the signal. This is shown in Figure 4.5.

One signficant drawback from the Zhang method is that it requires calculating the inverse of a large matrix at every iteration. Compared to the I-Mod-Poly method, the Zhang method is considerably slower. Seeing as we seek to implement our system in real-time, the Zhang method is therefore not an ideal candidate compared to the I-Mod-Poly method which will be explored further in the next section.

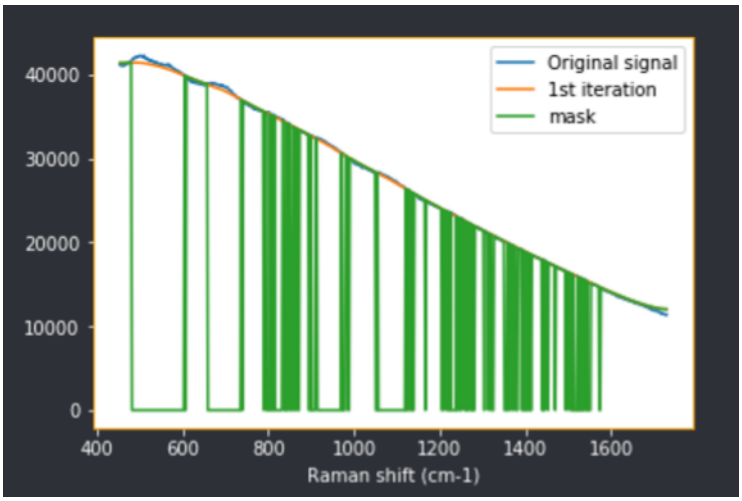


Figure 4.4: First iteration of Zhang AF removal with a binary mask

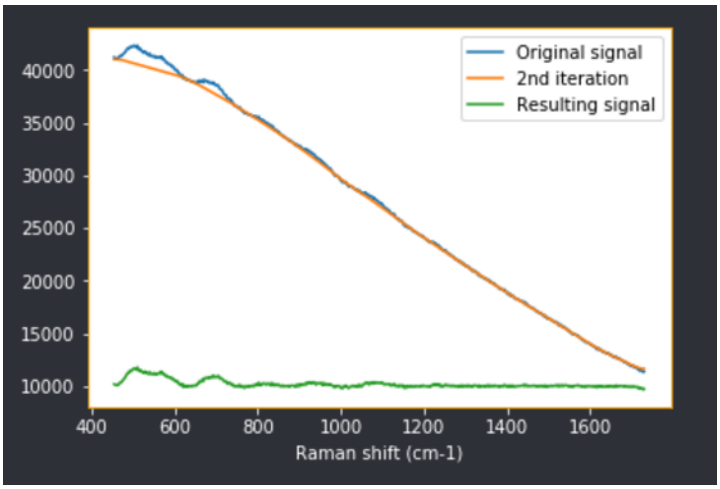


Figure 4.5: Second iteration of Zhang AF removal and the resulting signal

4.2.5 IModPoly

One method widely used in biological applications is the polynomial fitting (polyfit) method [65]. It is both relatively simple to implement and has been shown to be an effective means of autofluorescence removal. Simply put, the polyfit method seeks to fit a polynomial p_i of degree n to the original signal x_i and removes that polynomial from the original signal to remove autofluorescence and be left only with an approximation of the Raman signal, z_i, such that

$$p_i(\nu) = a_0 + a_1(\nu)^1 + a_2(\nu)^2 + ...a_n(\nu)^n \tag{4.15}$$

$$z_i = x_i - p_i \tag{4.16}$$

The polyfit method, while attractive in its simplicity and efficiency, has some drawbacks. It is dependent both on the spectral range $\Delta\nu$ and on the degree of the polynomial n. An example of this difference is shown in Figure 4.6, where the order and the range of the polynomial is varied for the same spectrum.

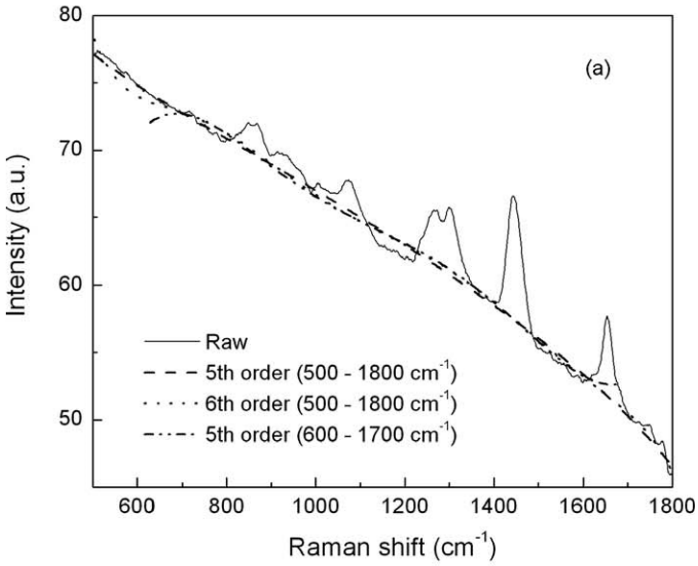


Figure 4.6: Example of how the polyfit method can disagree based on degree order of the polynomial (n) and spectral range $\Delta\nu$. Image source: [42]

An alternative to the polyfit method, modpoly, is proposed in [43]. Modpoly is an iterative method which compares the polynomial fit values to the original spectrum values and selects the lower values between the two, and refits a polynomial to the new concatenated signal. While this is an improvement upon the original polyfit method, the ModPoly method does not deal with noise in any manner, can introduce artificial peaks that are falsely detected as Raman peaks and could take many iterations before converging to an appropriate fit which is impractical for real-time spectroscopy [42].

Improved modpoly (IModPoly) is an adaptation of the polyfit and modpoly fit methods which seeks to address the noise issue by modelling it on the standard deviation of the difference between the polyfit and the original signal. To do so, a Residual, $R(\nu)$ is, defined as

$$R(\nu) = O(\nu) - P(\nu) \tag{4.17}$$

where $O(\nu)$ is the original signal and $P(\nu)$ is the polyfit computed. The standard deviation of $R(\nu)$ is then computed such that

$$DEV = std(R(\nu)) = \sqrt{\sum_{i}^{N} \frac{R(\nu_i) - \overline{R}}{N}} \tag{4.18}$$

where $\overline{R}$ is the mean of the Residual and DEV is a measure quantifying the noise levels in the spectrums. Thus, for a region to be considered a peak, it must be greater than the standard deviation of the overall signal and the fit at a current point. The process is repeated iteratively and is stopped by some user-based criteria, which could be either the number of iterations or an insignificant change from one fit to the next. The entire IModPoly process is outlined in Figure 4.7.

4.2.6 Savitsky-Golay Filtering

When dealing with any kind of device to record real-world observations, noise inherent from the recording device is almost always present in the measurements. It is generally recommended to use long time-averaging of a signal to suppress noise [12]. However, in the context of real-time Raman spectroscopy, time is a limiting factor and other noise-suppression methods should be considered. While many steps can be taken to mitigate the noise effects and increase the signal to noise ratio, it is necessary in most signal processing applications to remove noise digitally through the use of various filtering techniques by trying to reduce noise while preserving the signal of interest [38]. This is usually done by convolving the input signal $x[n]$, with a filter $h[n]$, such that the smoothed signal, $y[n]$, is related by equation 4.19:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \tag{4.19}$$

Other methods, such as filtering out high-frequency components, have also been suggested. The following section will discuss methods of noise reduction and its application in Raman spectroscopy. A widely used method for smoothing Raman signals is Savitsky-Golay filtering. A polynomial of degree d is fitted across $2m+1$ points centered about a given signal measurement $x[i]$. The method for solving the polynomial fit is the least-squared method. In their original paper, Savistky and Golay show that by using appropriate coefficients which can be derived analytically, the exact solution to the least square approximation for polynomials of low-degrees can be obtained [52]. In equation 4.20, the coefficients C_k can be looked up from existing tables and used to smooth Raman signals. This allows for a very fast and robust implementation of the smoothing signal.

$$y[n] = \sum_{k=-m}^{m} x[n+k] \cdot C_k \tag{4.20}$$

The amount of points used, as well as the degree of the polynomial fitted are user-defined settings and are application-dependent.

4.2.7 Normalization

The signals are then standardized and normalized using the standard-normal-variate scheme such that the condition in equation 4.21 is satisfied. This is a critical step so that classifiers can be trained on signals of the same orders of magnitudes and such that the non-linear activation functions used for classification operate in their intended ranges.

$$S_{SNV}(\Delta\nu) = (S_{meas}(\Delta\nu) - \overline{S_{meas}})/\sqrt{var(S_{meas})} \tag{4.21}$$

4.3 Classification

In this work, we seek to evaluate the performance of various classifiers on Raman spectrums. We focus particularly on three classifiers: support vector machines (SVM), multi-layer perceptrons (MLP) and convolutional neural networks (ConvNet). Each of these are

supervised learning algorithms and are trained on the available labelled data. In the case of cancer classification in humans, access to data is often the limiting factor and this is especially true in Raman spectroscopy. Since we are dealing with smaller datasets, we must use metrics and algorithms that allow us to compare the performance of our classifiers in an unambiguous and clear way. For example, considering only accuracy as an only metric of evaluation might be a poor choice when dealing with an unbalanced dataset, since simply outputting one classification result over another might result in very high accuracy but very poor disease detection. It is therefore important to monitor other metrics and find more robust means of evaluations which consider unbalanced datasets, and take into account the importance of sensitivity and specificity of a classifier. We introduce in this section different cross-validation methods and their implicatoins, as well as useful metrics that will be used to compare the classifiers in the next sections.

4.3.1 Cross-Validation

Cross-validation methods are a means of assessing how well a certain classification model might generalize when applying it to a new, independent data set. Cross-validation is especially important in the context of predictive analysis and assesing how accurate or precise the model truly is. cross-validation can be used for both model assessment and model selection. In the case of model assessment, one seeks to identify which models work best, by comparing for example the performance of different classifiers. Model assessment seeks for the best parameters that will make a given classifier optimal [20].

The data is generally split into subsets following user-defined and case-specific rules. Subsets of the data are left out once for testing and the remaining data is separated into a training and validation set. A set percentage of each training set is used as a validation set on every iteration. During training of the classifier, only the training data is used by the classifier. Upon each iteration of training, the classifier is deployed on the validation set and an evaluation metric such as a loss function is evaluated and recorded. The model is trained for a certain number of iterations until the evaluation metric reaches a minimum on the validation set. The model is then evaluated on the test set and the performance of the model is recorded.

4.3.2 K-fold Cross-Validation

The data is split into K random, evenly distributed, unique and non-overlapping subsets, spanning the entire dataset. The distribution of labels is thus similar in the K individual subsets compared to the origial dataset. K-fold CV has the advantage that each sample is evaluated once as part of a testing set. This process is repeated K times and the performance over each test set is compared. When comparing classifiers, it is important to set the randomness as 'fixed', or as pseudo-random. This means that the distribution is chosen at random once, but the same distribution is then used across classifiers to ensure a fair validation process.The results from the separated K folds are then interpolated and used as a metric to determine the performance of the classifier.

4.3.3 Leave-One-Patient-Out

A leave-one-patient-out (LOPO) Validation scheme is also used and compared to K-fold cross-validation. The datasets are labelled according to a unique identifier (UID) and it is therefore possible to split up samples per patient. Each patient is left out once for testing, and the remaining patients are used as training and testing sets. There are no fixed protocols while retrieving patient data and it is therefore possible that one patient have only one class of label associated to his name. Drastically different results to K-fold validation might indicate clustering of the data within patients and suggest that the models might be learning noise models as opposed to qualitative features to discriminate between hisotpathologies.

4.3.4 Leave-One-Spectrum-Out

A leave-one-spectrum-out (LOSO) approach leaves an individual spectrum out and uses the remaining data as training and validation sets. Since there can be many samples, it can be time-consuming and computationally expensive to evaluate every spectrum. LOSO can be done on a representative subset of samples and statistics can then be generalized. One drawback of LOSO is that similar samples (i.e. from a same patient in taken from a similar location) might give the system a bias towards a solution which it otherwise might not be able to discern (in the case of a never-before-seen patient, for example). It is also possible tor LOSO to model noise models as opposed to useful information.

4.3.5 Confusion Matrix

Consider a classifier that takes a sample as input x_i and returns one of a possible N diagnoses y_i for a given situation associated to a ground truth label l_i. We can build a matrix, $\mathbf{M}$, initialized as an NxN matrix of zeros, such that for each x_i, the element $\mathbf{M}[l_i, y_i]$ is incremented by one unit. After iterating through every sample, the resulting $\mathbf{M}$ will be what is known as a confusion matrix [15]. It maps the relationship between ground truth labels and predictions. Let us consider a binary classification problem in which our classifier must distinguish between 2 classes, one consisting of healthy specimens, h, and infected specimens, i. We consider a diagnosis as negative when the classifier returns a healthy diagnostic and positive when the classifier returns an infected diagnosis. There are four possible outcomes to the classification:

- The system classifies a healthy specimen (negative) as healthy (negative). This is considered a True Negative (TN)
- The system classifies a healthy specimen (negative) as infected (positive). This is considered a False Positive (FP)
- The system classifies an infected specimen (positive) as infected (positive). This is considered a True Positive (TP)
- The system classifies an infected specimen (positive) as healthy (negative). This is considered a False Negative (FN)

The resulting 2x2 confusion matrix is shown in Table 4.1. Note that $TP + FN + FP + FN$ is equal to the number of samples diagnosed. The confusion matrix helps understand trends in the classification process and can help identify strenghts and weaknesses of our classifier.

Table 4.1: Example of a confusion matrix for a 2-class binary classification problem.

		Prediction	
		Healthy	Infected
Ground Truth	Healthy	TN	FP
	Infected	FN	TP

4.3.6 Evaluation Metrics

There are different metrics to evaluate how well a classifier performs on a specific dataset.

- Accuracy (ACC), measures the proportion of samples that we're properly diagnosed, both positive and negative. This is calculated from the confusion matrix as:

$$ACC = \frac{(TP + TN)}{(Positives + Negatives)} = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{4.22}$$

- Sensitivity, or the true positive rate (TPR), measures the proportion of samples that we're diagnosed as positive (infected) that are ground-truth positive (infected). This is calculated from the confusion matrix as:

$$TPR = \frac{TP}{Positives} = \frac{TP}{(TP + FN)} \tag{4.23}$$

- Specitivity (SPC), measures the proportion of samples that we're diagnosed as negative (healthy) that are ground-truth negative (healthy). This is calculated from the confusion matrix as:

$$SPC = \frac{TN}{Negatives} = \frac{TN}{(TN + FP)} \tag{4.24}$$

- False positive rate (FPR), measures the proportion of negative samples that we're diagnosed as positive. This is calculated from the confusion matrix as:

$$FPR = \frac{FP}{Negatives} = \frac{FP}{(TN + FP)} = 1 - SPC \tag{4.25}$$

4.3.7 Receiver Operating Characteristic

A receiever operating characteristic (ROC) curve helps combine the evaluation metrics explained in the previous section into a visual and quantifiable means of analysis in order to compare the performance of classifiers based on different criterias and trade-offs [19]. ROC curves are used extensively in the biomedical field to assess the performance of classifiers,

specifically because of their utility in unbalanced classes [19] [1]. ROC curves combine sensitivity and specificity of a classifier as a single point on a graph. These metrics are attractive in the medical field since high sensitivity is useful to rule out disease and high specificity is useful to diagnose disease [1]. ROC curves are generated by plotting the TPR (y-axis) as a function of the FPR (x-axis). Figure 4.8 shows an example of ROC points plotted using different classifiers.

The dashed line in Figure 4.8 represents the expected performance of a classifier built on random performance. Points lying northwest of the dotted line are generally considered to perform better [19]. A classifier at point (0,0) would represent a classifier outputting a negative result regardless of the input, whereas a point at (1,1) would represent classifiers always outputting a positive result.

Classifiers like neural networks and SVMs can be architectured to output a real-valued number, between 0 and 1, representing the confidence of the network that a certain sample belongs to one class or another. While these numbers can in some cases be strict probabilities, they can also represent uncalibrated scores where a higher score means a higher probability of belonging to a class. In the case where outputs correspond to probabilities, it is natural to set a threshold T=0.5 such that a result is negative if the predicted value is smaller than T and positive if the predicted value is larger than T. However, a threshold T=0.5 might be a sub-optimal decision threshold in the case of uncalibrated scores [19]. It is therefore useful, for a given classifier, to evaluate the TPR and FPR at various thresholds T, and plot the resulting pairs of FPR and TPR points for a single classifier. Of course, by varying T in the range [0,1], we guarantee that (TPR,FPR) for T=0 will be (0,0) and for T=1 (1,1). What we are interested in is how the (TPR,FPR) trend varies between these points. As a rule of thumb, if the ROC curve tends more towards the point (1,0), its performance is considered better than other classifiers.

Figure 4.9 shows an example in which two algorithms are compared on a given binary classification task. The area under the curve (AUC) of each ROC curve of each of the algorithms is calculated and used as a metric to quantify which algorithm performed best. An AUC of 0.5 corresponds to a classifier built on random guessing, and an AUC of 1 corresponds to a perfect classifier. Typically, a higher AUC score is considered as a better performance, however, as noted in [1], the AUC gives equal importance to FPR and TPR when in practice one metric might be of higher value than another. In the case of K-fold cross-validation, we generate K ROC curves and linearly interpolate through the K curves to obtain a mean ROC curve and compute the AUC of the curve [27].

4.3.8 Data Augmentation

Some classification algorithms require large amounts of data for training. Neural networks are known to be particularly data intensive. Data augmentation consists in inflating a dataset artificially by exploiting transformations of the data in a manner that doesn't affect the labelling of the data. For example, transformations such as cropping and translations can be used on images to enhance a dataset [37]. Another form of data-augmentation can be the injection of noise in to the original data [41]. Since Raman signals are inherently weak to begin with, adding noise to the original signal would unlikely cause labels to change and might increase the classification performance. One means of data augmentation that is explored is the use of different averaging schemes. When Raman signals are collected, multiple spectrums are collected and averaged to reduce noise. The classic

averaging scheme consists of equally weighting all of the spectrums, such that if there are N spectrums recorded per position, each individual spectrum would contribute to $1/N$ of the final spectrum, such that sample x_i would be given by:

$$x_i = \frac{1}{N} \sum_{j}^{N} x_i^j \tag{4.26}$$

However, data augmentation can be performed such that different signals contribute with different weightings, determined stochastically from a uniform distribution, and equation 4.26 would then become:

$$x_i = \sum_{j}^{N} c_j x_i^j \tag{4.27}$$

where c_j is a random, positive coefficient in the interval [0,1], and $\sum c_j = 1$.

4.3.9 Conclusion

In this chapter, we highlight the important steps in the methodology involved in the classification of Raman spectrums. We look in detail at the steps involved in data collection, the type of probe used and histopathological assessment, as well as the steps involved in calibration of spectrums, and estimation/removal of fluorescence signals. We also look at noise filtering and normalization of signals and different cross-validation schemes and metrics to evaluate classification results.

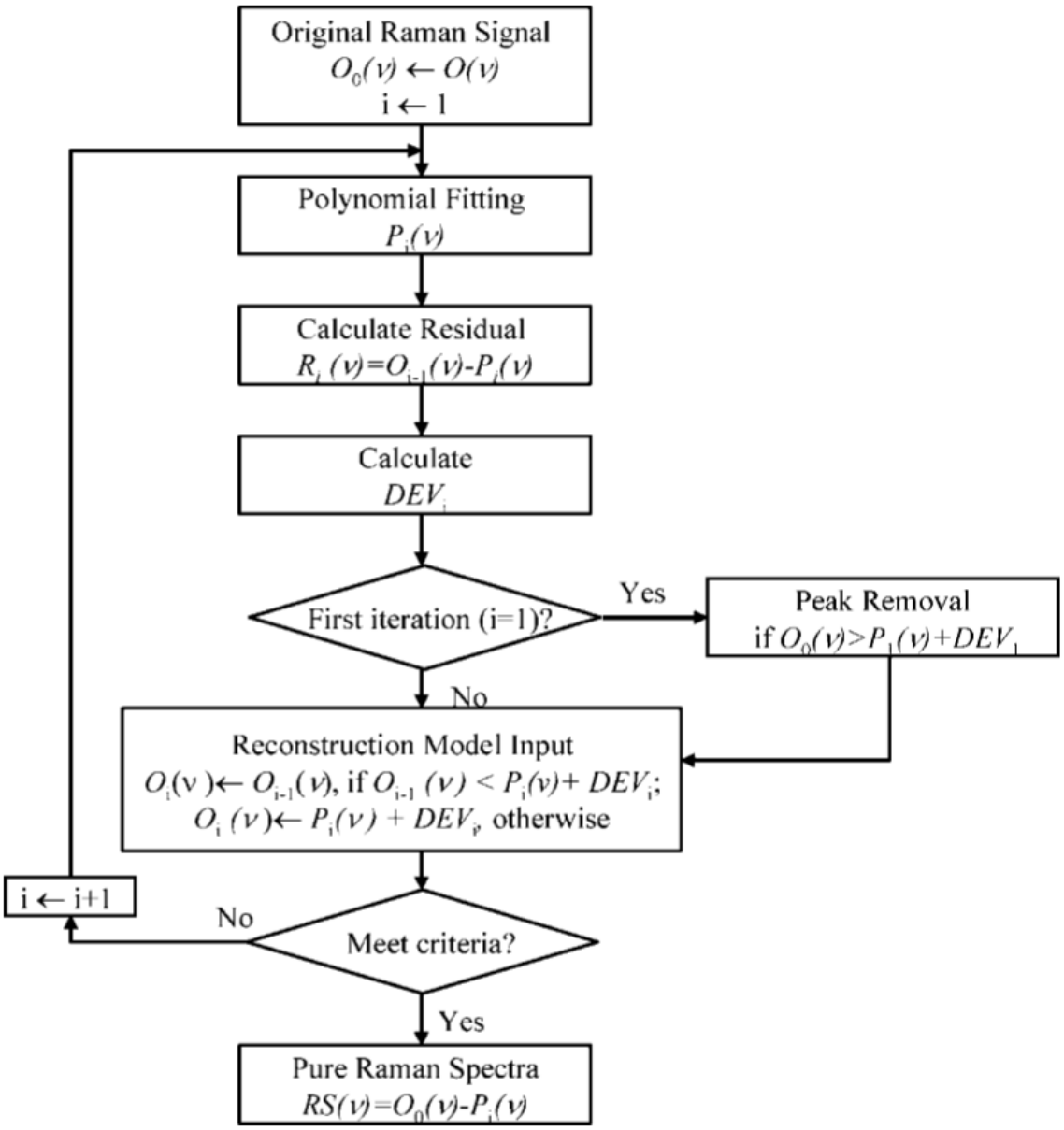


Figure 4.7: IModPoly iterative algorithm detailed in a step by step manner. Image source: [42]

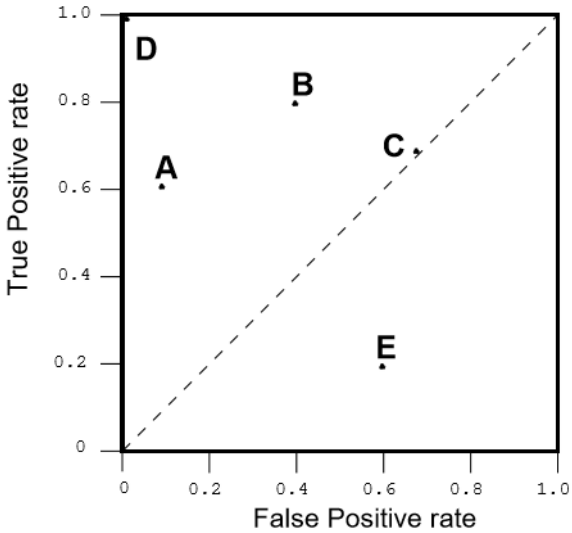


Figure 4.8: Example of a ROC plot generated for discrete points from different classifiers, A-E [19]

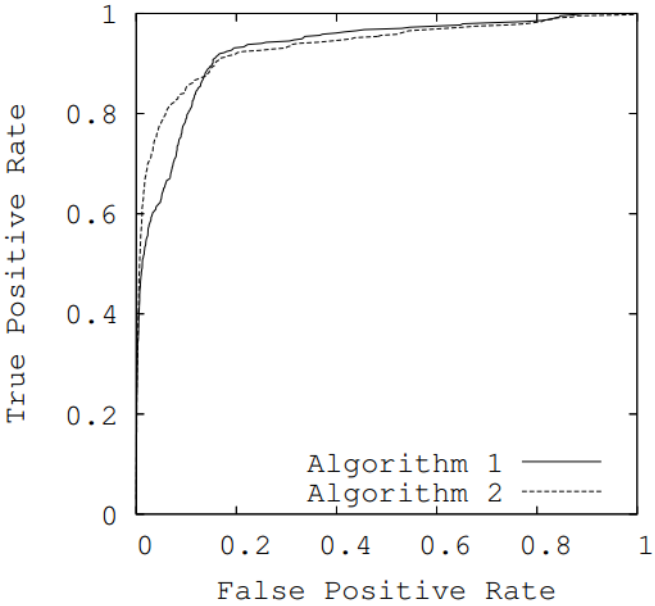


Figure 4.9: Example of a ROC curve generated for different algorithms at different thresholds T, Image source: [15]

Chapter 5

Experiments and Results

5.1 Prostate Dataset

The prostate dataset comprises of signals measured from 32 distinct human prostates ex-vivo following radical prostatectomy. The prostate removed from each patient was immediately sent to histopatholgy. A pathologist cut a slice approximately 1 cm thick which was sent to the lab for RS measurements to be taken within 60 minutes to ensure that the tissue was still fresh and simulating in-vivo measurements. Figure 5.1 shows an example of a prostate being measured on with the probe after RP.

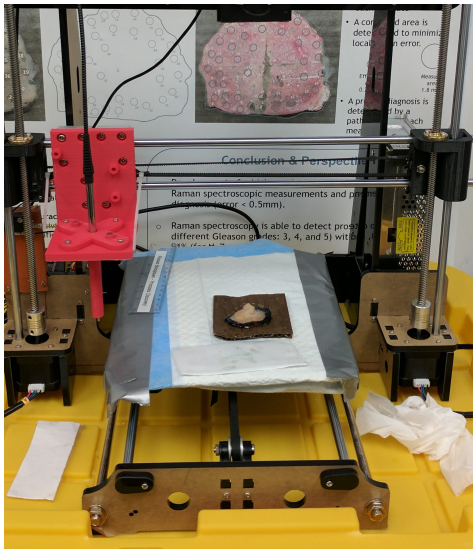


Figure 5.1: Example of a prostate ex-vivo after RP ready for measurements using RS

The Raman probe was used to acquire Raman spectra from a multitude of points by placing it in contact with the tissue and performing an acquisition. Each interrogated spot was labeled with ink and numbered. Each slice was then returned for histopathological assessment of all interrogated spots with a pathologist returning a diagnosis corresponding to the three defined classes in our dataset: benign, malignant, and extraprostatic. Some samples have no label due to inaccuracies or uncertainties from histopathology reports. Table 5.1 shows the distribution of classes in our dataset. The data is not uniformly distributed across labels. There are roughly five times more benign samples than there are malignant samples and relatively few extraprostatic samples.

Table 5.1: Data distribution in the prostate dataset

Label	# of samples
Benign	771
Malignant	149
Extraprostatic	25
No Label	337

The data is also not uniformly distributed across patients. Some patients have many more samples than others and some patients only have a single type of category of sample. Figure 5.2 shows the distribution of labeled data across patients, for the malignant and benign categories for patients 10 through 41. Patients are referred to using numvers to maintain anonymity. Since our objective is to distinguish cancerous from healthy tissue, only malignant and benign labels are considered in our classification, turning the problem into a binary classification problem.

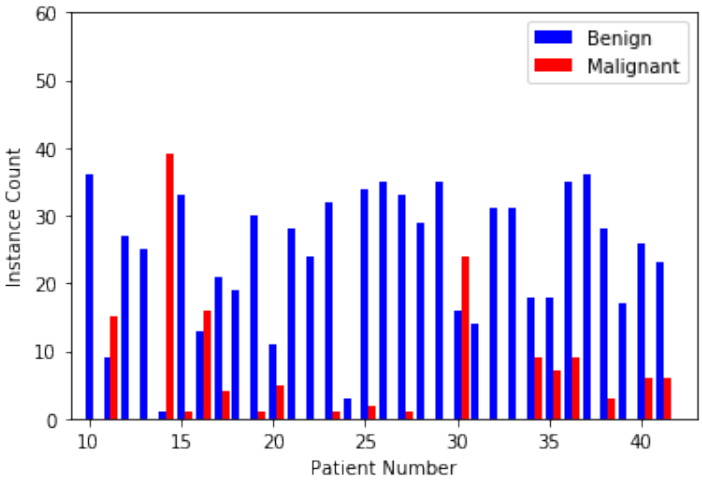


Figure 5.2: Distribution of the benign and malignant samples per patient in the prostate dataset

5.1.1 Calibration

The CCD is first calibrated along its x-axis using a sample of acetaminophen (Tylenol). To do so, the probe records the signal of acetaminophen powder sample and it is overlayed to a known benchmark reference spectrum of acetaminophen. Peaks are identified manually and lined up such that the Raman shift in cm^{-1} is known for the CCD. Figure 5.3 shows this step of the calibration process.

For each measurement of the prostate, 10 spectrums were recorded and a background measurement is also recorded. Each signal is a discrete spectrum obtained using a CCD camera array of a resolution of 1x1024. An example of a single raw acquisition signal is shown in Figure 5.4. We can see from Figure 5.4 that the laser does not operate fully in the spectral range of the CCD. Positions 0-380 do not record any meaningful data for our analysis. This is a physical limitation of the CCD and we disregard signals before position

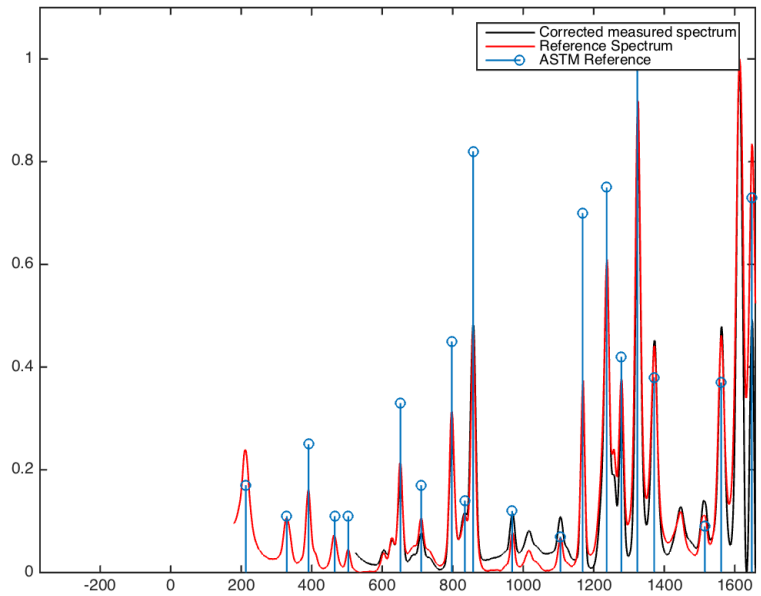


Figure 5.3: Calibration step of the x-axis on the CCD using acetaminophen

385. This leaves us with an input signal x_t of size 1xM where M $= 1024 - 384 = 640$. Figure 5.5 shows what the raw signal looks like in the region of interest, and with the proper x-axis labelled on the signal.

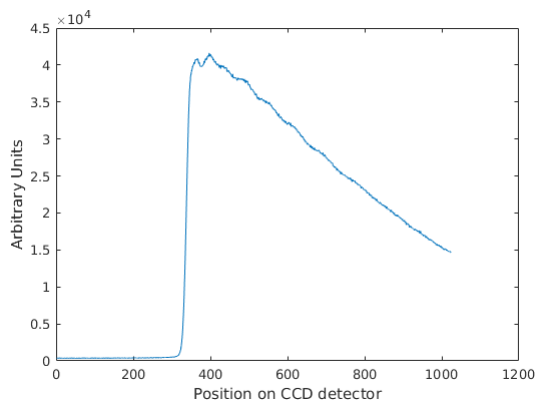


Figure 5.4: Example of a raw signal collected in the prostate dataset. 10 such signals are collected per location on a prostate

For each patient, a calibration measure, z_{stand}, of a known benchmarked material is taken before measurements are taken. This allows to calibrate the signal from the spectrometer to its known original values. Figure 5.6 shows all 32 correction curves retrieved before measurements of each of the 32 patients.

Figure 5.6 demonstrates that the correction curves, while similar in their shape, can vary in intensity and in overall shape. Variation across correction curves demonstrates the importance of proper calibration prior to measurement. Each recorded signal is then corrected as outlined in equation 4.1. Before correction, the 10 measurements are averaged together to reduce the noise from a single measurement. The background measurement,

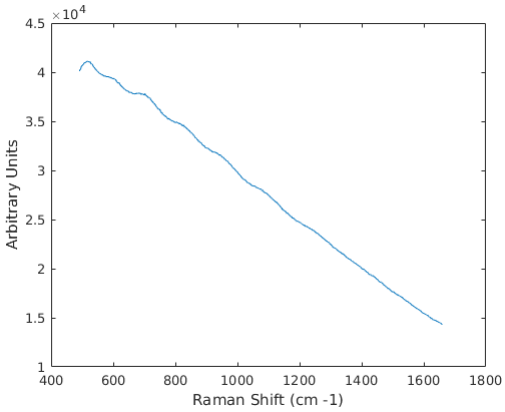


Figure 5.5: Example of a raw signal collected in the prostate dataset considering only the operating range of the CCD and the calibrated x-axis

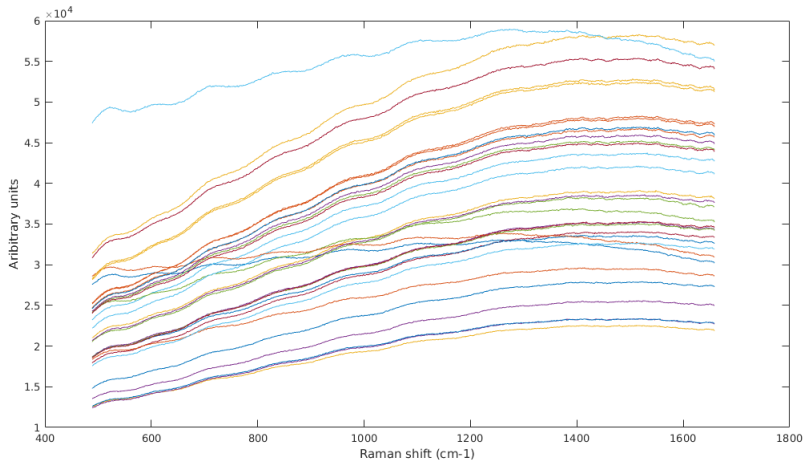


Figure 5.6: Example of a calibration measure, z_{stand}, used to calibrate the system response to known benchmarks

collected before each 10 measurements, is subtracted from the averaged signals. The resulting signal is then corrected using the procedure outlined in [11]. An example of a background corrected retrieved signal is shown in Figure 5.7.

5.1.2 AF removal

Once the raw signal is obtained, it is necessary to subtract the autofluorescence from the signal. We compare two methods, the Zhang method and the IModPoly method, as outlined in section 4.2.3. Specifically, we use $\lambda = [1, 20, 50]$ in equation 4.5 for the Zhang method, and we use a varying polynomial degree order, $d = [3, 4, 5, 6]$, for the polynomial fit for the IModPoly method. Figure 5.8 shows the approximation of the autofluorescence using these different methods in different spectral regions.

It is difficult to estimate empirically which fit is working best by visual inspection. We see that, as expected, by setting λ to smaller values, the autofluorescence estimation tries to reconstruct the original signal, and that by setting λ to higher values, the estimation learns the overall shape of the curve. However, there seems to be regions of overlap between

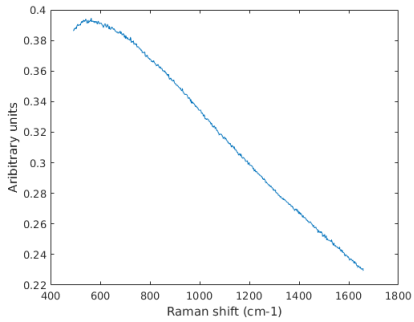


Figure 5.7: Example of a corrected signal using the correction procedure outlined in [11]

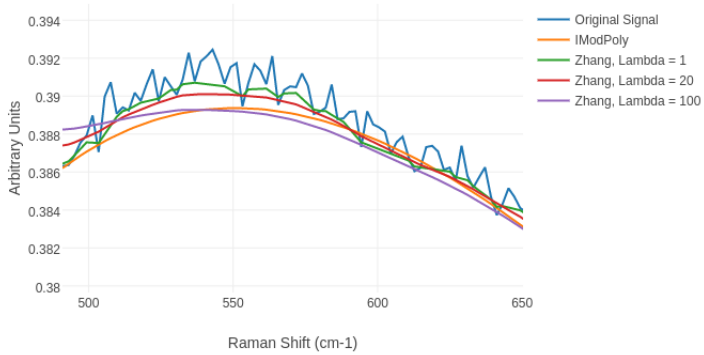


Figure 5.8: Autofluorescence estimation using the Zhang and IModPoly methods

the Zhang and IModPoly estimations, suggesting that one method might not be necessarily better than the other. One significant drawback from the Zhang method is the runtime, compared to the runtime for the IModPoly method. The runtime of IModPoly is, on average, of 0.0012 seconds per spectrum while the runtime for Zhang is of 0.23 seconds per spectrum. The Zhang method takes several orders of magnitude longer than the IModPoly, which is a very important consideration for real-time measurements. It is therefore important to assess how the Zhang performance overall compares to the IModPoly performance overall and if its costly run-time is justifiable.

The estimated fluorescence signal is then subtracted from the original signal. We are left with a relatively noisy signal, that we smooth using a Savistky-Golay Filter of Window-length of 21 and order of 3 as outlined in the Methods section. We then apply a standard-normal-variate (SNV) scheme to the signals to ensure that all signals are more or less within the same range and so that the classifiers do not try classifying on absolute intensities, since those were not controlled during signal gathering.

We compare the estimated Raman portion of the signal when using the Zhang AF removal method compared to the IModPoly AF method in Figure 5.9. We can see that most of the peaks can be seen whether using one method or the other, but that the polynomial fit seems to push them up or down at various locations while the Zhang method seems to keep them all around the same height. This is due to inherent differences in the way these algorithms were desgined. This suggests that local variations might be more important to focus on during classification as opposed to direct peak analysis. This also suggests the

importance of sticking to a well-defined protocol for Autofluorescence removal in eventual clinical use. Edges of the AutoFluorescence in Zhang and in IModPoly are also prone to divergence from the original signal, as seen in Figure 5.8 and this suggests ignoring portions of the signal close to the edges.

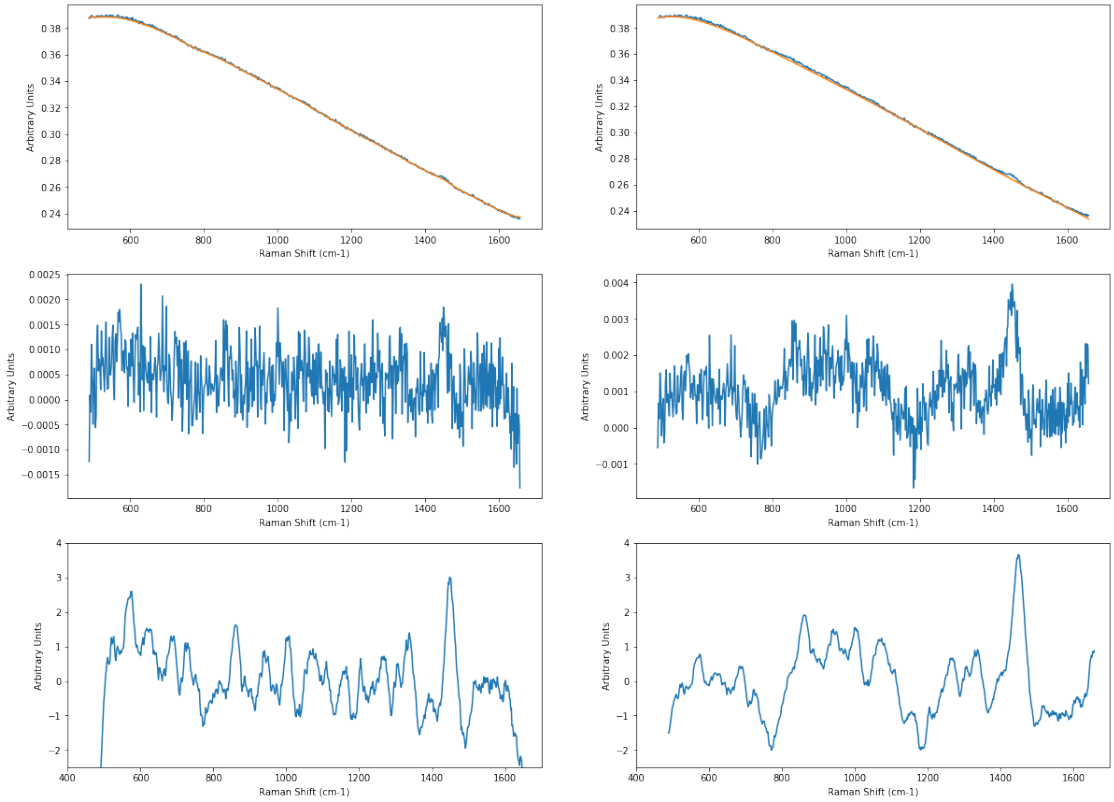
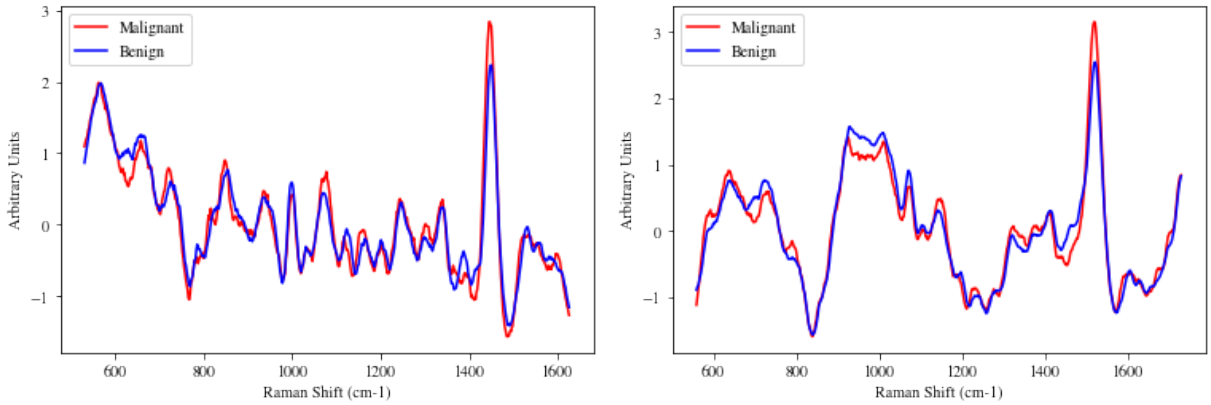


Figure 5.9: From top to bottom: Autufluorescence is estimated (top), removed from the original signal (middle), and the signal is smoothed using Savitsky-Golay filtering and Normalized using an SNV scheme (bottom). On the left, Zhang AF estimation using $\lambda = 50$ and on the right, PolyFit using a degree of 6.

Figure 5.10 compares the average of the signals separated by category (benign vs. malignant) using the Zhang AF removal compared to the IModPoly removal. While differences exist in both, it is difficult to determine which will perform best at the task of classification. We need a quantifiable means of differentiation between the different methods. To do so, we use the AUC of ROC curves for 10-fold cross-validation in the next section.

5.1.3 Evaluation of AutoFluorescence

We will first determine which AutoFluorescence (AF) removal method to consider for further analysis by determining which one works best using a single classifier. K-fold validation will be used as a scheme to evaluate the performance of the AF removal. K-fold validation is advantageous in this case because we only need to run the iterations K times, as opposed to N times where N would be the total number of samples we have if we resorted to using the Leave-one-spectrum-out approach. The data is split into K=10 random, unique and non-overlapping subsets, spanning the entire dataset, such that each individual signal is left out exactly once for evaluation once. Each subset is left out once for



(a) Zhang, $\lambda = 50$

(b) IModPoly

Figure 5.10: The means of all signals, separated by categories, after AF removal. On the left, AF was removed using Zhang's method with $\lambda = 50$, on the right IModPoly using a degree 6 polynomial.

testing and the remaining data is separated into a training and validation set. 10% of each training set is used as a validation set on every iteration. The data in the training set is duplicated such that the classes are approximately balanced during training. This is done to avoid the local minima consisting of a single prediction due to the class imbalance. This process is repeated K times and a mean receiving operating characteristic (ROC) curve is generated by plotting the True Positive Rate (TPR) as a function of the False Positive Rate (FPR) at different threshold values for every fold. The area under curve (AUC) of the ROC-curve is used as a metric to determine performance metrics such that a higher AUC indicates better classification performance. The AUC is computed across each fold and an average AUC is used to compare one method to another.

The classifier used in this case is a fully-connected MLP consisting of 2 layers, similar to the one used in [30]. The first and second layers consists of 40 activations each. A ReLu activation is used after each layer. The final layer consists of a Softmax activation returning the probability of classification. The network is trained using stochastic gradient descent on mini-batches. The mini-batches consist of signals chosen at random from the training set. After all signals from the training set have been seen by the network once (one epoch), the network is deployed on the validation set. The network is trained for a number of epochs and the weights from the epoch with highest performance on the validation set is used for evaluation on the test set. This avoids selecting for a network which might overfit on the training set.

We iterated through $\lambda = [1, 10, 50, 100, 200, 500]$ for Zhang's method and $deg = [4, 5, 6]$ for the IModPoly method. We then compare their respective AUC scores. Table 5.2 shows the results of the AUC found for all the different values of λ and deg and shows that $\lambda = 100$ is a local maximum of the AUC for the Zhang method, and $deg = 6$ is a local maximum of the IModPoly method. Since IModPoly and Zhang AF removal scored similarly, at best, with respective average AUCs of 0.925 ± 0.035 and 0.920 ± 0.019, and since, as discussed earlier, the IModPoly offers much quicker runtime than the Zhang implementation, we will focus all future results and discussions around using the IModPoly method with degree 6 for AF removal.

Table 5.2: Mean AUC score for the Zhang and IModPoly AF removal methods

Method	AUC
Zhang ($\lambda =$)	
1	0.797 ± 0.042
10	0.898 ± 0.029
50	0.914 ± 0.031
100	**0.920** ± 0.019
200	0.906 ± 0.046
500	0.890 ± 0.049
IModPoly (Deg =)	
4	0.913 ± 0.038
5	0.918 ± 0.026
6	**0.925** ± 0.035

5.2 K Fold cross-validation

We will use the same K-fold cross-validation method discussed in the previous section to evaluate different classifiers and parameters, particularly focusing on multi-layer perceptrons (MLP), support vector machines (SVM) and convolutionnal neural networks (CNN). We will use the same K-fold cross-validation scheme used in the previous section to split up the data in to 10 independent test sets, using the same pseudo-random scheme every time across classifiers to ensure consistency, and use the mean AUC as our evaluation metric.

5.2.1 MLP

We will now look more in depth at the layer sizes and hyper-parameters of the MLP used in the previous section. We use Keras with theano in the backend to train our neural networks [10]. We inspire ourselves on the architecture of the MLP from Jermyn et al. in which they use a 2-layer MLP to classify Raman spectrums sampled in-vivo from the brain [29]. Their layers consist of sizes $l_1 = 20$ and $l_2 = 10$ respectively. We base ourselves on this network to iterate between layer sizes $l_1 = [40, 20, 30, 10]$ and $l_2 = [40, 20, 30, 10]$ with the condition that $l_1 \geq l_2$ to ensure that our network is only down-sampling and not up-sampling. We summarize the results of the AUC for different layer sizes in Table 5.3.

Table 5.3: AUC scores for varying layer sizes using a 2-layer MLP

		l_2			
		10	20	30	40
l_1	10	0.925 ± 0.023	-	-	-
	20	0.932 ± 0.018	0.933 ± 0.017	-	-
	30	0.928 ± 0.021	0.932 ± 0.019	0.930± 0.021	-
	40	0.932 ± 0.024	0.932 ± 0.017	0.925 ± 0.022	0.926 ± 0.025

We can observe from Table 5.3 that varying the network layer sizes does not have a tremendous effect on the AUC suggesting that the MLP is able to adjust to our data under different conditions.

Dropout One common method to avoid overfitting of an MLP and improve classification results is to use dropout. Upon each iteration of the backpropagation algorithm, a user-defined percentage of weights are masked (or dropped) from the rest of the network. This inhibits the network from learning complex patterns from the data which might be meaningless in the real-world [57]. We compared the network performance without dropout, and with values of dropout $d = [0.1, 0.15, 0.2, 0.25, 0.5]$ and computed the AUC for layer sizes $l_1, l_2 = [20, 20]$ and $l_1, l_2 = [40, 40]$. Table 5.4 shows the AUC obtained in the different cases. We see that a slight amount of dropout (d=0.1) may benefit larger networks but does not seem to have much of an effect on the smaller network, and that they both score similarly in terms of AUC, with the larger network performing slightly better acording to the AUC metric. We will consider from now on the network $l_1, l_2 = [40, 40]$ with $d = 0.1$ as a basis for comparison to other classifiers.

Table 5.4: AUC results for different values of dropout. We see that the network performs optimally for a value of $d = 0.1$

Dropout ($d =$)	$l1, l2 = (40, 40)$ (AUC=)	$l1, l2 = (20, 20)$ (AUC=)
0	0.926 ± 0.025	**0.933** ± 0.017
0.1	**0.935** ± 0.021	0.923 ± 0.019
0.15	0.932 ± 0.020	0.920 ± 0.019
0.2	0.932 ± 0.019	0.927 ± 0.015
0.25	0.930 ± 0.017	0.925 ± 0.016
0.5	0.924 ± 0.021	0.915 ± 0.020

5.2.2 Convolutional Neural Network

In this section, a Convolutional Neural Network (ConvNet) is used as a classifier. Our Convolutional Neural Network will comprise of 1 convolutional layer, in which convolution filter weights will be learned, followed by a fully-connected layer operating on the flattened output from the convolution layer. There are more parameters to define when training a ConvNet. We need to choose the length of the 1-Dimensional kernels, l_{kern} that we would like to learn. The stride, $n_{strides}$, also needs to be defined (i.e. how many values to disance each filter application by) and determines the downsampling occuring within the signal. The number of filters to learn, l_{conv}, will also be focused on. We then need to choose the layer size for the fully-connected layer, l_{fc}. There are more parameters involved than in the MLP. We will experiment with varying parameters and use the AUC from K-fold validation as a metric for comparison. We will study, for example, the difference between using large kernels vs thin kernels and strides. The network complexity (i.e. the number of weights that will need to be learned) will be directly influenced by the aforementioned parameters. We will evaluate different architectures by presenting various combinations of $[l_{kern}, n_{strides}, l_{conv}, l_{fc}]$. We will focus more on the length of the kernels, l_{kern} as well as the number of strides, $n_{strides}$, as they are the backbone of the convolution operation and are likely to hold more importance in the classification task. The hyper-parameters will also have an impact on these performances, but will be kept fixed throughout the varying network architectures, with dropout at 0 and the learning rate, lr, at 0.01. The *ReLu*

activation function is used between every layer. We show the results of different AUC for the same K-fold scheme in Table 5.5

Table 5.5: AUC results for different values of the network parameters, $[l_{kern}, n_{strides}, l_{conv}, l_{fc}]$, for the ConvNet classifier

$[l_{kern}, n_{strides}, l_{conv}, l_{fc}]$	AUC
$[64, 32, 20, 8]$	0.922 ± 0.019
$[64, 64, 20, 8]$	0.913 ± 0.015
$[64, 32, 20, 16]$	0.931 ± 0.019
$[64, 64, 20, 16]$	0.921 ± 0.017
$[128, 64, 20, 8]$	0.933 ± 0.015
$[128, 128, 20, 8]$	0.931 ± 0.014
$[128, 64, 10, 16]$	0.922 ± 0.019
$[128, 64, 20, 16]$	$\mathbf{0.935} \pm 0.014$
$[128, 128, 20, 16]$	0.934 ± 0.012
$[128, 64, 40, 16]$	0.934 ± 0.021
$[128, 64, 20, 32]$	0.930 ± 0.018
$[256, 64, 20, 16]$	0.931 ± 0.021
$[256, 128, 20, 16]$	0.928 ± 0.016
$[256, 128, 20, 8]$	0.924 ± 0.017

It is difficult to evaluate empirically from Table 5.5 what parameters seem to influence the AUC most, however it seems that a kernel size $l_{kern} = 128$ seems to offer better performance. Other parameters, such as stride and number of filters, don't seem to have as great of an impact on the AUC overall.

Dropout Just like with the MLP, we compare the network performance with dropout values of $d = [0, 0.1, 0.15, 0.2, 0.25, 0.5]$ and computed the AUC. Table 5.6 shows the AUC obtained in the different cases. We see that dropout benefits peak at $d = 0.25$ and use this from this point onward as our dropout value for the ConvNet. The gains related to dropout were not as significant as with the MLP in this case.

Table 5.6: AUC results for different values of dropout. We see that the network performs optimally for a value of $d = 0.1$

Dropout ($d =$)	AUC
0	0.935 ± 0.014
0.1	0.935 ± 0.017
0.15	0.939 ± 0.018
0.2	0.939 ± 0.018
0.25	$\mathbf{0.941} \pm 0.018$
0.5	0.940 ± 0.014

Learning Rate Another hyper-parameter we can tune when building our ConvNet is the learning rate applied during stochastic gradient descent, which will determine the influence

of backpropagation over time in our algorithm on the weights learned by the network. A higher learning rate value will have higher impact on variation between weight values after a mini-batch, whereas a lower learning rate will allow the network to converge slowly over time. We iterate through the learning rates $l_{rate} = [0.5, 0.1, 0.01, 0.05, 0.001]$ in Table 5.7. We see that the learning rate has more important effect on the performance of the network than dropout. We found the optimal learning rate to be at $l_{rate} = 0.01$

Table 5.7: AUC results for different values of the learning rate. We see that the network performs optimally for a value of $l_{rate} = 0.01$

Learning rate (l_{rate} =)	AUC
0.5	0.910 ± 0.019
0.1	0.927 ± 0.019
0.01	**0.941** ± 0.017
0.02	0.936 ± 0.015
0.05	0.932 ± 0.017
0.001	0.885 ± 0.023

5.2.3 Support Vector Machines

We now look at using Support Vector Machines (SVM) as a classifier and compare its performance to the previous MLP and ConvNet. We use the SciKit-learn implementation of SVM which uses libsvm in its backend [49]. We use the same K-fold CV training and test sets used to train the ConvNet and MLP. There are two parameters that we seek to optimize: C which is the penalty parameter of the error term, and the type of kernel used. We consider $C = [0.001, 0.01, 0.1, 1, 5, 10]$ and the 'Linear' and 'Radial Basis Function' (RBF) kernels.

Table 5.8: AUC results for different values of C and using differnt Kernels for SVM. We see that the network performs optimally for a value of $C = 1$ and using an RBF Kernel

C	RBF Kernel (AUC=)	Linear Kernel (AUC=)
0.001	0.863 ± 0.026	0.914 ± 0.024
0.01	0.862 ± 0.026	0.933 ± 0.017
0.1	0.903 ± 0.025	0.912 ± 0.023
1	**0.943** ± 0.018	0.892 ± 0.033
5	0.936 ± 0.016	0.892± 0.033
10	0.932 ± 0.016	0.892 ± 0.033

Table 5.8 shows the results of the AUC for different parameters of the SVM implementation. We see a maximum for C = 1 using the RBF kernel and will use these parameters in future comparisons between classifiers.

5.2.4 K-fold discussion

Comparing the performance of SVMs, MLP and ConvNets using the K-fold cross-validation method, we summarize the best results for optimal conditions found in Table 5.9. To have a better understanding for how well the classifiers are performing, we use a threshold of 0.5 to evaluate the confusion matrices for each classifier and use them to evaluate their respective accuracy, sensitivity and specificity. It should be noted that varying the threshold would vary these results, which is the point of computing the AUC and not comparing direct classification results for a given threshold. Showing these results is nonetheless informative as to what classification results might actually look like for K-fold validation. The confusion matrices are shown in Table 5.10.

Table 5.9: Best performances noted for K-fold validation across classifiers

MLP (AUC=)	ConvNet (AUC=)	SVM (AUC=)
0.935 ± 0.021	0.941 ± 0.017	0.943 ± 0.018

We notice that the ConvNet and SVM score similarly in terms of AUC and they both surpass the MLP in performance. At the threshold of 0.5, the ConvNet had slightly better accuracy, specificity and sensitivity than the SVM. We notice from the confusion matrices in Table 5.10 that the accuracy, sensitivity and specificity metrics don't seem to differ much from each other across classifiers. This suggests that overall the classifiers are performing similarly and we cannot determine with certainty that one classifier performs significantly better or worse overall at the task of cancer classification in the prostate dataset using K-fold validation. This suggests that there might be a fundamental limit as to how well the data provided can be separated and that the choice of classifier will not have a tremendous effect on classification results.

		Prediction		
		Benign	Malignant	Accuracy: 0.908
MLP	Benign	724	47	Specifitiy: 0.939
	Malignant	38	111	Sensitivity: 0.744

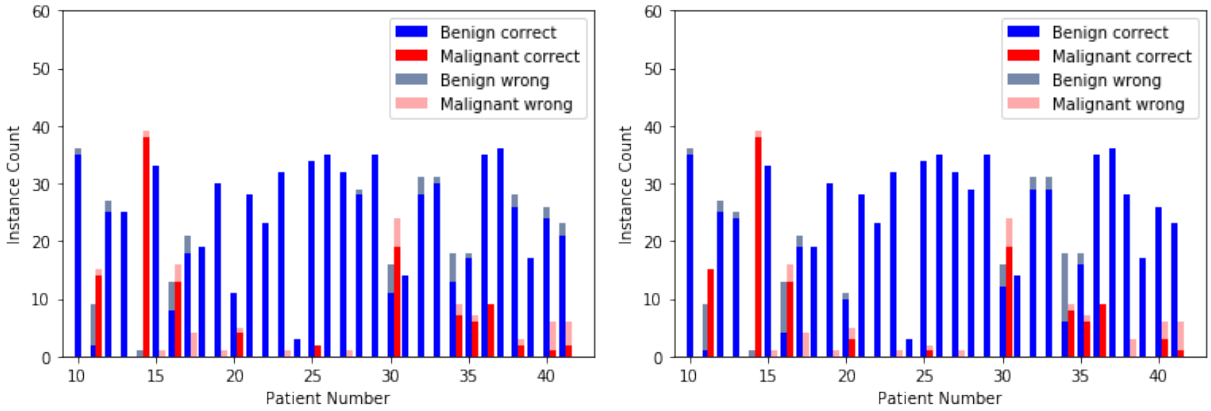
		Prediction		
		Benign	Malignant	Accuracy: 0.921
ConvNet	Benign	730	41	Specifitiy: 0.947
	Malignant	32	117	Sensitivity: 0.785

		Prediction		
		Benign	Malignant	Accuracy: 0.913
SVM	Benign	724	47	Specifitiy: 0.939
	Malignant	33	116	Sensitivity: 0.780

Table 5.10: Confusion matrices and metrics using K-fold CV evaluated at a threshold of 0.5 for the optimal architectures of MLP,ConvNet, and SVM

Figure 5.11 shows the classification results for both the SVM and ConvNet per patient per label using K-fold validation. This allows us to compare how different classifiers score across patients. We notice many similarities in the overall classification, suggesting that

perhaps the data we are analyzing is fundamentally limited due to factors such as experimental error, the possibility of disparity in labelling of spectrums compared to the actual ground truth, the limited amount of available data and the imbalance between benign and malignant specimens. It should also be noted that the K-fold validation scheme did not separate folds based on patient information, in order to best recreate results of prior works in which a Leave-One-Spectrum-Out approach was used [29][30][14][62]. K-fold is used over LOSO in the prostate dataset because of the larger scale of the dataset (N $\geq$ 700 samples) and the costly computing associated to running the experiments N times compared to K times since K$\ll$N. Presumably, LOSO should perform at least better than K-fold CV since the data associated to each patient except one sample will be available during training by the classifier. We will explore in the following sections the impact of leaving out an entire patients' data during training of the classification algorithm.



(a) ConvNet

(b) SVM

Figure 5.11: Classification results per patient per label for SVM and ConvNet using K-fold CV

5.3 Data Augmentation

In this section, we explore data augmentation and its impact on classification results. In the prostate dataset, each location probed on the prostate was measured for a total of 10 times. In the previous sections, we averaged the sampled signals and considered the average to be one signal. In order to augment the data, we use *aug* randomly weighted sums over the 10 signals as presented in equation 4.27 to artificially create *aug* times more signals for the classifiers to train on. We augment the dataset by *aug* = [5,10] using this scheme and record the AUC of the ROC curves using K-fold CV with the same pseudo-random scheme used previously, ensuring that signals associated to the same sample are assigned to the same training/validation/testing sets.

Table 5.11 summarizes the results of data augmentation using K-fold CV. We notice no impact on the SVM performance. This is expected since in a sense we are simply adding noise to our signals and barely affecting the overall distribution of the features, which in turn has little or no effect on the selection of support vectors. In the case of the ConvNet, data augmentation does not suggest any difference in AUC as well. In the case of MLP,

this method of data augmentation seems to be detrimental overall, suggesting that perhaps the MLP is more sensitive to noise and would be less robust for clinical use.

Table 5.11: AUC of the ROC curves for K-fold validation across classifiers after data augmentation

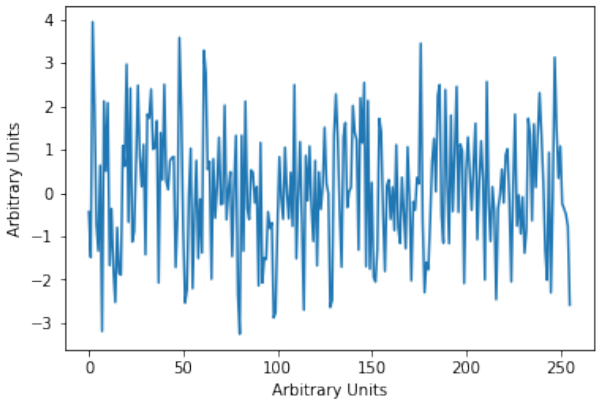
aug	MLP (AUC=)	ConvNet (AUC=)	SVM (AUC=)
1	0.935 ± 0.021	0.941 ± 0.017	0.943 ± 0.018
5	0.921 ± 0.019	0.943 ± 0.024	0.943 ± 0.018
10	0.914 ± 0.035	0.937 ± 0.016	0.940 ± 0.016

5.4 Dimensionality Reduction

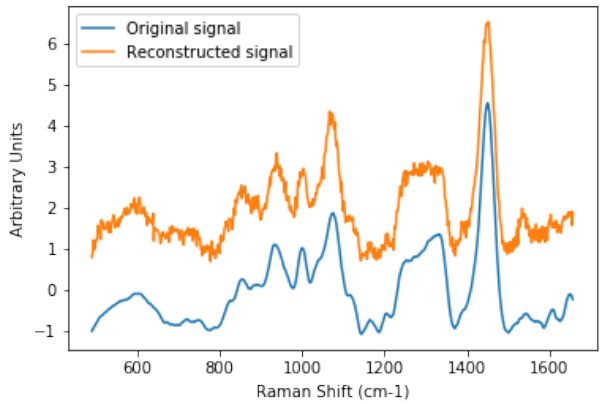
In the previous sections, we've been looking at classification results by using the Raman spectrum directly as an input. In this section, we explore the use of dimensionality reduction and its effect on classification. We will look at different methods for feature extraction to transform the data from higher-dimensionality space to lower-dimensionality space while trying to retain as much information as possible. Two unsupervised learning approaches will be compared: Principal Component Analysis (PCA) and autoencoders (AE). They are unsupervised since their implementation does not depend on the labels from the dataset. We will use the same K-fold validation scheme used in the previous sections to see how results contrast and compare to classification of the original input from prior sections. One reason to use dimensionality reduction is to attempt to remove uninformative data from the input and classify only the meaningful portion of the data, to avoid for example overfitting to noise and to leverage classification of a smaller feature space. We use the same classifiers found in the previous section which gave the highest AUC for the MLP, ConvNet and SVM. In the case of the ConvNet, we reduced the kernel size from 128 to 32 when the encoding dimension was less than 128 for implementation reasons.

5.4.1 Autoencoders

The AE seeks to find a lower-dimensional representation of the data by squeezing it through intermediate layers and reconstructing it at its output. In this case, we use one intermediate layer, and the mean squared error as the loss function. The size of the intermediate layer will determine how compressed the data will be. We vary the size of the compression layer with values $l_{size} = [64, 128, 256, 512]$ and use the same SVM, MLP and ConvNet classifiers as in the previous section to achieve classification. We exclude all of the test samples from the AE when estimating it to avoid overfitting. Figure 5.12a shows what an encoded signal looks like for $l_{size} = 256$. Each point corresponds to a feature in the new feature space. Figure 5.12 shows the difference between the reconstructed signal after being encoded and decoded by the AE compared to the original signal. We see that while the reconstruction isn't perfect (which is expected since information is lost through encoding), the overall shape is recognizable from the reconstruction and the major peaks are still distinguishable. This suggests that most of the information is properly captured by the encoder of the AE.



(a) Output of the encoding layer for the AutoEncoder with $l_{size} = 256$. This is the new feature space used by the classifiers



(b) Comparison between the original Raman Spectrum and the reconstructed Raman spectrum after going through an autoencoder with $l_{size} = 256$

Figure 5.12

Table 5.12: AUC results for different sizes of the encoding layer size l_{size} using an autoencoder for dimensionality reduction

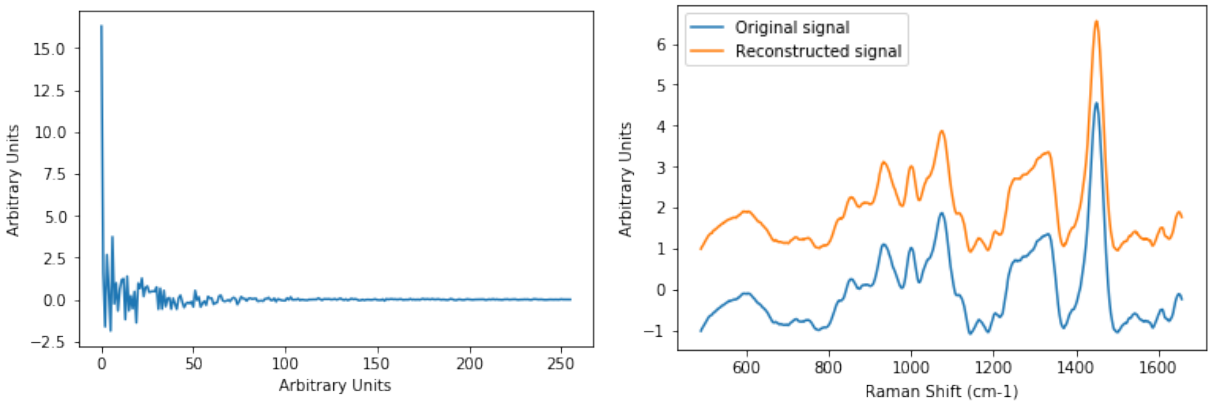
Encoding dimension (l_{size} =)	MLP (AUC=)	ConvNet (AUC=)	SVM (AUC=)
32	0.815 ± 0.043	0.866 ± 0.024	0.862 ± 0.028
64	0.848 ± 0.045	0.871 ± 0.035	0.885 ± 0.034
128	0.890 ± 0.042	0.911 ± 0.019	0.917 ± 0.029
256	0.920 ± 0.019	0.919 ± 0.021	0.927 ± 0.022
512	0.928 ± 0.018	0.934 ± 0.017	0.931 ± 0.023
None	0.934 ± 0.021	0.941 ± 0.017	0.943 ± 0.018

Table 5.12 shows the results of the AUC using the same K-fold distribution as in the prior sections, allowing us to compare metrics. We see that as we reduce the size of the encoding layer, the AUC scores decrease. However, we find that we are able to achieve similar AUC, 0.919 ± 0.021 compared to 0.941 ± 0.017, for the ConvNet, and similarly for the MLP and SVM, by reducing the feature space from 640 to 256, which corresponds to approximately 2.5 times reduction in the feature space. We observe once again that the ConvNet and SVM perform similarly and that both perform in general better than the MLP.

5.4.2 Principal Component Analysis

PCA reduces the data by selecting for a combination of features which maximizes variance across features. We used the same values of dimension reduction as with the AE by going through all values $l_{size} = [64, 128, 256, 512]$ and use the same SVM, MLP and ConvNet classifiers as in the previous section to achieve classification. We excluded all of the test samples from the PCA estimation when estimating it to avoid overfitting. Figure 5.13a shows what an encoded signal looks like for $l_{size} = 256$. Each point corresponds to a feature

in the new feature space. Figure 5.13 shows the difference between the reconstructed signal after being encoded and decoded by PCA compared to the original signal. Compared to the the reconstruction with the AE, PCA reconstruction is much less noisy and appears to reproduce the signal with higher fidelity. This is in part due to the mathematical formulation of PCA which captures the highest variation from the data.



(a) Output of the encoding layer for PCA with $l_{size} = 256$. This is the new feature space used by the classifiers

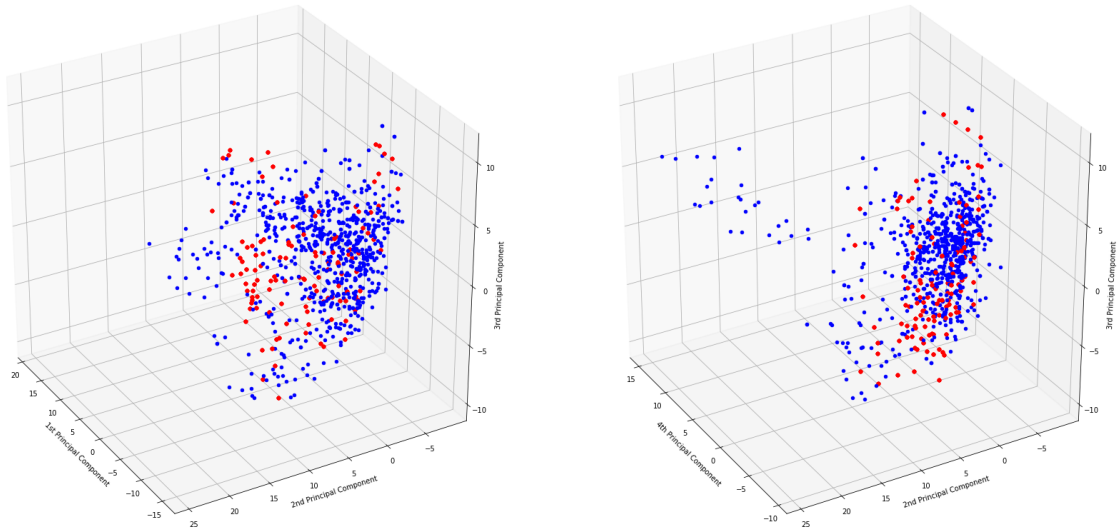
(b) Comparison between the original Raman Spectrum and the reconstructed Raman spectrum after going through PCA with $l_{size} = 256$

Figure 5.13

Another thing to notice when comparing dimensionality reduction of PCA to AE is that the distribution of intensities in the AE feature space seems uniform whereas most of the intensities of features are concentrated in the first few features in PCA. This is once again due to the mathematical formulation of PCA, which seeks features of highest variance as primary features, compared to AE which seek to learn a good representation based on stochastic rules. An interesting consequence of this is that plotting the first few PCA features against each other reveal interesting patterns. This is shown in Figure 5.14, where the first few PCA components from a sample training set from K-fold validation are plotted. We observe some clustering of the data in both graphs, but it is not clear exactly how to separate this data simply using planes. This corroborates with the results in the previous sections as to why linear kernels did not work as well as the RBF kernels in SVMs, and motivates the use of non-linear mappings for classification. Table 5.13 summarizes the different values of the AUC found when using PCA.

Table 5.13: AUC results for different sizes of the encoding layer size l_{size} using an PCA for dimensionality reduction

Encoding dimension (l_{size} =)	MLP (AUC=)	ConvNet (AUC=)	SVM (AUC=)
32	0.908 ± 0.031	0.911 ± 0.028	0.868 ± 0.036
64	0.924 ± 0.032	0.918 ± 0.029	0.897 ± 0.031
128	0.924 ± 0.026	0.929 ± 0.022	0.925 ± 0.022
256	0.917 ± 0.035	0.922 ± 0.022	0.942 ± 0.016
512	0.928 ± 0.020	0.928 ± 0.021	0.944 ± 0.018
None	0.934 ± 0.021	0.941 ± 0.017	0.943 ± 0.018



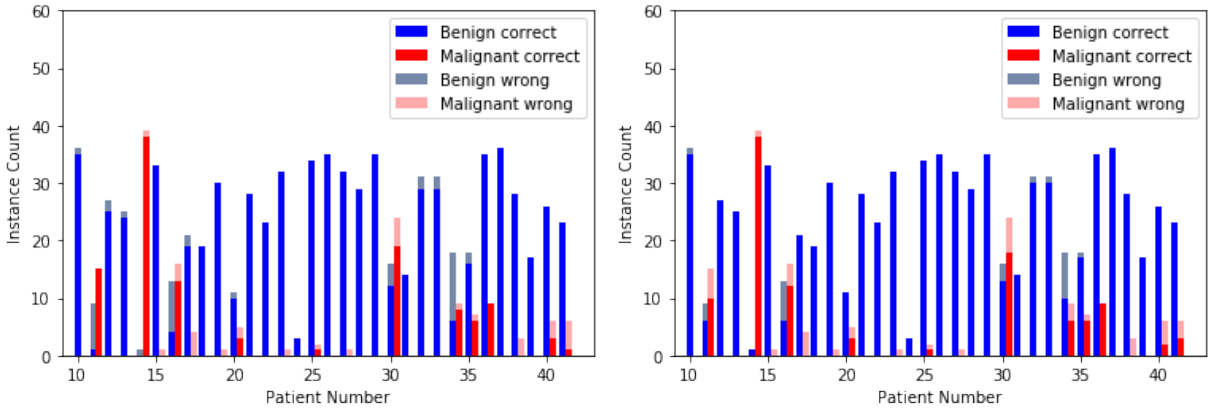
(a) 3D plot of the first second and third PCA components based on patient labels

(b) 3D plot of the second, third and fourth PCA components based on patient labels.

Figure 5.14

5.4.3 Discussion

When comparing the AUC after using PCA and AE, we notice that with AE, reducing the dimension meant a reduced AUC across classifiers. This is corroborated with the fact that reconstruction of the signals were noisy, suggesting that useful information from the signal was lost upon AE transformation, however the AUC did not suffer dramatically, even with reductions of the feature space up to 2.5x. It appears that reducing data through PCA by 2.5x barely affects the SVM classifier, whereas it has a more noticeable effect on the ConvNet and MLP. This is possibly in part due to the fact that the SVM seeks the best hyperplane separations, whereas ConvNets are trying to learn kernel representations which might be harder to distinguish in PCA space. This suggests that using an SVM as a classifier is more robust when considering dimensionality reduction. It might also be that the ConvNet structure is less adaptive than the SVM, which was optimized for a higher feature space. We notice that when compared to the AE, the results don't decrease as steadily when decreasing the encoding dimension with comparable AUC to the original input space. We notice that SVMs behave significantly worse when considering a much smaller feature space ($l_{size} = 32$) while ConvNets and MLP perform similarly with dimension reductions of up to 20 times. This suggests that using PCA for data reduction is more efficient than using AE, however, it also demonstrates that overall the classifiers perform better when considering the raw spectrums as opposed to reduced feature spaces. We compare the classification results per patient of SVMs for the regular feature space and for an encoding dimension of 256 in Figure 5.15. We notice little variation in classification error between the two, suggesting that most of the information is retained when applying PCA. We notice in some cases, for example with patient 11, that the original feature space has higher sensitivity compared to the reduced feature space, but that specificity is decreased. This might suggest combining the results of both feature spaces to be used as a classifier could help increase detection.



(a) Original Raman spectrum as input with SVM classifier

(b) PCA with $l_{size} = 256$ as input with SVM classifier

Figure 5.15: Classification results per patient per label for SVM using K-fold CV on the raw input (left) and PCA dimensionality reduction (right)

5.5 Leave-One-Patient-Out

We've looked in the previous section at a K-fold validation approach to differentiate between classifier performances. In this section, we explore a leave-one-patient-out (LOPO) approach, in which all the spectrums of an individual patient are left out for testing, while the remainder of the data is used for training and validation purposes. We use the same classifiers and architectures used in the previous sections, i.e. MLP, ConvNet, and SVM for classification. LOPO enables us to explore how the system would perform in real-life scenarios when a new patient is introduced since no prior information from that patient would be available in the training set. Drastically different results to K-fold validation might indicate clustering of the data within patients and suggest that the models might be learning irrelevant features such as noise models as opposed to qualitative features discriminating between hisotpathologies. This would also suggest that using patient priors for a pre-training of the network might be a good idea: for example, scanning certain regions of the brains that the surgeon can with confidence assert are malignant and/or benign, and fine-tuning the classifiers on those regions. Because of the imbalance between patients, and because some patients might have only a single type of label associated to them (i.e. only benign), metrics such as the AUC used in the previous section cannot be used per patient since calculating the TPR might incur divisions by zero (i.e. in the case of perfect classification of a patient with only benign classes, TP+FN = 0). Instead we present the confusion matrices overall in Table 5.14 for a threshold of 0.5 and figures of classification results per patient per label in Figure 5.16.

Figure 5.16 compares results per patient of a LOPO approach compared to the K-fold approach, using the same classifiers, and Table 5.14 presents the respective confusion matrices and metrics of the LOPO approach for a threshold of 0.5. We observe that the overall sensitivity and accuracy are consistently much worse in the case of LOPO, while the overall specificity seems to be barely affected. This seems to suggest that classification results suffer tremendously without prior information of patients and that the classifiers have a bias towards classifying samples as benign. This seems to suggest that introducing known classification results of a patient to the training set will improve classification

significantly. This also seems to suggest that the data might be clustered within patients, and the assumption that Raman spectrums share commonalities across patients might not be entirely valid. This also sheds light on the foundation of classification results in literature when the validation method used is a Leave-One-Spectrum-Out approach, if multiple spectrums from a same patient are used during training of the classifier.

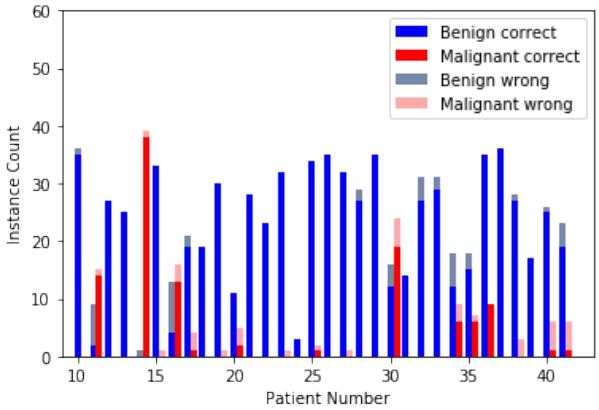
One way to verify these claims is by using the t-Distributed Stochastic Neighbor Embedding (t-SNE) method to the reduced data via PCA [44]. t-SNE allows the representation of highly-dimensional data in 2D by grouping similar objects in hyper-dimensional space together and dissimilar objects further apart. Figure 5.17 visualizes the data per patient and per label. We can clearly distinguish clusters of patients in Figure 5.17. We can also corroborate those clusters with the results from Figure 5.16: we observe that within a patient cluster, if the data appears separable using the t-SNE method, it seems to perform well in K-fold and seems to perform poorly otherwise. For example, patient 36 has what looks to be a linearly separable cluster in Figure 5.17 and performs well in both the K-fold and LOPO scheme as seen in Figure 5.16. Patient 30 and 40 also demonstrate a cluster which in this case seem visually hard to draw boundaries between labels, and the classifier struggles to correctly label the malignant samples in both LOPO and K-fold, in this case due to the prevalence of benign samples otherwise present in the area. This suggests that taking samples of known pathology during surgery could lead to an increase in classification performance.

		Prediction		
		Benign	Malignant	Accuracy: 0.842
MLP	Benign	699	72	Specifitiy: 0.907
	Malignant	73	76	Sensitivity: 0.510

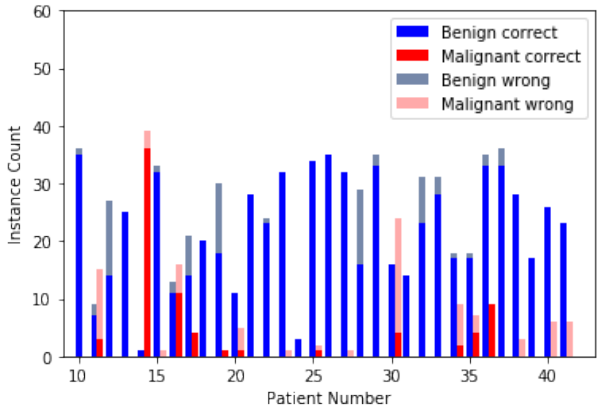
		Prediction		
		Benign	Malignant	Accuracy: 0.842
ConvNet	Benign	702	69	Specifitiy: 0.910
	Malignant	76	73	Sensitivity: 0.489

		Prediction		
		Benign	Malignant	Accuracy: 0.876
SVM	Benign	722	49	Specifitiy: 0.936
	Malignant	65	84	Sensitivity: 0.564

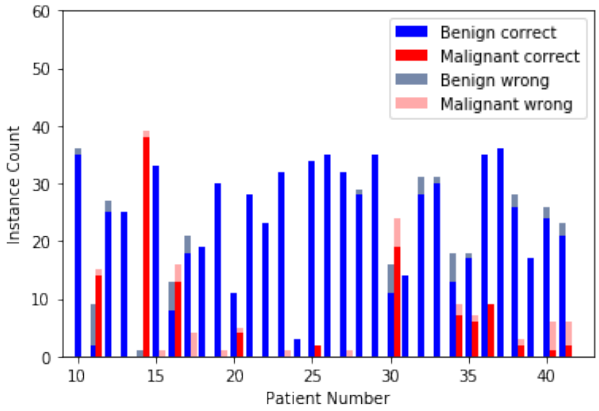
Table 5.14: Confusion matrix for K-fold using a 2-Layer MLP for Fold 7 (best AUC), Fold 0 (worst AUC) and overall performance across folds



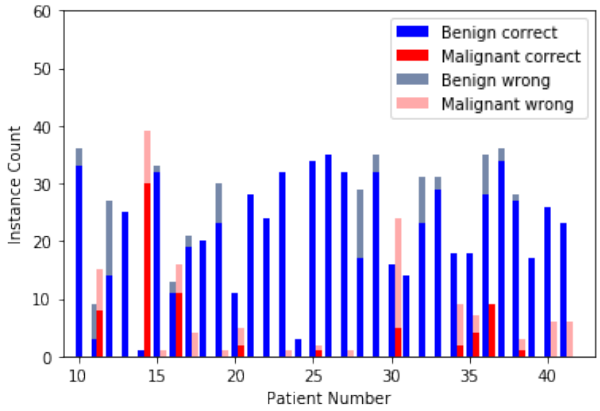
(a) K-fold CV with MLP classifier



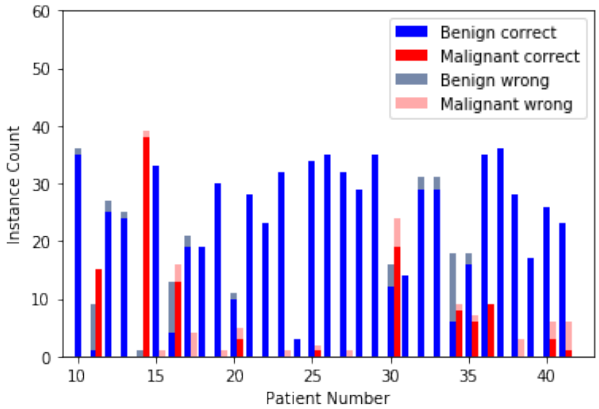
(b) LOPO with MLP classifier



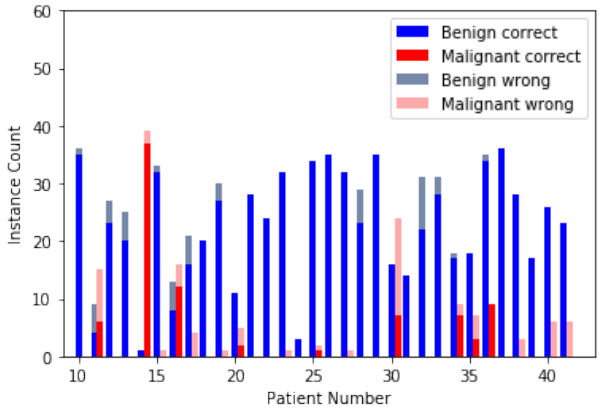
(c) K-fold CV with ConvNet classifier



(d) LOPO with ConvNet classifier

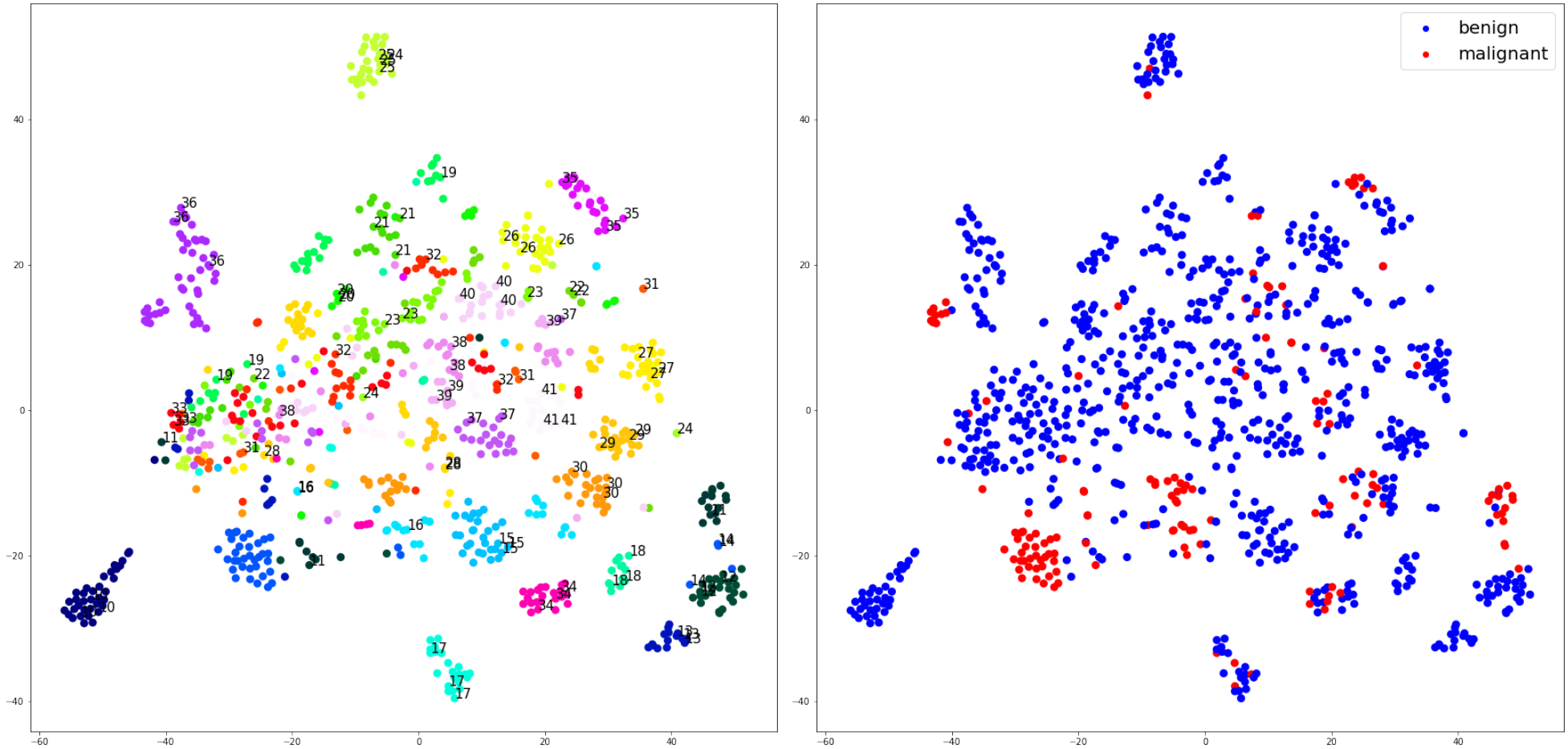


(e) K-fold CV with SVM classifier



(f) LOPO with SVM classifier

Figure 5.16: Classification results per patient per label for all classifiers. Figures on the left use a K-fold CV scheme and on the right use a LOPO CV scheme



(a) t-SNE labelled according to patient number

(b) t-SNE labelled according to pathology diagnosis

Figure 5.17: t-SNE dimensionality reduction visualization for the prostate dataset

5.6 Brain Dataset

The brain dataset comprises of signals measured from 12 distinct human brains in-vivo by a neurosurgeon during surgery. Raman spectrums were collected from various regions of interest during surgery and the same sampled locations were sent to histopathology for labelling. The equipment used for the brain dataset and setup is very similar to the setup used for the prostate dataset, and the same pre-processing steps are used for the brain dataset as were used in the prostate dataset. The dataset comprises of a total of 152 samples. There are 3 types of labels returned from histopatology in the brain dataset: benign, infiltrated and malignant. In the context of brain tumor removal, differentiating between 'Benign' and 'Infiltrated' is the priority, since 'Malignant' samples often lie in the visible tumorous region while infiltrated regions are invisible to the surgeon and could lead to residual cancer after surgery, significantly impacting survival rate. Removing critical benign brain tissue could cause serious functional damage within patients. Figuring out where the limits are between infiltrated cells and healthy cells is thus more important and will be the focus of this section, turning once again the problem into a binary classification problem, focusing on the 'Benign' and 'Infiltrated' classess. Table 5.15 shows the distribution of samples per label, and Figure 5.18 shows the distribution of labels per patient. As in the prostate dataset, the data is not equally distributed among patients, however the data is more balanced than in the prostate dataset. There is also apprixmately 10x less data in the brain dataset compared to the prostate dataset.

Table 5.15: Data distribution

Label	# samples
Benign	71
Infiltrated	57
Malignant	24

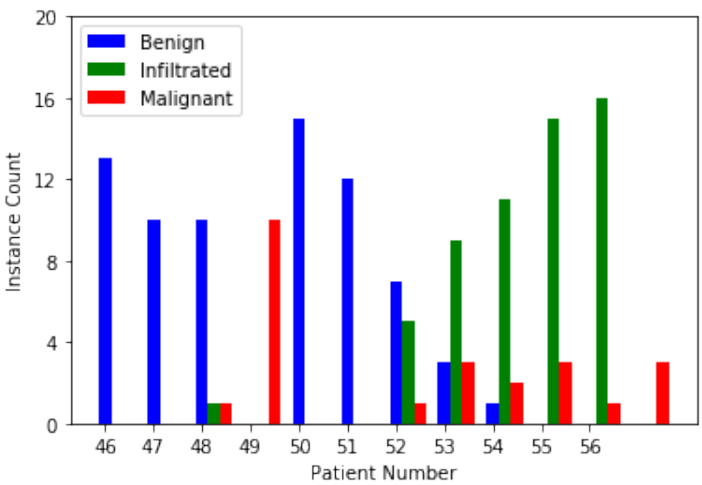


Figure 5.18: Distribution of benign and malignant samples per patient in the brain dataset

Similar methods we're used for calibration of the x-axis and the same reference material was used for curve correction before measurements. The same parameters used in the prostate dataset for AF removal and SG filtering are used on the brain dataset for

consistency. The same spectral range was selected from the brain dataset to match the spectral range of the prostate dataset and signal lengths were kept consistent between sets so that classifiers across datasets would still be relevant. Figure 5.19 shows the means of all the signals across labels.

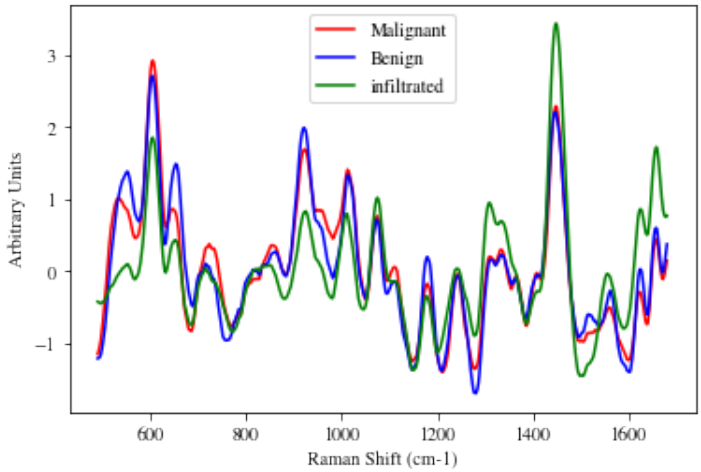


Figure 5.19: The means of all signals, separated by categories, after AF removal, on the brain dataset

We will proceed using the same analysis as in the previous section, first deploying our classifiers using a K-fold cross-validation method, using K=10, and then use a leave-one-patient-out approach to see how the method generalizes when new patients are introduced within the set. The same types of classifiers will be used in this section as were used in the prostate section, i.e. an MLP, a ConvNet and an SVM. Since we are operating in the same spectral range, using the same AF removal and noise filtering methods, and using as inputs signals of the same dimensions, we can assume that the structures of the classfiers found to be most efficient in the prostate dataset will be just as efficient in the brain dataset. We once again use the AUC of the ROC curves as a metric to differentiate between classifier performance.

5.6.1 K-fold cross-validation

Just like with the prostate dataset, we use a K-fold cross-validation approach by splitting the data into K=10 unique, randomly distributed non-overlapping sets. Every set is left out once for evaluation as a test set, using the remaining data as a training and validation set. The classifier is trained on the training set, and evaluated on the validation set at every epoch. Table 5.16 summarizes the results using an MLP, ConvNet and SVM.

Table 5.16: Best performances noted for K-fold validation across classifiers

MLP [20,20] (AUC=)	MLP [40,40] (AUC=)	ConvNet (AUC=)	SVM (AUC=)
0.939 ± 0.033	0.846 ± 0.130	0.941 ± 0.036	0.955 ± 0.029

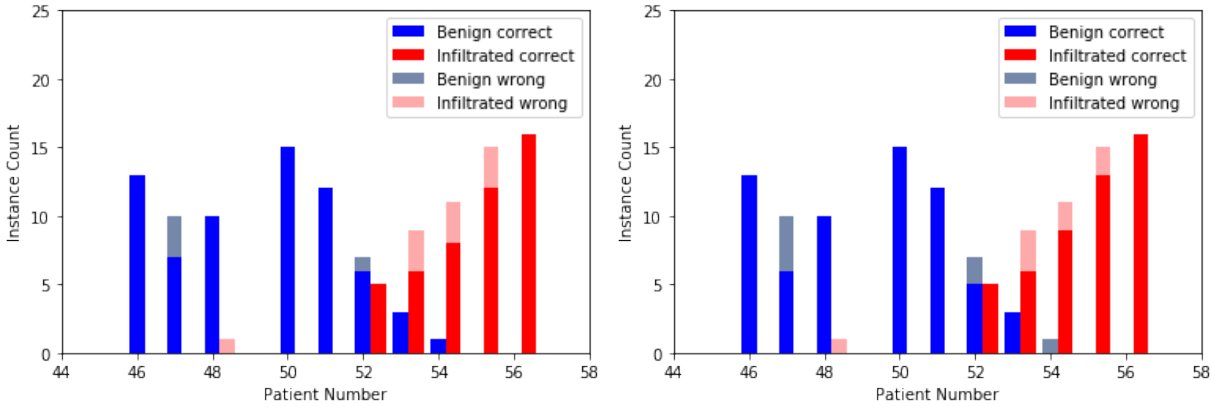
We notice once more that the ConvNet and SVMs perform similarly in terms of AUC, while in this case the MLP with layer size $l1$=$l2$=40 performs significantly worse. This

could be explained by the complexity of the MLP model versus that of the ConvNet model, and the reduced amount of data available in the brain dataset compared to the prostate dataset. The architecture of the MLP consists of approximately 5x more weights than the ConvNet, explaining the disparity in both the prostate and brain datasets between the MLP and the other classifiers. When using an MLP with $l1$=$l2$=20, we observe similar performance between the ConvNet and MLP. In the brain dataset, the SVM scores higher than either of the neual networks in terms of AUC. We notice that when considering a threshold of 0.5, the confusion matrices shown in Table 5.17 do not vary much in terms of classification results between the ConvNet and SVM. While the ConvNet shows higher specificity, the SVM offers higher sensitivity. We notice that the MLP with $l1$=$l2$=40 scores significantly worse in terms of accuracy, sensitivity and specificity, and that while the MLP with $l1$=$l2$=20 has better accuracy and specificity overall than its MLP counterpart, it scored much lower in sensitivity. This suggests once more that the SVM and ConvNet are better suited at the task of classification than the MLP. Figure 5.20 illustrate these same results per patient per label for both ConvNet and SVMs, and we see similar results per patient per label for both methods.

		Prediction		
		Benign	Infiltrated	Accuracy: 0.836
MLP [20,20]	Benign	69	2	Specifitiy: 0.972
	Infiltrated	19	38	Sensitivity: 0.667
		Prediction		
		Benign	Infiltrated	Accuracy: 0.797
MLP [40,40]	Benign	63	8	Specifitiy: 0.887
	Infiltrated	18	39	Sensitivity: 0.684
		Prediction		
		Benign	Infiltrated	Accuracy: 0.891
ConvNet	Benign	67	4	Specifitiy: 0.944
	Infiltrated	10	47	Sensitivity: 0.825
		Prediction		
		Benign	Infiltrated	Accuracy: 0.883
SVM	Benign	64	7	Specifitiy: 0.901
	Infiltrated	8	49	Sensitivity: 0.860

Table 5.17: Confusion matrices for K-fold CV and associated metrics for MLP, ConvNet and SVM

We notice from Table 5.17 that using the K-fold approach, the SVM obtains better sensitivity scores while the ConvNet obtains better specificity scores. This suggests that using multiple classifiers in an ensemble method might be beneficial to classification performance and is worthwhile exploring in future work. Figure 5.20 shows the classification results per patient per label for both the SVM and ConvNet. The fact that both classifiers scored similarly across patients even though they operate very differently from one another from an algorithmic point of view also seems suggests that they are both capturing similar features for classification and that perhaps fundamentally either some of the labels are inconsistent or some of the measurements erroneous due to experimental error or simply due to subtle changes in procedure that were overlooked. Having access to larger amounts of data would definitely help answer these questions, however data collection is limited due to the very nature of the procedure.



(a) ConvNet (b) SVM

Figure 5.20: Classification results per patient per label for SVM and ConvNet using the K-fold CV scheme

5.6.2 Leave-One-Patient-Out

Just like with the prostate dataset, we use a LOPO scheme for the brain dataset and compare it to results using K-fold CV. Using a LOPO apporach mimics how the system would perform in a clinical setting on a new patient without any priors from that patient, since K-fold validation could potentially overfit on priors of a patient. Significantly different results to K-fold CV would indicate that the learned features are perhaps not able to generalize across patients.

Table 5.18 summarizes the confusion matrices and metrics obtained using the LOPO CV scheme and a threshold of 0.5, while Figure 5.21 compares the performance per patient per label of the different classifiers using K-fold and LOPO. While we observe a decrease in the overall performance across metrics, the drop in sensitivity is not nearly as important as it was in the prostate dataset. Looking closer at Figure 5.21, we notice that the drop in specificity, the ability to identify the 'Benign' category correctly, is mainly attributed to patient 47 in the case of the SVM and the ConvNet, and that the classifiers also had a harder time classifying this patient in the K-fold CV approach. This might suggest that something in the procedure of the data collection for that patient might have been slightly different and could potentially be attributed to experimental or human error.

We once more use the t-SNE approach on the brain dataset, which we have reduced via PCA, to see if we observe the same level of clustering per patient as we did in the prostate dataset. Figure 5.22 shows the projection both per patient and per label. We notice some clustering per patient once more in the brain dataset, however in this case, there also seems to be clustering with respect to the labels, and a clearer separation between labels using only the first 2 components of PCA reduction. We notice for some patients, for example patients 52 and 53, clustering occuring in both regions of the diagram, such that the samples labelled as benign cluster together in one region and the samples labelled as infiltrated cluster in another region. This seems to agree with the classification results from Figure 5.21, where patients 52 and 53 have samples of both labels and classification results are good. In the case of patient 47, which had bad classification results overall both using ConvNets and SVM classifiers, we notice from Figure 5.22.(a) that the patients' data seems scattered throughout and inconsistently with respect to its label. This might, once

		Prediction		
		Benign	Infiltrated	Accuracy: 0.75
MLP [20,20]	Benign	66	5	Specifitiy: 0.923
	Infiltrated	27	30	Sensitivity: 0.526

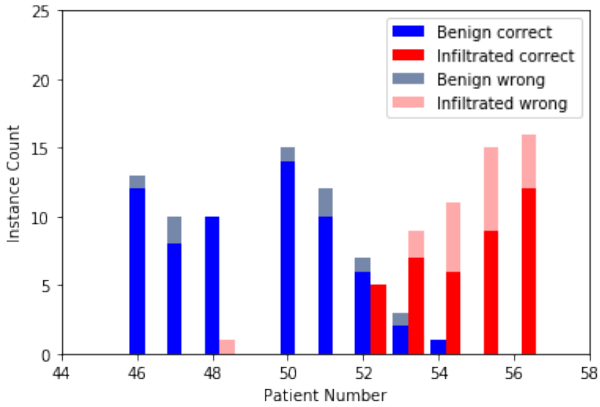
		Prediction		
		Benign	Infiltrated	Accuracy: 0.742
MLP [40,40]	Benign	54	17	Specifitiy: 0.761
	Infiltrated	16	41	Sensitivity: 0.719

		Prediction		
		Benign	Infiltrated	Accuracy: 0.844
ConvNet	Benign	62	9	Specifitiy: 0.873
	Infiltrated	11	46	Sensitivity: 0.807

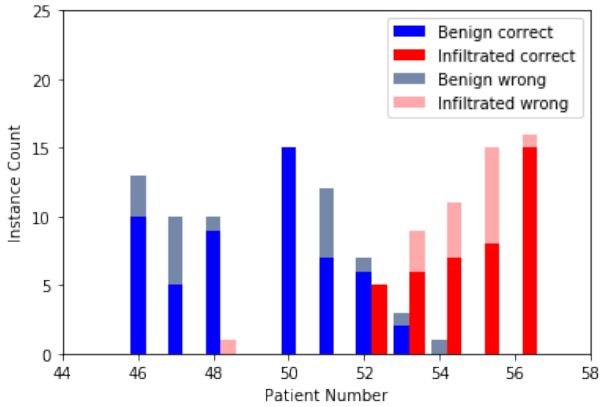
		Prediction		
		Benign	Infiltrated	Accuracy: 0.836
SVM	Benign	61	10	Specifitiy: 0.859
	Infiltrated	11	46	Sensitivity: 0.807

Table 5.18: Confusion matrices and metrics using a LOPO CV scheme and different classifiers in the brain dataset

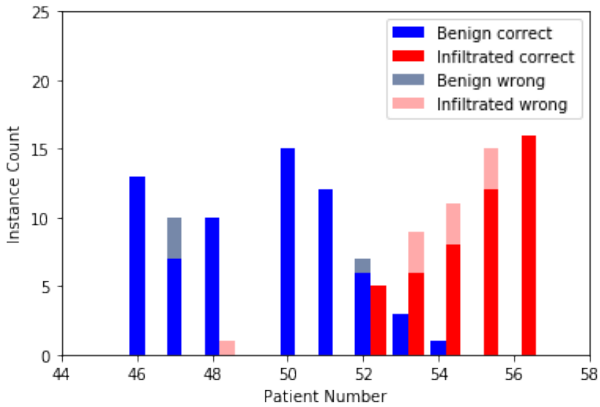
again, indicate that this patients' data might be an outlier and could be due to a myriad of factors.



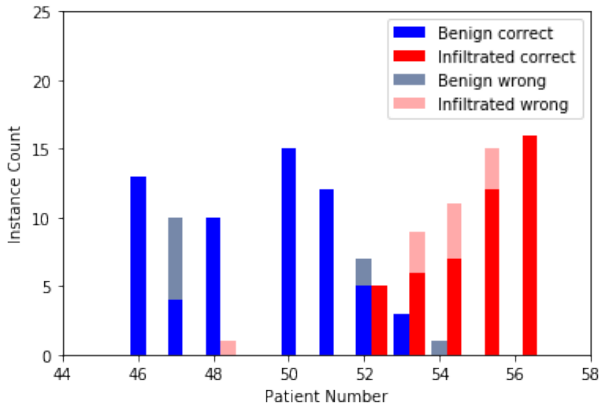
(a) K-fold CV with MLP classifier



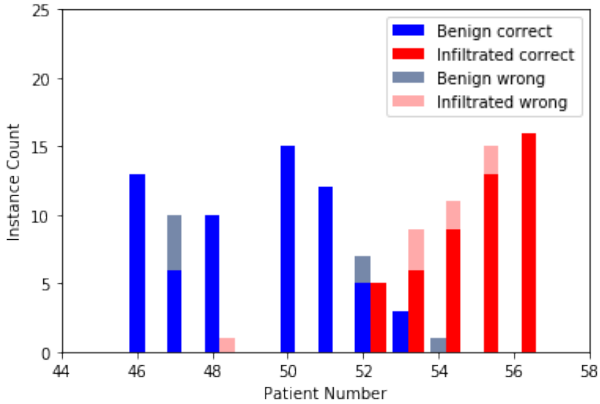
(b) LOPO with MLP classifier



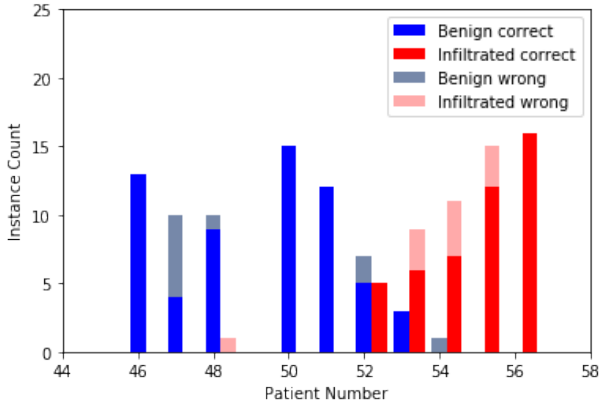
(c) K-fold CV with ConvNet classifier



(d) LOPO with ConvNet classifier

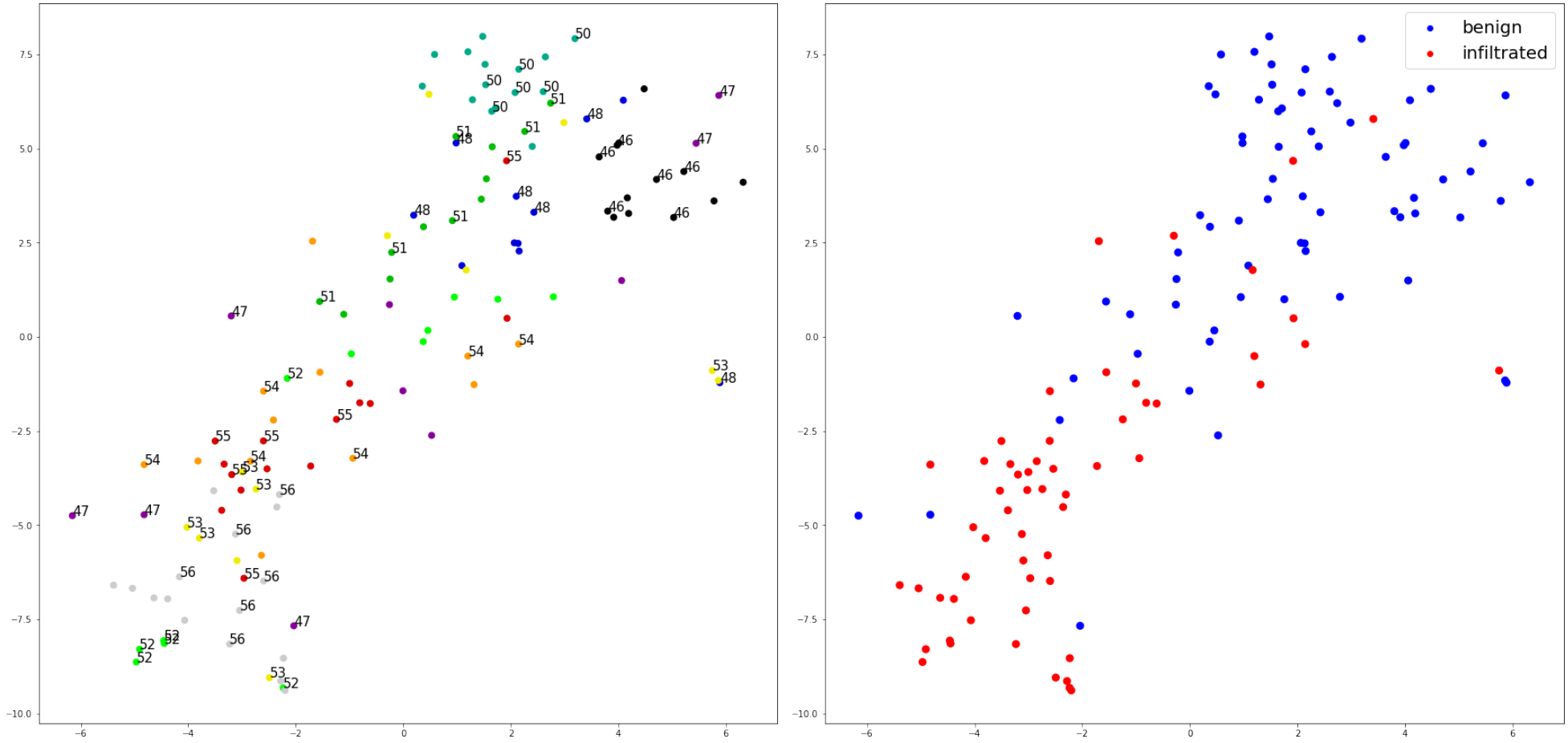


(e) K-fold CV with SVM classifier



(f) LOPO with SVM classifier

Figure 5.21: Classification results per patient per label for SVM and ConvNet using the K-fold CV scheme (left) and LOPO CV scheme (right) in the brain dataset



(a) t-SNE labelled according to patient number

(b) t-SNE labelled according to pathology diagnosis

Figure 5.22: t-SNE dimensionality reduction visualization for the prostate dataset

5.7 Transfer Learning

In both the brain and prostate datasets, one of the main limitations is the amount of data available for analysis. In the case of prostates, obtaining more samples per patient is more feasible since the data is currently probed ex-vivo directly after RP and the main limitation is simply how many samples can be acquired before the prostate dries up. In the case of the brain data, samples are acquired at the neurosurgeons' discretion while operating thus substantially limiting how many signals can be acquired per patient. It would be of great value if we could find common features from one organ to another, i.e. from the brain to the prostate, in order to train on an extensive set of signals and complement the limited dataset with the information obtained from the plentiful dataset. Since the Raman spectrums are a result of the excitation of certain molecules, it is not unreasonable to hypothesize that there might be similarities between datasets. In order to explore this hypothesis, we combined the data of the brain and prostate datasets together. To do so, we aligned the x-axes according to the calibration results of each dataset and used the same preprocessing algorithms and parameters to ensure that the signals be as closely related as possible and avoid sources impacting signal divergence. We then reduced the combined data via PCA and used t-SNE to visualize if clustering occurred between datasets. If the data shows clear clusters between sets, then this suggests that the feature space might be inherently different in both cases and that transfer learning might not be adequate.

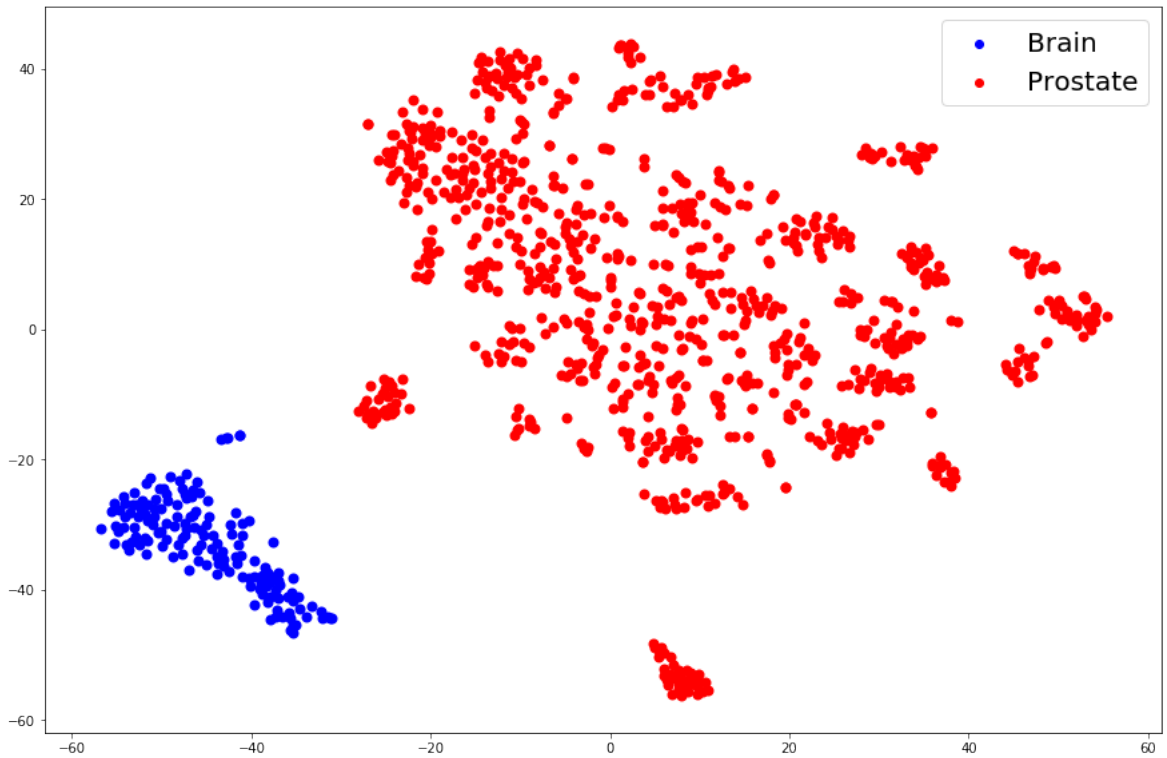


Figure 5.23: t-SNE applied to the combination of the brain and prostate datasets reduced via PCA

Figure 5.23 shows the t-SNE representation of the combined brain and prostate datasets. t-SNE allows us to visualize a clear separation between the brain and prostate datasets with what appears to be a linearly separable plane. This suggests that the data appears to

Table 5.19: AUC of the ROC on the brain dataset complemented with prostate training data using K-fold CV

MLP (AUC=)	ConvNet (AUC=)	SVM (AUC=)
0.907 ± 0.037	0.906 ± 0.061	0.948 ± 0.030

Table 5.20: AUC of the ROC on the brain dataset when using the weights from the prostate dataset as a starting point using K-fold CV

MLP (AUC=)	ConvNet (AUC=)
0.928 ± 0.034	0.955 ± 0.030

contain mutually exclusive information in terms of their PCA components. Thus, simply augmenting the dataset by complementing the brain dataset with prostate data or vice-versa would likely not contribute to any meaningful performance. To test this hypothesis, we complemented the brain dataset with prostate data such that the brain training sets were augmented with the prostate data. We used the same K-fold scheme as in the previous sections for the brain dataset and computed the AUC of the ROC curves. Results are presented in Table 5.19.

We noticed a significant performance drop in the ConvNet, while the SVM performance seems less affected by the addition of prostate data to the brain dataset, which might indicate that support vectors were found by discriminating the prostate data from the brain data. Furthermore, we notice a performance boost with the MLP for the brain dataset compared to the original MLP on the brain dataset. This suggests that the complexity of the MLP model used might simply be too high to capture any meaningful representation from the smaller brain dataset compared to the larger prostate dataset when compared to the SVM and ConvNet. This also suggests that augmenting the training set by simply injecting data from another organ does not increase classification results in any meaningful way.

Alternatively, in the case of MLP and ConvNets, it is possible to use the weights learned by the ConvNets and MLP from the prostate dataset as a starting point for learning features on the brain dataset. By doing so, we attempt to get the network to learn a representation for the data from a local minima instead of initializing the weights from a pseudo-random state as was done in the previous sections. To do so, we used the prostate dataset in its entirety to train a ConvNet and MLP, and used the resulting weights as the initialization to our ConvNet and MLP on the brain dataset. We then used the same K-fold CV scheme used previously and compute the AUC of the ROC and the same LOPO approach to compare performances. Table 5.20 summarizes the AUC of the brain dataset using transfer learning and a K-fold scheme. We saw a significant gain in the MLP and we also noticed some gain in the ConvNet in terms of AUC. Transfer learning appears to increase the AUC in the case of the ConvNet and MLP when compared to its initial performance on the brain dataset and the ConvNet AUC matches SVM performance which was found to be highest in the previous sections.

Tables 5.21 and 5.22 show the confusion matrices and metrics when using both K-fold CV and LOPO approaches for a threshold of 0.5. We notice that once again, the

		Prediction		
		Benign	Infiltrated	Accuracy: 0.867
MLP	Benign	64	7	Specifitiy: 0.901
	Infiltrated	10	47	Sensitivity: 0.825

		Prediction		
		Benign	Infiltrated	Accuracy: 0.891
ConvNet	Benign	66	5	Specifitiy: 0.930
	Infiltrated	9	48	Sensitivity: 0.842

Table 5.21: Confusion matrices and metrics using a K-fold CV scheme and different classifiers in the brain dataset

confusion matrices metrics are better when using the K-fold CV approach compared to the LOPO approach, and that these are comparable to what was obtained without use of transfer learning in the previous sections. This seems to suggest that the feature spaces of different organs are independent, and that while transfer learning might help the network to converge quicker, it does not offer a significant boost to the overall metrics. This also suggests that obtaining more data for a given organ is of more importance than obtaining data from another organ to improve training.

		Prediction		
		Benign	Infiltrated	Accuracy: 0.813
MLP	Benign	61	10	Specifitiy: 0.859
	Infiltrated	14	43	Sensitivity: 0.754

		Prediction		
		Benign	Infiltrated	Accuracy: 0.828
ConvNet	Benign	63	8	Specifitiy: 0.887
	Infiltrated	14	43	Sensitivity: 0.754

Table 5.22: Confusion matrices and metrics using a LOPO CV scheme and different classifiers in the brain dataset

Chapter 6

Conclusions

In this thesis, we explore cancer diagnosis of cells in-vivo in the human body in real-time using Raman spectroscopy. The need for such a method arises when surgically removing tumors to aid surgeons in determining the boundaries of cancer cells, since residual cancer cells directly impact survial rates. We focused on the preprocessing steps involved and on classification results using different classifier architectures, metrics and cross-validation schemes. We focused on data obtained clinically from human brain and prostates.

We studied different AF removal methods and showed that the Zhang and IModPoly methods resulted in similar classification results using K-fold CV on the prostate dataset, with AUC of 0.920 $\pm$ 0.019 and 0.925 $\pm$ 0.035 respectively, the IModPoly method was orders of magnitude faster to implement and thus more appropriate for any real-time implementation.

We then studied the performance of classifiers on the prostate dataset in distinguishing between malignant and benign cells. MLP, ConvNet and SVM were compared using the AUC of the mean ROC from K-fold cross-validation as a means to select for the best parameters and hyper-parameters of each classifier. We showed that in the prostate set, MLP structure didn't have significant impact on the performance of the classifier, and found a maximum AUC score for a newtork consiting of layer size $l1$=$l2$=40 and dropout $d = 0.1$. ConvNets were found to be optimal for kernel size l_{kern}=128 and dropout rate of 0.25. SVM were found to be optimal using an RBF kernel. Overall, SVM and ConvNet performed similarly in terms of AUC, with AUC of 0.943$\pm$0.018 and 0.941$\pm$0.017 and slightly outperformed MLP with AUC of 0.935$\pm$0.021. For all three classifiers, using a threshold of 0.5, we found that the sensitivity was worse overall than the specificity of the system. The ConvNet achieved an accuracy, specificity and sensitivity of 0.921, 0.947, 0.785 respectively, while the SVM achieved 0.913, 0.939 and 0.780 respectively and the MLP achieved 0.908, 0.939 and 0.744. We've shown that augmenting the data directly in the original feature space by varying averaging coefficients did not improve classification results. We've shown that PCA is more robust than AE when implementing dimensionality reduction and have shown that we can obtain comparable classification results when reducing the datasets by 2.5x using both PCA and AE, however classification results were better overall when using the original spectrum as input. We've also shown the effect of using a LOPO CV scheme and noted significantly worse sensitivity metrics when employing a LOPO approach over a K-fold CV approach, suggesting clustering within patients, and have shown visually using t-SNE evidence of said clustering. This suggests that more robust protocols during signal acquisition might be needed and that pre-training of the network during surgery

with signals acquired from regions of known pathology by the surgeon might help improve sensitivity and overall performance of the system.

In the brain dataset, we studied the performance of classifiers at distinguishing between benign and infiltrated regions of the brain. We have also shown that ConvNet and SVM perform similarly and overall and that in the brain dataset, the architecture of MLP had more of an effect on the overall performance, likely due to the much smaller dataset size. We also show that LOPO results are worse than K-fold CV results, as was the case in the prostate dataset, suggesting once more that evaluation metrics should take in to account patient ID and cross-validation methods should account for patient groups. However, clustering within patients was not as obvious when employing t-SNE as in the prostate dataset, suggesting that the protocols involved during brain data acquisition were subject to less variability from patient to patient. We also studied the prospect of using transfer learning by leveraging the information in the prostate dataset on the brain dataset. We have shown that simply complementing the brain dataset with prostate signals decreased performance for both SVM and ConvNet. We also showed that using transfer learning, i.e. using the weights learned from the prostate set as a starting point to train the brain networks, in the case of both the ConvNet and MLP, K-fold results performed slightly better, suggesting that pre-training the weights on another dataset can slightly improve calssification accuracy, however this demonstrates that obtaining more data for a given organ is more important than leveraging existing data from another organ.

6.1 Contributions to the field

We focused our attention on data gathered from human brains and prostates in-vivo using Raman spectroscopy. Our contributions to the field are as follows:

- We've shown that preprocessing steps have a significant impact on classification results and presented those that are best suited for real-time Raman spectroscopy
- We've shown that SVM and ConvNet consistently outperform MLP on our brain and prostate datasets
- We've explored the use of data augmentation on our datasets and have shown negligible improvements on classification metrics
- We've explored the use of dimensionality reduction using PCA and AE and have shown that while PCA is better suited than AE, classification results are optimal on the original feature space
- We've explored transfer learning, and shown that there was marginal improvement in using one organ to fine-tune the weights of a classifier on a different organ
- We've shown evidence of clustering within patients in our datasets suggesting that metrics acquired using LOSO in literature can be inherently biased and overly optimistic and that LOPO should be used to avoid bias and to mimic clinical settings applications

Overall, we've shown that the choice of classifier does not have significant impact on classification in the limited datasets that we have. Clustering of the data in both datasets suggests that having robust protocols for data acquisition and processing are necessary and perhaps even more important than intricate classifiers. In fact, in both the prostate and brain datasets, SVM seem to generalize well and seem to be well suited for the task of classification when applied directly to the original feature space. Future work should focus on collecting more data and seeing if the results presented in this thesis still hold on larger amounts of data. If clustering still occurs, studying in depth the ramification of LOPO versus K-fold CV and LOSO is necessary, since high performance of K-fold and low performance of LOPO can lead to poor performance when deployed clinically. Future work should also focus on real-time pre-training of networks from spectrums acquired by the surgeon in regions of known or obvious pathology.

In conclusion, Raman spectroscopy shows promise as a real-time diagnostics tool in the brain and prostate, however there are still limitations to classification performance and improvements that need to be made before it can be implemented for real-time clinical use. The lack of available data, due to the very nature of the procedures, make classification more challenging and perhaps exploring developing models as opposed to training supervised learning algorithms could complement current classification results and prove to be beneficial in the long-run.

References

[1] Demissie Alemayehu and Kelly H Zou. Applications of roc analysis in medical research: recent developments and future directions. *Academic radiology*, 19(12):1457–1464, 2012.

[2] Lauren A Austin, Sam Osseiran, and Conor L Evans. Raman technologies in cancer diagnostics. *Analyst*, 141(2):476–503, 2016.

[3] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

[4] Rasmus Bro and Age K Smilde. Principal component analysis. *Analytical Methods*, 6(9):2812–2831, 2014.

[5] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[6] BWTEK. Theory of raman scattering, Accessed: 2017-05-31. http://bwtek.com/raman-theory-of-raman-scattering/.

[7] Orchid Cancer. Surgery (radical prostatectomy). https://orchid-cancer.org.uk/prostate-cancer/localised/surgery/. Accessed: 2017-07-04.

[8] LJ Cao, Kok Seng Chua, WK Chong, HP Lee, and QM Gu. A comparison of pca, kpca and ica for dimensionality reduction in support vector machine. *Neurocomputing*, 55(1):321–336, 2003.

[9] Sigrid Carlsson and Andrew Vickers. Spotlight on prostate cancer: the latest evidence and current controversies. *BMC medicine*, 13(1):60, 2015.

[10] François Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[11] Steven J Choquette, Edgar S Etz, Wilbur S Hurst, Douglas H Blackburn, and Stefan D Leigh. Relative intensity correction of raman spectrometers: Nist srms 2241 through 2243 for 785 nm, 532 nm, and 488 nm/514.5 nm excitation. *Applied spectroscopy*, 61(2):117–129, 2007.

[12] Martin Člupek, Pavel Matějka, and Karel Volka. Noise reduction in raman spectra: Finite impulse response filtration versus savitzky–golay smoothing. *Journal of Raman Spectroscopy*, 38(9):1174–1179, 2007.

[13] Norman Colthup. *Introduction to infrared and Raman spectroscopy*. Elsevier, 2012.

[14] P Crow, A Molckovsky, N Stone, J Uff, B Wilson, and L-M WongKeeSong. Assessment of fiberoptic near-infrared raman spectroscopy for diagnosis of bladder and prostate cancer. *Urology*, 65(6):1126–1130, 2005.

[15] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[16] Wolfgang Demtröder. *Laser spectroscopy: basic concepts and instrumentation*. Springer Science & Business Media, 2013.

[17] Joannie Desroches, Michael Jermyn, Kelvin Mok, Cédric Lemieux-Leduc, Jeanne Mercier, Karl St-Arnaud, Kirk Urmey, Marie-Christine Guiot, Eric Marple, Kevin Petrecca, et al. Characterization of a raman spectroscopy probe system for intraoperative brain tissue classification. *Biomedical optics express*, 6(7):2380–2397, 2015.

[18] Mary Weinstein Dunn and Meredith Wallace Kazer. Prostate cancer overview. In *Seminars in oncology nursing*, volume 27, pages 241–250. Elsevier, 2011.

[19] Tom Fawcett. Roc graphs: Notes and practical considerations for researchers. *Machine learning*, 31(1):1–38, 2004.

[20] Miguel Fernandez. Cross-validation. [Online; accessed June 12, 2017].

[21] World Cancer Research Fund. Worldwide data. http://www.wcrf.org/int/cancer-facts-figures/worldwide-data. Accessed: 2017-08-09.

[22] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)âĂŤa review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14):2627–2636, 1998.

[23] Glosse.ca. Artificial neural network with layer coloring. [Online; accessed June 12, 2017].

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[25] Markus Graefen, Thorsten Schlomm, Guido Sauter, Hartwig Huland, et al. Detailed quantification of high-grade cancer allows precise prediction of prostate cancer prognosis. *European urology*, 69(3):436–437, 2016.

[26] EB Hanlon, R Manoharan, T_W Koo, KE Shafer, JT Motz, M Fitzmaurice, JR Kramer, I Itzkan, RR Dasari, and MS Feld. Prospects for in vivo raman spectroscopy. *Physics in medicine and biology*, 45(2):R1, 2000.

[27] IBM. Interpolation of the points (roc curve algorithms). https://www.ibm.com/support/knowledgecenter/fr/SSLVMB_20.0.0/com.ibm.spss.statistics.help/alg_roc_interpolation.htm. Accessed: 2017-08-09.

[28] Michael Jermyn, Joannie Desroches, Kelly Aubertin, Karl St-Arnaud, Wendy-Julie Madore, Etienne De Montigny, Marie-Christine Guiot, Dominique Trudel, Brian C Wilson, Kevin Petrecca, et al. A review of raman spectroscopy advances with an emphasis on clinical translation challenges in oncology. *Physics in Medicine and Biology*, 61(23):R370, 2016.

[29] Michael Jermyn, Joannie Desroches, Jeanne Mercier, Marie-Andrée Tremblay, Karl St-Arnaud, Marie-Christine Guiot, Kevin Petrecca, and Frederic Leblond. Neural networks improve brain cancer detection with raman spectroscopy in the presence of operating room light artifacts. *Journal of biomedical optics*, 21(9):094002–094002, 2016.

[30] Michael Jermyn, Kelvin Mok, Jeanne Mercier, Joannie Desroches, Julien Pichette, Karl Saint-Arnaud, Liane Bernstein, Marie-Christine Guiot, Kevin Petrecca, and Frederic Leblond. Intraoperative brain cancer detection with raman spectroscopy in humans. *Science translational medicine*, 7(274):274ra19–274ra19, 2015.

[31] Rachel Kast, Gregory Auner, Sally Yurgelevic, Brandy Broadbent, Aditya Raghunathan, Laila M Poisson, Tom Mikkelsen, Mark L Rosenblum, and Steven N Kalkanis. Identification of regions of normal grey matter and white matter from pathologic glioblastoma and necrosis in frozen sections using raman imaging. *Journal of neuro-oncology*, 125(2):287, 2015.

[32] Rachel E Kast, Stephanie C Tucker, Kevin Killian, Micaela Trexler, Kenneth V Honn, and Gregory W Auner. Emerging technology: applications of raman spectroscopy for prostate cancer. *Cancer and Metastasis Reviews*, 33(2-3):673, 2014.

[33] keras.io. Autoencoder schema. [Online; accessed June 12, 2017].

[34] Matthias Kirsch, Gabriele Schackert, Reiner Salzer, and Christoph Krafft. Raman spectroscopic imaging for in vivo detection of cerebral brain metastases. *Analytical and bioanalytical chemistry*, 398(4):1707–1713, 2010.

[35] S Koljenović, TC Bakker Schut, R Wolthuis, AJPE Vincent, G Hendriks-Hagevi, L Santos, JM Kros, and GJ Puppels. Raman spectroscopic characterization of porcine brain tissue using a single fiber-optic probe. *Analytical chemistry*, 79(2):557–564, 2007.

[36] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques, 2007.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[38] Bhagwandas Pannalal Lathi et al. *Linear systems and signals*, volume 2. Oxford University Press New York:, 2005.

[39] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[40] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256. IEEE, 2010.

[41] Biao Leng, Kai Yu, and QIN Jingyan. Data augmentation for unbalanced face recognition training sets. *Neurocomputing*, 235:10–14, 2017.

[42] Chad A Lieber and Anita Mahadevan-Jansen. Automated method for subtraction of fluorescence from biological raman spectra. *Applied spectroscopy*, 57(11):1363–1367, 2003.

[43] Chad A Lieber and Anita Mahadevan-Jansen. Automated method for subtraction of fluorescence from biological raman spectra. *Applied spectroscopy*, 57(11):1363–1367, 2003.

[44] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[45] Aritake Mizuno, Takashi Hayashi, Kouichi Tashibu, Shuichi Maraishi, Kazuaki Kawauchi, and Yukihiro Ozaki. Near-infrared ft-raman spectra of the rat brain tissues. *Neuroscience letters*, 141(1):47–52, 1992.

[46] Monica Monici. Cell and tissue autofluorescence research and diagnostic applications. *Biotechnology annual review*, 11:227–256, 2005.

[47] Michael A Nielsen. Neural networks and deep learning. *URL: http://neuralnetworksanddeeplearning. com/.(visited: 01.11. 2014)*, 2015.

[48] World Health Organization. Cancer (fact sheet). http://www.who.int/mediacentre/factsheets/fs297/en/. Accessed: 2017-08-09.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[50] Marc D Porter, Thomas B Bright, David L Allara, and Christopher ED Chidsey. Spontaneously organized molecular assemblies. 4. structural characterization of n-alkyl thiol monolayers on gold by optical ellipsometry, infrared spectroscopy, and electrochemistry. *Journal of the American Chemical Society*, 109(12):3559–3568, 1987.

[51] Markus Ringnér. What is principal component analysis? *Nature biotechnology*, 26(3):303, 2008.

[52] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[53] Simon Schaffer. Glass works: NewtonâĂŹs prisms and the uses of experiment. *The uses of experiment: Studies in the natural sciences*, pages 67–104, 1989.

[54] Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2015. *CA: a cancer journal for clinicians*, 65(1):5–29, 2015.

[55] Douglas Arvid Skoog and James J Leary. Principles of instrumental analysis. *Clinical Chemistry-Reference Edition*, 40(8):1612, 1994.

[56] American Cancer Society. Which treatments are used for prostate cancer? https://www.cancer.org/cancer/prostate-cancer/treating.html. Accessed: 2017-07-04.

[57] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[58] BWKP Stewart, Christopher P Wild, et al. World cancer report 2014. *Health*, 2017.

[59] Nicholas Stone, Maria Consuelo Hart Prieto, Paul Crow, Jeremy Uff, and Alistair William Ritchie. The use of raman spectroscopy to provide an estimation of the gross biochemistry associated with urological pathologies. *Analytical and bioanalytical chemistry*, 387(5):1657–1668, 2007.

[60] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[61] JF Villa-Manríquez, J Castro-Ramos, F Gutiérrez-Delgado, MA Lopéz-Pacheco, and AE Villanueva-Luna. Raman spectroscopy and pca-svm as a non-invasive diagnostic tool to identify and classify qualitatively glycated hemoglobin levels in vivo. *Journal of Biophotonics*, 2016.

[62] Lei Wang, Dalin He, Jin Zeng, Zhenfeng Guan, Qiang Dang, Xinyang Wang, Jun Wang, Liqing Huang, Peilong Cao, Guanjun Zhang, et al. Raman spectroscopy, a potential tool in diagnosis and prognosis of castration-resistant prostate cancer. *Journal of biomedical optics*, 18(8):087001–087001, 2013.

[63] Steven B Zeliadt, Scott D Ramsey, David F Penson, Ingrid J Hall, Donatus U Ekwueme, Leonard Stroud, and Judith W Lee. Why do men choose one treatment over another? *Cancer*, 106(9):1865–1874, 2006.

[64] Zhi-Min Zhang, Shan Chen, Yi-Zeng Liang, Zhao-Xia Liu, Qi-Ming Zhang, Li-Xia Ding, Fei Ye, and Hua Zhou. An intelligent background-correction algorithm for highly fluorescent samples in raman spectroscopy. *Journal of Raman Spectroscopy*, 41(6):659–669, 2010.

[65] Jianhua Zhao, Harvey Lui, David I McLean, and Haishan Zeng. Automated autofluorescence background subtraction algorithm for biomedical raman spectroscopy. *Applied spectroscopy*, 61(11):1225–1232, 2007.